DATECS

Communication protocol

version: 2.10

Programmer's

Manual

2020

The description implied that the reader has programming skills in one or several programming languages, as well as is familiar with the equipment used, at least at the level of the operator's manual supplied with it.

Programmer's Manual
Document version: 2.10.0 / 14-Feb-2020

Contents

1. Introduction....1
2. Copyright....1
3. Low level protocol....1
3.1. Protocol type – Master (Host) / Slave....1
3.2. Sequence of the messages....1
3.3. Non-wrapped messages – time-out....1
3.4. Wrapped message format....2
3.5. Message composition, syntax and meanings....3
3.6. Command explanations....3
4. Commands....4
4.1. Command 33 (21h) Clears the external display....4
4.2. Command 35 (23h) Displaying text on second line of the external display....5
4.3. Command 38 (26h) Opening a non-fiscal receipt....6
4.4. Command 39 (27h) Closing a non-fiscal receipt....7
4.5. Command 42 (2Ah) Printing of a free non-fiscal text....8
4.6. Command 43 (2Bh) Opening of storno documents....9
4.7. Command 44 (2Ch) Paper feed....10
4.8. Command 45 (2Dh) Check for mode connection with PC....11
4.9. Command 46 (2Eh) Paper cutting....12
4.10. Command 47 (2Fh) Displaying text on upper line of the external display....13
4.11. Command 48 (30h) Open fiscal receipt....14
4.12. Command 49 (31h) Registration of sale....15
4.13. Command 50 (32h) Return the active VAT rates....17
4.14. Command 51 (33h) Subtotal....18
4.15. Command 53 (35h) Payments and calculation of the total sum (TOTAL)....19
4.15.1. Standard payment types....19
4.15.2. Payment type – Card with pinpad....20
4.15.3. Payment type – foreign currency....21
4.15.4. Payment type – Card with pinpad and returning data for transaction....22
4.16. Command 54 (36h) Printing of a free fiscal text....24
4.17. Command 55 (37h) Pinpad commands....25
4.17.1. Pinpad commands – option ‘1’-Void....25
4.17.2. Pinpad commands – option ‘2’-Copy of last document....26
4.17.3. Pinpad commands – option ‘3’-Copy of document by type....27
4.17.4. Pinpad commands – option ‘4’-Copy of all documents....27
4.17.5. Pinpad commands – option ‘5’-End of day from pinpad....28
4.17.6. Pinpad commands – option ‘6’-Report from pinpad....28
4.17.7. Pinpad commands – option ‘7’-Full report from pinpad....29
4.17.8. Pinpad commands – option ‘8’-Enter date and time for pinpad....29
4.17.9. Pinpad commands – option ‘9’-Check connection with pinpad....30
4.17.10. Pinpad commands – option ‘10’-Check connection with server....30
4.17.11. Pinpad commands – option ‘11’-Loyalty balance....30
4.17.12. Pinpad commands – option ‘12’-Get update....31
4.17.13. Pinpad commands – option ‘13’-After errors by CMD53 or CMD55 (opt 14)....31
4.17.14. Pinpad commands – option ‘14’-Make sale from pinpad, without fiscal receipt....32
4.17.15. Pinpad commands – option ‘15’-Print receipt for pinpad after successful transaction....34
4.18. Command 56 (38h) Close fiscal receipt....35
4.19. Command 57 (39h) Enter and print invoice data....36
4.20. Command 58 (3Ah) Registering the sale of a programmed item....37
4.21. Command 60 (3Ch) Cancel fiscal receipt....38
4.22. Command 61 (3Dh) Set date and time....39
4.23. Command 62 (3Eh) Read date and time....40
4.24. Command 63 (3Fh) Show current date and time on the external display....41
4.25. Command 64 (40h) Information on the last fiscal entry....42
4.26. Command 65 (41h) Information on daily taxation....43
4.27. Command 66 (42h) Set invoice interval....44
4.28. Command 68 (44h) Number of remaining entries for Z-reports in FM....45
4.29. Command 69 (45h) Reports....46
4.29.1. Report X and report Z....46
4.29.2. Report D, report G....47
4.29.3. Report P – print the periodical report....48
4.30. Command 70 (46h) Cash in and Cash out operations....48
4.31. Command 71 (47h) General information, modem test....50
4.31.1. General information, modem test....50
4.31.2. Information about the connection with NRA server....51
4.32. Command 72 (48h) Fiscalization....52
4.33. Command 74 (4Ah) Reading the Status....53
4.34. Command 76 (4Ch) Status of the fiscal transaction....55
4.35. Command 80 (50h) Play sound....56
4.36. Command 83 (53h) Programming of VAT rates....57
4.37. Command 84 (54h) Printing of barcode....58
4.38. Command 86 (56h) The date of the last record in the fiscal memory....59
4.39. Command 87 (57h) Get item groups information....60
4.40. Command 88 (58h) Get department information....61
4.41. Command 89 (59h) Test of the fiscal memory....62
4.42. Command 90 (5Ah) Diagnostic information....63
4.43. Command 91 (5Bh) Programming of Serial number and FM number....63

4.44. Command 92 (5Ch) Printing of separating line....64
4.45. Command 94 (5Eh) Fiscal memory report by date....65
4.46. Command 95 (5Fh) Fiscal memory report by number of Z-report....66
4.47. Command 96 (60h) Set software password....67
4.48. Command 98 (62h) Programming of TAX number....68
4.49. Command 99 (63h) Reading the programmed TAX number....69
4.50. Command 100 (64h) Reading an error....70
4.51. Command 101 (65h) Set operator password....71
4.52. Command 103 (67h) Information for the current receipt....72
4.53. Command 105 (69h) Print of operator's report....73
4.54. Command 106 (6Ah) Drawer opening....74
4.55. Command 107 (6Bh) Defining and reading items....75
4.55.1. Item programming – option 'P' - Programming item....75
4.55.2. Item programming – option 'I' - Item information....77
4.55.3. Item programming – option 'A' - Add stock quantity for item....77
4.55.4. Item programming – option 'D' - Item deleting....78
4.55.5. Item programming – option 'R' - Reading item....79
4.55.6. Item programming – option 'F' - Data about the first found programmed item....80
4.55.7. Item programming – option 'L' - Data about the last found programmed item....81
4.55.8. Item programming – option 'N' - Data for the next found programmed item....82
4.55.9. Item programming – option 'f' - Data about the first found item with sales on it....83
4.55.10. Item programming – option 'l' - Data about the last found item with sales on it....84
4.55.11. Item programming – option 'n' - Data for the next found programmed item with sales on it....85
4.55.12. Item programming – option 'X' - Find the first not programmed item....86
4.55.13. Item programming – option 'x' - Find the last not programmed item....86
4.56. Command 109 (6Dh) Print duplicate copy of last fiscal receipt....87
4.57. Command 110 (6Eh) Additional daily information....88
4.57.1. Payments (sell operations)....88
4.57.2. Payments (storno operations)....89
4.57.3. Number and sum of sells....90
4.57.4. Count and sum of discounts and surcharges....90
4.57.5. Count and sum of corrections and annulled receipts....91
4.57.6. Count and sum of cash in and cash out operations....92
4.58. Command 111 (65h) Print PLU report....93
4.59. Command 112 (70h) Information for operator....94
4.60. Command 116 (74h) Reading fiscal memory binary data....95
4.61. Command 122 (7Ah) Printing of a free vertical fiscal text....95
4.62. Command 123 (7Bh) Device information....96
4.63. Command 124 (7Ch) Search receipt number by period....99
4.64. Command 125 (7Dh) Information from EJ....100
4.64.1. Set document to read....100
4.64.2. Read one line as text....101
4.64.3. Read as data....101
4.64.4. Print document....101
4.64.5. Set document to read in CSV formatted data....102
4.64.6. Read CSV formatted data....103
4.65. Command 126 (7Eh) Fiscal memory-structured information....103
4.65.1. Ask for non-empty and max records....103
4.65.2. Ask for Z reports structured information....105
4.65.3. Ask for device ID number....107
4.65.4. Ask for fiscal memory number....107
4.65.5. Ask for date of fiscalization....108
4.65.6. Ask for TAX number changes....108
4.65.7. Ask for vat rate changes....109
4.65.8. Ask for memory resetting events....110
4.65.9. Ask for NRA registrations events....110
4.65.10. Ask for NRA unregistrations events....111
4.65.11. Ask for EJ (KLEN) changes....111
4.66. Command 127 (7Fh) Stamp operations [*32]....112
4.67. Command 135 (87h) Modem information....112
4.67.1. Modem information – option 's' - Read IMEI of the modem....112
4.67.2. Modem information – option 'i' - Read the IMSI of the SIM card....112
4.67.3. Modem information – option 'M' - Modem status....113
4.68. Command 140 (8Ch) Defining and reading clients....114
4.68.1. Clients programming – option 'I' - Clients information....114
4.68.2. Clients programming – option 'P' - Programming clients....115
4.68.3. Clients programming – option 'D' - Client deleting....116
4.68.4. Clients programming – option 'R' - Reading client data....117
4.68.5. Clients programming – option 'F' - Data about the first found programmed client....118
4.68.6. Clients programming – option 'L' - Data about the last found programmed client....119
4.68.7. Clients programming – option 'N' - Data for the next found programmed client....120
4.68.8. Clients programming – option 'T' - Find a client by tax number....121
4.68.9. Clients programming – option 'X' - Find the first not programmed client....122
4.68.10. Clients programming – option 'x' - Find the last not programmed client....123
4.69. Command 202 (CAh) Customer graphic logo loading. [*32]....124
4.69.1. Parameters description....124
4.70. Command 203 (CAh) Stamp image loading [*32]....125
4.70.1. Parameters description....125
4.71. Command 253 (FDh) Service operations....126

4.71.1. Entering service password....126
4.71.2. Change service password....126
4.71.3. Close the current EJ....126
4.71.4. Factory setting of configuration parameters....126
4.71.5. Clear errors from NRA server communication. Unblock the blocked device....127
4.71.6. Send all unsent documents to the NRA servers....127
4.72. Command 255 (FFh) Programming....127
4.72.1. Read parameters....127
4.72.2. Write parameters....128
4.72.3. Parameters description....129
5. Remarks....138
6. Status bits....139
7. Error codes....141

1. Introduction

The communication protocol v.2 is intended for writing drivers of the following fiscal device's models:

- Bulgarian versions:
  - DP-25X
  - DP-05C
  - WP-500X
  - WP-50X
  - FP-700X
  - FP-700XR
  - FMP-350X
  - FMP-55X

The fiscal device operates under the control of an application program, with which communicates via RS232 (USB, LAN or WLAN) 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.

2. Copyright

This protocol is subject to the copyright of Datecs ltd. The protocol can be freely used only for writing the drivers of these fiscal devices and connecting various devices with them. This exchange protocol cannot be used for implementation in other fiscal devices without the written consent of Datecs ltd.

3. Low level protocol

3.1. Protocol type – Master (Host) / Slave

The fiscal device performs the commands sent by the Host and returns messages, which depend on the result. The fiscal device 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 device maintains the communication via the RS232 serial connection at baud rates of 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200 b/s, 8N1. For other types of interfaces, the speed can't be configured.

3.2. Sequence of the messages

Host sends a wrapped message, containing a command for the fiscal device. Device executes the requested operation and response with a wrapped message. Host has to wait for a response from the fiscal device 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.

3.3. 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:

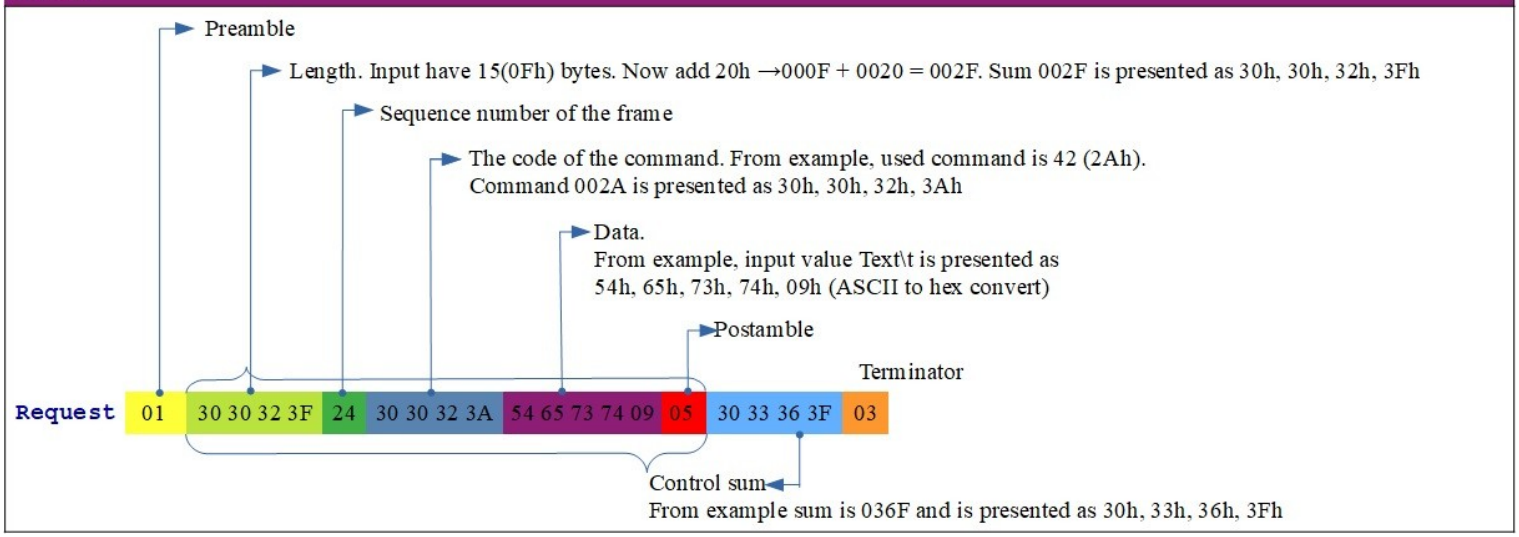
- NAK 15H - this code is sent by Slave when an error in the control sum or the form of the received message is found. When Host receives a NAK it must again send a message with the same sequence number.
- SYN 16H - this code is sent by Slave upon receiving a command which needs longer processing time. SYN is sent every 60 ms until the wrapped message is not ready for transmitting.

3.4. Wrapped message format

Request from host to fiscal device			
Field name	Length in bytes	Value	Description
<PRE>	1	01h	Preamble
<LEN>	4	30303230h…3F3F3F3Fh	Message length. Number of bytes from <PRE> preamble (excluded) to <PST> (included) plus the fixed offset of 20h. ASCII-hex format is used. Each digit from the four bytes is sent after 30h is added to it.
<SEQ>	1	20h...FFh	Sequence number of the frame. The fiscal device saves the same <SEQ> in the return message. If the fiscal device 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>	4	30303230h…3F3F3F3Fh	The code of the command. The fiscal device saves the same <CMD> in the return message. If the fiscal device receives a non-existing code it returns a wrapped message with zero length in the data field and sets the respective status bit. ASCII-hex format is used. Each digit from the four bytes is sent after 30h is added to it.
<DATA>	0...496	20h...FFh	Command data. 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.
<PST>	1	05h	Postamble
<BCC>	4	30303030h…3F3F3F3Fh	Control sum. The sum includes between <PRE> preamble (excluded) to <PST>. ASCII-hex format is used. Each digit from the four bytes is sent after 30h is added to it.
<EOT>	1	03h	Terminator

Answer from fiscal device to host			
Field name	Length in bytes	Value	Description
<PRE>	1	01h	Preamble
<LEN>	4	30303230h…3F3F3F3Fh	Message length. Number of bytes from <PRE> preamble (excluded) to <PST> (included) plus the fixed offset of 20h. ASCII-hex format is used. Each digit from the four bytes is sent after 30h is added to it.
<SEQ>	1	20h...FFh	Sequence number of the frame. The fiscal device saves the same <SEQ> in the return message. If the fiscal device 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>	4	30303230h…3F3F3F3Fh	The code of the command. The fiscal device saves the same <CMD> in the return message. If the fiscal device receives a non-existing code it returns a wrapped message with zero length in the data field and sets the respective status bit. ASCII-hex format is used. Each digit from the four bytes is sent after 30h is added to it.
<DATA>	0...480	20h...FFh	Command data. 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.
<SEP>	1	04h	Separator
<STAT>	8	8080808080808080h...FFFFFFFFFFFFFFFF	The field with the current status bits of the fiscal device
<PST>	1	05h	Postamble
<BCC>	4	30303030h…3F3F3F3Fh	Control sum. The sum includes between <PRE> preamble (excluded) to <PST>. ASCII-hex format is used. Each digit from the four bytes is sent after 30h is added to it.
<EOT>	1	03h	Terminator

Example



3.5. Message composition, syntax and meanings

- The data field depends on the command.
- The parameters sent to the fiscal device may be separated with a [\t] and/or may have a fixed length.
- The separator([\t]) between the parameters shows that it is mandatory.
- 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.

Example: when we write 255,ExchangeRate[\t][\t][\t] for the data field then in that field there will be 45 78 63 68 61 6E 67 65 52 61 74 65 09 09 09 where each hexadecimal digit is an ASCII value.

3.6. Command explanations

Example command syntax: {Parameter1}<SEP>{Parameter2}<SEP>{Parameter3}<SEP><DateTime><SEP>

<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;
- 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;

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.

4. Commands

4.1. Command 33 (21h) Clears the external display

Request (no parameters)					
	Name	Type	Opt	Value	Description

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3A 24 30 30 32 31 05 30 31 3B 38 03
Answer	01 30 30 33 35 24 30 30 32 31 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 31 31 03
Human oriented log	
Request	
Answer	0[\t]

The command is not used on FMP-350X and FMP-55X

4.2. Command 35 (23h) Displaying text on second line of the external display

Request

	Name	Type	Opt	Value	Description
1	**Text**	char		Up to 20 symbols	Text to be sent directly to the external display.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example

Binary log	
Request	01 30 30 33 3C 25 30 30 32 33 54 65 73 74 20 74 65 78 74 20 64 69 73 70 6C 61 79 09 05 30 38 36 32 03
Answer	01 30 30 33 35 25 30 30 32 33 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 31 34 03
Human oriented log	
Request	Test text display[\t]
Answer	0[\t]

The command is not used on FMP-350X and FMP-55X

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

Request (no parameters)					
	Name	Type	Opt	Value	Description

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**SlipNumber**	uint	1...9999999	Current slip number.

Example	
Binary log	
Request	01 30 30 32 3A 2D 30 30 32 36 05 30 31 3C 36 03
Answer	01 30 30 33 3B 2D 30 30 32 36 30 09 31 36 34 39 35 09 04 80 80 A0 80 86 9A 80 80 05 30 37 35 37 03
Human oriented log	
Request	
Answer	0[\t]467[\t]

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

Request (no parameters)					
	Name	Type	Opt	Value	Description

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**SlipNumber**	uint	1...9999999	Current slip number.

Example	
Binary log	
Request	01 30 30 32 3A 2E 30 30 32 37 05 30 31 3C 38 03
Answer	01 30 30 33 3B 2E 30 30 32 37 30 09 31 36 34 39 35 09 04 80 80 80 80 86 9A 80 80 05 30 37 33 39 03
Human oriented log	
Request	
Answer	0[\t]467[\t]

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

Request					
	Name	Type	Opt	Value	Description
1	**Text**	char		Up to XX symbols	XX depends on print columns. • XX = 42,48,64[*11] • XX = 42[*8] • XX = 32[*12].
2	**Bold**	uint	●	0...1	1 = print bold text; empty field = normal text;
3	**Italic**	uint	●	0...1	1 = print italic text; empty field = normal text;
4	**Height**	uint	●	0...2	0 = normal height, 1= double height, 2 = half height; empty field = normal height text;
5	**Underline**	uint	●	0...1	1 = print underlined text; empty field = normal text;
6	**alignment**	uint	●	0...2	0 = left alignment, 1 = center, 2 = right; empty field = left alignment;

Answer				
	NaTme	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 34 38 30 30 30 32 3A CF E5 F7 E0 F2 20 ED E0 20 F1 E2 EE E1 EE E4 E5 ED 20 F2 E5 EA F1 F2 09 31 09 09 09 09 09 05 31 34 3C 38 03
Answer	01 30 30 33 35 30 30 30 32 3A 30 09 04 80 80 A0 80 86 9A 80 80 05 30 36 34 36 03
Human oriented log	
Request	Печат на свободен текст[\t]1[\t][\t][\t][\t][\t]
Answer	0[\t]

4.6. Command 43 (2Bh) Opening of storno documents

Request

	Name	Type	Opt	Value	Description
1	**OpCode**	uint		1...30	Operator number.
2	**OpPwd**	char		8 digits	Operator password[*1].
3	**TillNmb**	uint		1...99999	Number of point of sale.
4	**Storno**	uint		0...2	• **'0'** – Opens storno receipt. Reason "operator error"; • **'1'** – Opens storno receipt. Reason "refund"; • **'2'** – Opens storno receipt. Reason "tax base reduction";
5	**DocNum**	uint		1...9999999	Number of the original document.
6	**DateTime**	char		See remark: [*27]	Date and time of the original document (min date 1-1-2000 00:00:00)
7	**FMNumber**	uint		1...99999999	Fiscal memory number of the device the issued the original document.
8	**Invoice**	char	●	Space or “I”	If this parameter has value 'I' it opens an invoice storno/refund receipt.
9	**ToInvoice**	char	●	1...9999999999	If **Invoice** is 'I' - Number of the invoice that this receipt is referred to. If **Invoice** is blank this parameter has to be blank too.
10	**Reason**	char	●	Up to 64 symbols	If **Invoice** is 'I' - Reason for invoice storno/refund. If **Invoice** is blank this parameter has to be blank too.
11	**NSale**	char	●	"LLDDDDDD-CCCC-DDDDDDD", L[A-Z], C[0-9A-Za-z], D[0-9]	Unique sale number. The parameter is not required only if the original document is printed by the cashier and not by the PC program.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**SlipNumber**	uint	1...9999999	Current slip number.

Example

Binary log

Request	01 30 30 36 3F 2F 30 30 32 3B 32 31 09 32 31 09 39 38 37 36 09 30 09 34 32 38 09 32 34 2D 30 34 2D 31 39 20 30 38 3A 33 36 3A 32 37 09 30 32 36 33 36 35 37 31 09 09 09 09 44 54 36 33 36 34 39 37 2D 30 30 32 31 2D 30 30 31 30 30 30 31 09 05 30 3D 3E 3F 03
Answer	01 30 30 33 39 2F 30 30 32 3B 30 09 34 37 30 09 04 80 80 88 80 86 9A 80 80 05 30 36 3D 36 03

Human oriented log

Request	21[\t]21[\t]9876[\t]0[\t]428[\t]24-04-19 08:36:27[\t]02636571[\t][\t][\t][\t]DT636497-0021-0010001[\t]
Answer	0[\t]470[\t]

4.7. Command 44 (2Ch) Paper feed

Request

	Name	Type	Opt	Value	Description
1	**Lines**	uint	●	1...99	Number of lines to feed.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example

Binary log	
Request	01 30 30 32 3C 30 30 30 32 3C 34 09 05 30 32 30 3E 03
Answer	01 30 30 33 35 30 30 30 32 3C 30 09 04 80 80 88 80 86 9A 80 80 05 30 36 33 30 03
Human oriented log	
Request	4[\t]
Answer	0[\t]

4.8. Command 45 (2Dh) Check for mode connection with PC

Request (no parameters) (syntax #1)					
	Name	Type	Opt	Value	Description

Request (no parameters) (syntax #2)					
	Name	Type	Opt	Value	Description
1	**DisablePrinting**	uint	●	0...1	Enable/disable printout[*29]: • **'0'** – Disable printout; • **'1'** – Enable printout;

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3A 22 30 30 32 3D 05 30 31 3C 32 03
Answer	01 30 30 33 35 22 30 30 32 3D 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 31 3B 03
Human oriented log	
Request	
Answer	0[\t]

4.9. Command 46 (2Eh) Paper cutting

Request (no parameters)					
	Name	Type	Opt	Value	Description

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3A 23 30 30 32 3E 05 30 31 3C 34 03
Answer	01 30 30 33 35 23 30 30 32 3E 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 31 3D 03
Human oriented log	
Request	
Answer	0[\t]

The command is only used on FP-700X

4.10. Command 47 (2Fh) Displaying text on upper line of the external display

Request

	Name	Type	Opt	Value	Description
1	**Text**	char		Up to 20 symbols	Text to be sent directly to the external display.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example

Binary log	
Request	01 30 30 33 3C 24 30 30 32 3F 54 65 73 74 20 74 65 78 74 20 64 69 73 70 6C 61 79 09 05 30 38 36 3D 03
Answer	01 30 30 33 35 24 30 30 32 3F 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 31 3F 03
Human oriented log	
Request	Test text display[\t]
Answer	0[\t]

The command is not used on FMP-350X and FMP-55X

4.11. Command 48 (30h) Open fiscal receipt

Request (syntax #1)

	Name	Type	Opt	Value	Description
1	**OpCode**	uint		1...30	Operator number from 1...30.
2	**OpPwd**	char		8 digits	Operator password[*1].
3	**TillNmb**	uint		1...99999	Number of point of sale.
4	**Invoice**	char		Space or “I”	If this parameter has value 'I' it opens an invoice receipt. If left blank it opens fiscal receipt.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**SlipNumber**	uint	1...9999999	Current slip number.

Example

Binary log	
Request	01 30 30 33 33 2C 30 30 33 30 31 09 31 09 32 34 09 49 09 05 30 32 3E 3F 03
Answer	01 30 30 33 39 2C 30 30 33 30 30 09 34 37 32 09 04 80 80 88 80 86 9A 80 80 05 30 36 3C 3B 03
Human oriented log	
Request	1[\t]1[\t]24[\t]I[\t]
Answer	0[\t]472[\t]

Request (syntax #2)

	Name	Type	Opt	Value	Description
1	**OpCode**	uint		1...30	Operator number from 1...30.
2	**OpPwd**	char		8 digits	Operator password[*1].
3	**NSale**	char	●	"LLDDDDDD-CCCC-DDDDDDD", L[A-Z], C[0-9A-Za-z], D[0-9]	Unique sale number.
4	**TillNmb**	uint		1...99999	Number of point of sale.
5	**Invoice**	char		Space or “I”	If this parameter has value 'I' it opens an invoice receipt. If left blank it opens fiscal receipt.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**SlipNumber**	uint	1...9999999	Current slip number.

Example

Binary log	
Request	01 30 30 34 38 32 30 30 33 30 31 09 31 09 44 54 36 33 36 35 33 33 2D 30 30 32 30 2D 30 30 31 30 31 31 30 09 31 09 49 09 05 30 37 31 30 03
Answer	01 30 30 33 39 32 30 30 33 30 30 09 34 37 33 09 04 80 80 88 80 86 9A 80 80 05 30 36 3D 32 03
Human oriented log	
Request	1[\t]1[\t]DT636533-0020-0010110[\t]1[\t]I[\t]
Answer	0[\t]473[\t]

4.12. Command 49 (31h) Registration of sale

Request (syntax #1)

	Name	Type	Opt	Value	Description
1	**PluName**	char		Up to 72 symbols	Name of product.
2	**TaxCd**	uint		1...8	**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;
3	**Price**	uint		0.01...9999999.99[*4]	Product price.
4	**Quantity**	uint	●	0.01...999999.999[*4]	Quantity of the product (default: 1.000).
5	**DiscountType**	uint	●	0...4	**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;
6	**DiscountValue**[*2]	uint	●	0.01...9999999.99 0.01...99.99[*28]	Value of discount.
7	**Department**	uint		0...99	Number of the department. If parameter **Department** = '0' - Without department.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**SlipNumber**	uint	1...9999999	Current slip number.

Example

Binary log

Request	01 30 30 34 37 36 30 30 33 31 D2 EE EF E5 ED EE 20 F1 E8 F0 E5 ED E5 09 32 09 32 2E 36 35 09 33 09 32 09 35 09 32 09 05 30 3E 3E 31 03
Answer	01 30 30 33 39 36 30 30 33 31 30 09 34 37 33 09 04 80 80 88 80 86 9A 80 80 05 30 36 3D 37 03

Human oriented log

Request	Топено сирене[\t]2[\t]2.65[\t]3[\t]2[\t]5[\t]2[\t]
Answer	0[\t]473[\t]

Request (syntax #2)					
	Name	Type	Opt	Value	Description
1	**PluName**	char		Up to 72 symbols	Name of product.
2	**TaxCd**	uint		1...8	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;
3	**Price**	uint		0.01...9999999.99[*4]	Product price.
4	**Quantity**	uint	●	0.01...999999.999[*4]	Quantity of the product (default: 1.000).
5	**DiscountType**	uint	●	0...4	Type of discount: • **'0'** or empty – no discount; • **'1'** - surcharge by percentage; • **'2'** - discount by percentage; • **'3'** - surcharge by sum; • **'4'** - discount by sum;
6	**DiscountValue**[*2]	uint	●	0.01...9999999.99 0.01...99.99[*28]	Value of discount.
7	**Department**	uint		0...99	Number of the department. If '0' - Without department;
8	**Unit**	char	●	Up to 6 symbols	Not empty string.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**SlipNumber**	uint	1...9999999	Current slip number.

Example	
Binary log	
Request	01 30 30 34 39 39 30 30 33 31 D2 EE EF E5 ED EE 20 F1 E8 F0 E5 ED E5 09 31 09 32 2E 36 35 09 33 09 09 09 32 09 E1 F0 2E 09 05 31 30 38 36 03
Answer	01 30 30 33 39 39 30 30 33 31 30 09 34 37 33 09 04 80 80 88 80 86 9A 80 80 05 30 36 3D 3A 03
Human oriented log	
Request	Топено сирене[\t]1[\t]2.65[\t]3[\t][\t][\t]2[\t]бр.[\t]
Answer	0[\t]473[\t]

4.13. Command 50 (32h) Return the active VAT rates

Request (no parameters)

	Name	Type	Opt	Value	Description

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**nZreport**	uint	1...3650	Number of first Z report.
3	**TaxA**	uint	0.00...99.99	Value of Tax group A, 100.00=disabled.
4	**TaxB**	uint	0.00...99.99	Value of Tax group B, 100.00=disabled.
5	**TaxC**	uint	0.00...99.99	Value of Tax group C, 100.00=disabled.
6	**TaxD**	uint	0.00...99.99	Value of Tax group D, 100.00=disabled.
7	**TaxE**	uint	0.00...99.99	Value of Tax group E, 100.00=disabled.
8	**TaxF**	uint	0.00...99.99	Value of Tax group F, 100.00=disabled.
9	**TaxG**	uint	0.00...99.99	Value of Tax group G, 100.00=disabled.
10	**TaxH**	uint	0.00...99.99	Value of Tax group H, 100.00=disabled.
11	**EndDate**	char	See remark: [*27]	Date of entry.

Example

Binary log

Request	01 30 30 32 3A 3A 30 30 33 32 05 30 31 3D 30 03
Answer	01 30 30 37 34 3A 30 30 33 32 30 09 31 09 30 30 2E 30 30 09 32 30 2E 30 30 09 32 30 2E 30 30 09 30 39 2E 30 30 09 31 30 30 2E 30 30 09 31 30 30 2E 30 30 09 31 30 30 2E 30 30 09 31 30 30 2E 30 30 09 30 31 2D 30 31 2D 30 30 09 04 80 80 88 80 86 9A 80 80 05 31 30 37 3C 03

Human oriented log

Request	
Answer	0[\t]1[\t]00.00[\t]20.00[\t]20.00[\t]09.00[\t]100.00[\t]100.00[\t]100.00[\t]100.00[\t]01-01-00[\t]

4.14. Command 51 (33h) Subtotal

Request

	Name	Type	Opt	Value	Description
1	**Print**	uint	●	0...1	Print out: • **'0'** - default, no print out; • **'1'** - the sum of the subtotal will be printed out;
2	**Display**[*3]	uint	●	0...1	Show the subtotal on the external client display: • **'0'** - default, no print out; • **'1'** - the sum of the subtotal will be printed out;
3	**DiscountType**	uint	●	0...4	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.
4	**DiscountValue**[*2]	uint	●	0.01..9999999.99 0.01..99.99[*28]	Value of discount.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**SlipNumber**	uint	1...9999999	Current slip number.
3	**Subtotal**	uint	0.00...9999999.99 or 0...999999999[*7]	Subtotal of the receipt.
4	**TaxA**	uint	0.00...9999999.99 or 0...999999999[*7]	Receipts turnover by vat A.
5	**TaxB**	uint	0.00...9999999.99 or 0...999999999[*7]	Receipts turnover by vat B.
6	**TaxC**	uint	0.00...9999999.99 or 0...999999999[*7]	Receipts turnover by vat C.
7	**TaxD**	uint	0.00...9999999.99 or 0...999999999[*7]	Receipts turnover by vat D.
8	**TaxE**	uint	0.00...9999999.99 or 0...999999999[*7]	Receipts turnover by vat E.
9	**TaxF**	uint	0.00...9999999.99 or 0...999999999[*7]	Receipts turnover by vat F.
10	**TaxG**	uint	0.00...9999999.99 or 0...999999999[*7]	Receipts turnover by vat G.
11	**TaxH**	uint	0.00...9999999.99 or 0...999999999[*7]	Receipts turnover by vat H.

Example

Binary log	
Request	01 30 30 33 32 3B 30 30 33 33 31 09 09 32 09 31 30 09 05 30 32 3B 33 03
Answer	01 30 30 36 39 3B 30 30 33 33 30 09 34 37 33 09 33 35 2E 37 37 09 32 31 2E 34 36 09 31 34 2E 33 31 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 04 80 80 88 80 86 9A 80 80 05 30 3E 39 3C 03
Human oriented log	
Request	1[\t][\t]2[\t]10[\t]
Answer	0[\t]473[\t]35.77[\t]21.46[\t]14.31[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]

4.15. Command 53 (35h) Payments and calculation of the total sum (TOTAL)

4.15.1. Standard payment types

Request					
	Name	Type	Opt	Value	Description
1	**PaidMode**	uint		0...5	Type of payment: • **'0'** – cash; • **'1'** – credit card; • **'2'** – debit card; • **'3'** – other pay#3; • **'4'** – other pay#4; • **'5'** – other pay#5; See remark: [*13]
2	**Amount**	uint		0.00...9999999.99 or 0...999999999[*7]	Amount to pay.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**Status**	char	'D' or 'R'	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.
3	**Amount**	uint	0.00...9999999.99 or 0...999999999[*7]	The sum tendered.

Example	
Binary log	
Request	01 30 30 32 3F EF 30 30 33 35 30 09 31 30 09 05 30 33 33 30 03
Answer	01 30 30 33 3C EF 30 30 33 35 30 09 52 09 35 2E 30 39 09 04 80 80 88 80 86 9A 80 80 05 30 38 32 30 03
Human oriented log	
Request	0[\t]10[\t]
Answer	0[\t]R[\t]5.09[\t]

4.15.2. Payment type – Card with pinpad

Request (syntax 1)

	Name	Type	Opt	Value	Description
1	**PaidMode**	uint		2	Type of payment: • **'2'** – debit card; See remark: [*13]
2	**Amount**	uint		0.00...9999999.99 or 0...999999999[*7]	Amount to pay.
3	**Type**	uint	●	1 or 12	Type of card payment (only with PinPad connected and PaidMode=2): • **'1'** – with money; • **'12' –** with points from loyal scheme;

Answer when Type=1 or Type=12 and PaidMode=2 (For payment with pinpad when transaction may be successful in pinpad, but unsuccessful in fiscal device)

	Name	Type	Value	Description
1	**ErrorCode**	int	-111560	Indicates an error code.
2	**Sum**	uint	0...999999999	Sum from last transaction in cents.
3	**CardNum**	uint	0000...9999	Last digits from card number.

Answer when Type=1 or Type=12 and PaidMode=2 (For payment with pinpad when error from pinpad occured)

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example

Binary log

Request	01 30 30 33 30 FC 30 30 33 35 36 09 36 09 30 09 05 30 33 34 33 03
Answer	01 30 30 33 3C FC 30 30 33 35 30 09 52 09 35 2E 39 35 09 04 80 80 88 80 86 9A 80 80 05 30 38 33 32 03

Human oriented log

Request	2[\t]2[\t]
Answer	-111555[\t]

4.15.3. Payment type – foreign currency

Request

	Name	Type	Opt	Value	Description
1	**PaidMode**	uint		6	Type of payment – Foreign currency.
2	**Amount**	uint		0.00...9999999.99 or 0...999999999[*7]	Amount to pay.
3	**Change**	uint		0...1	Change: • **'0'** - current currency; • **'1'** - foreign currency;

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**Status**	char	'D' or 'R'	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.
3	**Amount**	uint	0.00...9999999.99 or 0...999999999[*7]	The sum tendered.

Example

Binary log	
Request	01 30 30 33 30 FC 30 30 33 35 36 09 36 09 30 09 05 30 33 34 33 03
Answer	01 30 30 33 3C FC 30 30 33 35 30 09 52 09 35 2E 39 35 09 04 80 80 88 80 86 9A 80 80 05 30 38 33 32 03
Human oriented log	
Request	6[\t]6[\t]0[\t]
Answer	0[\t]R[\t]5.95[\t]

4.15.4. Payment type – Card with pinpad and returning data for transaction

Request					
	Name	Type	Opt	Value	Description
1	**PaidMode**	uint		12	Type of payment: • **'12'** – payment with pinpad and returning data for transaction(if pinpad is configured);
2	**Amount**	uint		0.00...9999999.99 or 0...999999999[*7]	Amount to pay.
3	**Type**	uint	●	1 or 12	Type of card payment (with PinPad connected). Only for payment with debit card: • **'1'** - with money; • **'12' -** with points from loyal scheme;

Answer – successful operation				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**Status**	char	'D' or 'R'	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.
3	**Amount**	uint	0.00...9999999.99 or 0...999999999[*7]	The sum tendered.
4	**AC**	char	Up to 64	Authorization code for transaction;
5	**CardData**	uint	-1...3	Type of card payment: • '-1' - unknown; • '0' - chip; • '1' - contactless • '2' - magnetic stripe; • '3' - manually
6	**CardNumber**	char	Up to 64	Card number;
7	**MIDNumber**	char	Up to 64	Merchant ID;
8	**RRN**	char	Up to 64	RRN number for transaction;
9	**TIDNumber**	char	Up to 64	Terminal ID;
10	**TransAmount**	uint	0.00...9999999.99	Transaction amount;
11	**TransDate**	char	DD.MM.YY	Transaction date;
12	**TransTime**	char	hh:mm:ss	Transaction time;
13	**TransNumber**	uint	0...999999999	Transaction number;
14	**TransStatus**	uint	0...2	Transaction status: • '0' - approved; • '1' - declined; • '2' - error;
15	**TransType**	uint	0...999999999	Transaction type;
16	**FullResponseCode**	uint	0...999999999	Complete response code;

Example	
Binary log	
Request	01 30 30 33 34 4A 30 30 33 35 31 32 09 30 2E 30 31 09 31 09 05 30 33 34 3C 03
Answer	01 30 30 39 3F 4A 30 30 33 35 30 09 44 09 30 2E 30 30 09 30 31 38 33 37 31 09 31 09 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 32 33 34 30 09 39 39 39 33 39 30 30 30 30 31 30 30 30 30 30 09 30 30 30 30 30 36 37 31 34 31 34 34 09 39 33 38 30 30 31 33 32 09 30 2E 30 31 09 31 37 2E 31 32 2E 31 39 09 31 36 3A 32 34 3A 30 37 09 36 36 36 30 33 09 30 09 31 09 30 09 04 80 80 88 81 86 9A 80 80 05 31 38 37 3F 03
Human oriented log	
Request	12[\t]0.01[\t]1[\t]
Answer	0[\t]D[\t]0.00[\t]018371[\t]1[\t]**********2340[\t]999390000100000[\t]000006714144[\t]93800132[\t]0.01[\t]17.12.19[\t]16:24:07[\t]66603[\t]0[\t]1[\t]0[\t]

Answer – unsuccessful operation (when transaction may be successful in pinpad, but unsuccessful in fiscal device)

	Name	Type	Value	Description
1	**ErrorCode**	int	-111560	Indicates an error code.
2	**Sum**	uint	0...999999999	Sum from last transaction in cents.
3	**CardNum**	uint	0000...9999	Last digits from card number.

Answer – unsuccessful operation (when error from pinpad occurred)

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

4.16. Command 54 (36h) Printing of a free fiscal text

Request

	Name	Type	Opt	Value	Description
1	**Text**	char		Up to XX symbols	XX depends on print columns. • XX = 42,48,64[*11] • XX = 42[*8] • XX = 32[*12]
2	**Bold**	uint	●	0...1	• empty field – normal text; • **'1'** - print bold text;
3	**Italic**	uint	●	0...1	• empty field – normal text; • **'1'** - print italic text;
4	**DoubleH**	uint	●	0...2	• empty field – normal height text; • **'0'** - normal height; • **'1'** - double height; • **'2'** - half height;
5	**Underline**	uint	●	0...1	• empty field – normal text; • **'1'** - print underlined text;
6	**alignment**	uint	●	0...2	• empty field – left alignment; • **'0'** - left alignment; • **'1'** - center; • **'2'** - right;

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example

Binary log

Request	01 30 30 33 39 2D 30 30 33 36 54 65 78 74 09 30 09 30 09 30 09 30 09 30 09 05 30 34 39 32 03
Answer	01 30 30 33 35 2D 30 30 33 36 30 09 04 80 80 88 80 86 9A 80 80 05 30 36 32 38 03

Human oriented log

Request	Text[\t]0[\t]0[\t]0[\t]0[\t]0[\t]
Answer	0[\t]

4.17. Command 55 (37h) Pinpad commands

4.17.1. Pinpad commands – option '1'-Void

Request (Syntax if pinpad is configured for Borica)

	Name	Type	Opt	Value	Description
1	**Option**	uint		1	Operation type.
2	**PayType**	uint		7 or 13	Type of payment: • **'7'** - Return with money; • **'13'** - Return with points from loyal scheme;
3	**Amount**	uint		0.00...9999999.99 or 0...999999999[*7]	The amount of the transaction.
4	**RRN**	char		Up to 12 digits	RRN of the transaction.
5	**AC**	char		Up to 6 digits	AC of the transaction.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example

Binary log	
Request	01 30 30 34 35 2B 30 30 33 37 31 09 37 09 30 2E 30 31 09 36 36 39 37 37 38 39 33 35 36 09 39 37 38 33 34 39 09 05 30 36 38 31 03
Answer	01 30 30 33 35 2B 30 30 33 37 30 09 04 80 80 80 80 80 82 80 80 05 30 36 30 31 03
Human oriented log	
Request	1[\t]7[\t]0.01[\t]6697789356[\t]978349[\t]
Answer	0[\t]

Request (Syntax if pinpad is configured for UBB)

	Name	Type	Opt	Value	Description
1	**Option**	uint		1	Operation type.
2	**PayType**	uint		16 or 17	Type of payment: • **'16'** - Return with AC number; • **'17'** - Return with receipt;
3	**Amount**	uint		0.00...9999999.99 or 0...999999999[*7]	The amount of the transaction.
4	**Number**	char		Up to 6 digits (if PayType=16) Up to 7 digits (if PayType=17)	Depends on PayType (16 – AC number, 17 – receipt number).

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Request (Syntax if pinpad is configured for DSK)

	Name	Type	Opt	Value	Description
1	**Option**	uint		1	Operation type.
2	**PayType**	uint		16	Type of payment: • **'16'** - Return with money;
3	**Amount**	char		0.00...9999999.99 or 0...999999999[*7]	The amount of the transaction.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Request (Syntax if pinpad is configured for DSK)					
	Name	Type	Opt	Value	Description
1	**Option**	uint		1	Operation type.
2	**PayType**	uint		17	Type of payment: • **'17'** - Void last document.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

4.17.2. Pinpad commands – option '2'-Copy of last document

Request					
	Name	Type	Opt	Value	Description
1	**Option**	uint		2	Operation type.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3C 42 30 30 33 37 32 09 05 30 32 31 3A 03
Answer	01 30 30 33 35 42 30 30 33 37 30 09 04 80 80 80 80 80 82 80 80 05 30 36 31 38 03
Human oriented log	
Request	2[\t]
Answer	0[\t]

4.17.3. Pinpad commands – option '3'-Copy of document by type

Request					
	Name	Type	Opt	Value	Description
1	**Option**	uint		3	Operation type.
2	**Type**	uint		1...3	Type: • **'1'** - RRN • **'2'** - AC • **'3'** - Number of the transaction.
3	**Number**	char		Up to 12 symbols	Depends on Type: • RRN – 12 digits max • AC – 6 digits max • Number – 6 digits max.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 33 3B 2D 30 30 33 37 33 09 31 09 30 30 30 30 30 36 37 30 36 37 35 32 09 05 30 34 3A 3A 03
Answer	01 30 30 33 35 2D 30 30 33 37 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 32 31 03
Human oriented log	
Request	3[\t]1[\t]000006706752[\t]
Answer	0[\t]

4.17.4. Pinpad commands – option '4'-Copy of all documents

Request					
	Name	Type	Opt	Value	Description
1	**Option**	uint		4	Operation type.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3C 3B 30 30 33 37 34 09 05 30 32 31 35 03
Answer	01 30 30 33 35 3B 30 30 33 37 30 09 04 80 80 80 80 80 82 80 80 05 30 36 31 31 03
Human oriented log	
Request	4[\t]
Answer	0[\t]

4.17.5. Pinpad commands – option ‘5’-End of day from pinpad

Request					
	Name	Type	Opt	Value	Description
1	**Option**	uint		5	Operation type.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3C 3C 30 30 33 37 35 09 05 30 32 31 37 03
Answer	01 30 30 33 35 3C 30 30 33 37 30 09 04 80 80 80 80 80 82 80 80 05 30 36 31 32 03
Human oriented log	
Request	5[\t]
Answer	0[\t]

4.17.6. Pinpad commands – option ‘6’-Report from pinpad

Request					
	Name	Type	Opt	Value	Description
1	**Option**	uint		6	Operation type.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3C 3D 30 30 33 37 36 09 05 30 32 31 39 03
Answer	01 30 30 33 35 3D 30 30 33 37 30 09 04 80 80 80 80 80 82 80 80 05 30 36 31 33 03
Human oriented log	
Request	6[\t]
Answer	0[\t]

4.17.7. Pinpad commands – option '7'-Full report from pinpad

Request					
	Name	Type	Opt	Value	Description
1	**Option**	uint		7	Operation type.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3C 43 30 30 33 37 37 09 05 30 32 32 30 03
Answer	01 30 30 33 35 43 30 30 33 37 30 09 04 80 80 80 80 80 82 80 80 05 30 36 31 39 03
Human oriented log	
Request	7[\t]
Answer	0[\t]

4.17.8. Pinpad commands – option '8'-Enter date and time for pinpad

Request					
	Name	Type	Opt	Value	Description
1	**Option**	uint		8	Operation type.
2	**DateTime**	char		See remark: [*27]	Date and time.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 34 32 45 30 30 33 37 38 09 31 33 2D 30 35 2D 31 39 20 31 36 3A 33 31 3A 30 30 20 44 53 54 09 05 30 36 37 3B 03
Answer	01 30 30 33 35 45 30 30 33 37 30 09 04 80 80 80 80 80 82 80 80 05 30 36 31 3B 03
Human oriented log	
Request	8[\t]13-05-19 16:39:00 DST[\t]
Answer	0[\t]

4.17.9. Pinpad commands – option ‘9’-Check connection with pinpad

Request					
	Name	Type	Opt	Value	Description
1	**Option**	uint		9	Operation type.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3C 46 30 30 33 37 39 09 05 30 32 32 35 03
Answer	01 30 30 33 35 46 30 30 33 37 30 09 04 80 80 80 80 80 82 80 80 05 30 36 31 3C 03
Human oriented log	
Request	9[\t]
Answer	0[\t]

4.17.10. Pinpad commands – option ‘10’-Check connection with server

Request					
	Name	Type	Opt	Value	Description
1	**Option**	uint		10	Operation type.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3D 47 30 30 33 37 31 30 09 05 30 32 34 3F 03
Answer	01 30 30 33 35 47 30 30 33 37 30 09 04 80 80 80 80 80 82 80 80 05 30 36 31 3D 03
Human oriented log	
Request	10[\t]
Answer	0[\t]

4.17.11. Pinpad commands – option ‘11’-Loyalty balance

Request					
	Name	Type	Opt	Value	Description
1	**Option**	uint		11	Operation type.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

4.17.12. Pinpad commands – option '12'-Get update

Request					
	Name	Type	Opt	Value	Description
1	**Option**	uint		12	Operation type.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3D 49 30 30 33 37 31 32 09 05 30 32 35 33 03
Answer	01 30 30 33 35 49 30 30 33 37 30 09 04 80 80 80 80 80 82 80 80 05 30 36 31 3F 03
Human oriented log	
Request	12[\t]
Answer	0[\t]

4.17.13. Pinpad commands – option '13'-After errors by CMD53 or CMD55 (opt 14)

Request (Used when command 53(paying with pinpad) and command 55 (option 14) returns error along with sum and last digits of card number)					
	Name	Type	Opt	Value	Description
1	**Option**	uint		13	Operation type.
2	**Operation**	uint		1...2	Operation for execution; • **'1'** - Print receipt; • **'2'** - Void transaction from pinpad;

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3F 5C 30 30 33 37 31 33 09 31 09 05 30 32 3A 33 03
Answer	01 30 30 33 35 5C 30 30 33 37 30 09 04 80 80 88 80 86 9A 80 80 05 30 36 35 38 03
Human oriented log	
Request	13[\t]1[\t]
Answer	0[\t]

4.17.14. Pinpad commands – option '14'-Make sale from pinpad, without fiscal receipt

Request (Make sale from pinpad, without fiscal receipt)

	Name	Type	Opt	Value	Description
1	**Option**	uint		14	Operation type.
2	**Amount**	uint		0.00...9999999.99 or 0...999999999[*7]	Amount for sale.

Answer (When command passed)

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**AC**	char	Up to 6 symbols	Authorization code for transaction.
3	**CardData**	int	-1...3	Type of card payment: • **'-1'** – unknown; • **'0'** – chip; • **'1'** – contactless; • **'2'** - magnetic stripe; • **'3'** - manually;
4	**CardNumber**	char	Up to 16 symbols	Card number.
5	**MIDNumber**	char	Up to 16 symbols	Merchant ID.
6	**RRN**	char	Up to 12 symbols	RRN number for transaction.
7	**TIDNumber**	char	Up to 8 symbols	Terminal ID.
8	**TransAmount**	uint	0.00...9999999.99 or 0...999999999[*7]	Transaction amount.
9	**TransDate**	char	See remark: [*27]	Transaction date.
10	**TransTime**	char	See remark: [*27]	Transaction time.
11	**TransNumber**	uint	0...999999	Transaction number.
12	**TransStatus**	uint	0...2	Transaction status: • **'0'** – approved; • **'1'** – declined; • **'2'** – error;
13	**TransType**	uint	1	Transaction type.
14	**FullResponseCode**	uint	0...999999	Complete response code.

Example

Binary log

Request	01 30 30 33 32 4A 30 30 33 37 31 34 09 30 2E 30 32 09 05 30 33 31 35 03
Answer	01 30 30 39 35 4A 30 30 33 37 30 09 09 39 37 38 34 37 32 09 31 09 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 38 39 30 37 09 39 39 39 33 39 30 30 30 30 31 30 30 30 30 30 09 30 30 30 30 30 36 36 39 37 39 31 32 09 39 33 38 30 30 32 34 31 09 30 2E 30 32 09 31 33 2E 30 35 2E 31 39 09 31 37 3A 31 32 3A 30 30 09 38 38 09 30 09 31 09 30 04 80 80 80 80 80 82 80 80 05 31 36 3C 3B 03

Human oriented log

Request	14[\t]0.02[\t]
Answer	0[\t]978472[\t]1[\t]**********8907[\t]999390000100000[\t]000006697912[\t]93800241[\t]0.02[\t]13.05.19[\t]17:12:00[\t]88[\t]0[\t]1[\t]0[\t]

Answer (When command did not pass and the error is from pinpad)				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Answer (When transaction may be successful in pinpad, but unsuccessful in fiscal device)				
	Name	Type	Value	Description
1	**ErrorCode**	int	-111560	Indicates an error code.
2	**Sum**	uint	0...999999999	Sum from last transaction in cents.
3	**CardNum**	uint	0000...9999	Last digits from card number.

4.17.15. Pinpad commands – option '15'-Print receipt for pinpad after successful transaction

Request (Print receipt for pinpad after successful transaction. Must be executed after command 53(when paying with pinpad) and after command 56(when paying with pinpad))					
	Name	Type	Opt	Value	Description
1	**Option**	char		15	Operation type.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3D 25 30 30 33 37 31 35 09 05 30 32 33 32 03
Answer	01 30 30 33 35 25 30 30 33 37 30 09 04 80 80 80 80 80 82 80 80 05 30 35 3F 3B 03
Human oriented log	
Request	15[\t]
Answer	0[\t]

4.18. Command 56 (38h) Close fiscal receipt

Request (no parameters)					
	Name	Type	Opt	Value	Description

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**SlipNumber**	uint	1...9999999	Current slip number.

Example	
Binary log	
Request	01 30 30 32 3A 29 30 30 33 38 05 30 31 3C 35 03
Answer	01 30 30 33 39 29 30 30 33 38 30 09 32 36 39 09 04 80 80 80 80 80 82 80 80 05 30 36 3A 3E 03
Human oriented log	
Request	
Answer	0[\t]269[\t]

4.19. Command 57 (39h) Enter and print invoice data

Request

	Name	Type	Opt	Value	Description
1	**Seller**	char	●	Up to 36 symbols	Name of the client. If left blank prints empty space for hand-writing.
2	**Receiver**	char	●	Up to 36 symbols	Name of the receiver. If left blank prints empty space for hand-writing.
3	**Buyer**	char	●	Up to 36 symbols	Name of the buyer. If left blank prints empty space for hand-writing.
4	**Address1**	char	●	Up to 36 symbols	First line of the address. If left blank prints empty space for hand-writing.
5	**Address2**	char	●	Up to 36 symbols	Second line of the address. If left blank prints empty space for hand-writing.
6	**TypeTAXN**	uint		0...3	Type of client's tax number: • **'0'** – BULSTAT; • **'1'** – EGN; • **'2'** – Personal ID; • **'3'** – service number;
7	**TAXN**	char		8...13 symbols	Client's tax number.
8	**VATN**	char		10...14 symbols	VAT number of the client.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example

Binary log

Request	01 30 30 38 39 34 30 30 33 39 C8 E2 E0 ED 20 C8 E2 E0 ED EE E2 09 C8 E2 E0 ED 20 C8 E2 E0 ED EE E2 09 C4 E5 F2 F1 EA E0 20 EA F3 F5 ED FF 20 2D 20 D6 E5 ED F2 FA F0 09 E3 F0 2E D1 EE F4 E8 FF 2C 20 F3 EB 2E 20 D6 E5 ED F2 F0 E0 EB ED E0 09 09 30 09 30 30 30 37 31 33 33 39 31 09 52 4F 30 30 30 37 31 33 33 39 31 09 05 33 39 39 35 03
Answer	01 30 30 33 39 35 30 30 33 39 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 3C 3C 03

Human oriented log

Request	Иван Иванов[\t]Иван Иванов[\t]Детска кухня - Център[\t]гр.София, ул. Централна[\t][\t]0[\t]000713391[\t]RO000713391[\t]
Answer	0[\t]

4.20. Command 58 (3Ah) Registering the sale of a programmed item

Request

	Name	Type	Opt	Value	Description
1	**PluCode**	uint		1...100000[*8] 1...3000[*9]	The code of the item.
2	**Quantity**[*4]	uint	●	Up to 999999.999	Quantity of the product (default: 1.000).
3	**Price**[*4]	char	●	0.00...9999999.99 or 0...999999999[*7]	Product price. Format: 2 decimals.
4	**DiscountType**	uint	●	0...4	Type of discount: • **'0'** or empty – no discount; • **'1'** – surcharge by percentage; • **'2'** – discount by percentage; • **'3'** – surcharge by sum; • **'4'** – discount by sum;
5	**DiscountValue**[*5]	uint	●	0.01...9999999.99 0.01...100.00[*28]	Value of discount.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**SlipNumber**	uint	1...9999999	Current slip number.

Example

Binary log

Request	01 30 30 33 34 42 30 30 33 3A 34 09 35 09 09 32 09 31 30 09 05 30 33 30 34 03
Answer	01 30 30 33 39 42 30 30 33 3A 30 09 35 30 31 09 04 80 80 88 80 86 9A 80 80 05 30 36 3E 34 03

Human oriented log

Request	4[\t]5[\t][\t]2[\t]10[\t]
Answer	0[\t]501[\t]

4.21. Command 60 (3Ch) Cancel fiscal receipt

Request (no parameters)					
	Name	Type	Opt	Value	Description

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3A 43 30 30 33 3C 05 30 31 3E 33 03
Answer	01 30 30 33 35 43 30 30 33 3C 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 33 3C 03
Human oriented log	
Request	
Answer	0[\t]

4.22. Command 61 (3Dh) Set date and time

Request

	Name	Type	Opt	Value	Description
1	**DateTime**	char		See remark: [*27]	Set date and time.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example

Binary log	
Request	01 30 30 34 30 44 30 30 33 3D 31 34 2D 30 35 2D 31 39 20 31 31 3A 31 38 3A 30 30 20 44 53 54 09 05 30 36 33 3E 03
Answer	01 30 30 33 35 44 30 30 33 3D 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 33 3E 03
Human oriented log	
Request	14-05-19 11:18:00 DST[\t]
Answer	0[\t]

4.23. Command 62 (3Eh) Read date and time

Request (no parameters)					
	Name	Type	Opt	Value	Description

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**DateTime**	char	See remark: [*27]	Read date and time.

Example	
Binary log	
Request	01 30 30 32 3A 45 30 30 33 3E 05 30 31 3E 37 03
Answer	01 30 30 34 3B 45 30 30 33 3E 30 09 31 34 2D 30 35 2D 31 39 20 31 31 3A 33 32 3A 31 33 20 44 53 54 09 04 80 80 80 80 86 9A 80 80 05 30 3A 3A 38 03
Human oriented log	
Request	
Answer	0[\t]14-05-19 11:32:13 DST[\t]

4.24. Command 63 (3Fh) Show current date and time on the external display

Request (no parameters)					
	Name	Type	Opt	Value	Description

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**DateTime**	char	See remark: [*27]	Date and time.

Example	
Binary log	
Request	01 30 30 32 3A 46 30 30 33 3F 05 30 31 3E 39 03
Answer	01 30 30 34 3B 46 30 30 33 3F 30 09 31 34 2D 30 35 2D 31 39 20 31 31 3A 35 31 3A 30 37 20 44 53 54 09 04 80 80 80 80 86 9A 80 80 05 30 3A 3A 3E 03
Human oriented log	
Request	
Answer	0[\t]14-05-19 11:51:07 DST[\t]

The command is not used on FMP-350X and FMP-55X….

4.25. Command 64 (40h) Information on the last fiscal entry

Request

	Name	Type	Opt	Value	Description
1	**Type**	uint	●	0...3	Type of returned data. Default: 0; • **'0'** – Turnover on TAX group; • **'1'** – Amount on TAX group; • **'2'** – Storno turnover on TAX group; • **'3'** – Storno amount on TAX group;

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**nRep**	uint	1...3650	Number of report.
3	**SumA**	uint	0.00...9999999.99 or 0...999999999[*7]	Depends on **Type**.
4	**SumB**	uint	0.00...9999999.99 or 0...999999999[*7]	Depends on **Type**.
5	**SumC**	uint	0.00...9999999.99 or 0...999999999[*7]	Depends on **Type**.
6	**SumD**	uint	0.00...9999999.99 or 0...999999999[*7]	Depends on **Type**.
7	**SumE**	uint	0.00...9999999.99 or 0...999999999[*7]	Depends on **Type**.
8	**SumF**	uint	0.00...9999999.99 or 0...999999999[*7]	Depends on **Type**.
9	**SumG**	uint	0.00...9999999.99 or 0...999999999[*7]	Depends on **Type**.
10	**SumH**	uint	0.00...9999999.99 or 0...999999999[*7]	Depends on **Type**.
11	**Date**	char	See remark: [*27]	Date of fiscal record.

Example

Binary log	
Request	01 30 30 32 3C 47 30 30 34 30 30 09 05 30 32 31 37 03
Answer	01 30 30 36 39 47 30 30 34 30 30 09 36 09 30 2E 30 30 09 32 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 38 2D 30 35 2D 31 39 09 04 80 80 80 80 86 9A 80 80 05 30 3E 37 3F 03
Human oriented log	
Request	0[\t]
Answer	0[\t]6[\t]0.00[\t]20.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]08-05-19[\t]

4.26. Command 65 (41h) Information on daily taxation

Request

	Name	Type	Opt	Value	Description
1	**Type**	uint	●	0...3	Type of returned data. Default: 0; • **'0'** – Turnover on TAX group; • **'1'** – Amount on TAX group; • **'2'** – Storno turnover on TAX group; • **'3'** – Storno amount on TAX group;

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**nRep**	uint	1...3650	Number of report.
3	**SumA**	uint	0.00...9999999.99 or 0...999999999[*7]	Depends on **Type**.
4	**SumB**	uint	0.00...9999999.99 or 0...999999999[*7]	Depends on **Type**.
5	**SumC**	uint	0.00...9999999.99 or 0...999999999[*7]	Depends on **Type**.
6	**SumD**	uint	0.00...9999999.99 or 0...999999999[*7]	Depends on **Type**.
7	**SumE**	uint	0.00...9999999.99 or 0...999999999[*7]	Depends on **Type**.
8	**SumF**	uint	0.00...9999999.99 or 0...999999999[*7]	Depends on **Type**.
9	**SumG**	uint	0.00...9999999.99 or 0...999999999[*7]	Depends on **Type**.
10	**SumH**	uint	0.00...9999999.99 or 0...999999999[*7]	Depends on **Type**.

Example

Binary log	
Request	01 30 30 32 3C 4B 30 30 34 31 30 09 05 30 32 31 3C 03
Answer	01 30 30 36 32 4B 30 30 34 31 30 09 37 09 32 32 2E 34 30 09 31 32 37 2E 32 32 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 04 80 80 80 80 86 9A 80 80 05 30 3D 35 38 03
Human oriented log	
Request	0[\t]
Answer	0[\t]7[\t]22.40[\t]127.22[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]

4.27. Command 66 (42h) Set invoice interval

Request (syntax #1)

	Name	Type	Opt	Value	Description
1	**End**	uint		1...9999999999	If the current invoice counter didn't reached the end of the interval.

Request (syntax #2)

	Name	Type	Opt	Value	Description
1	**Start**	uint		1...9999999999	The starting number of the interval.
2	**End**	uint		1...9999999999	The ending number of the interval.

Request (syntax #3 read current values, no parameters)

	Name	Type	Opt	Value	Description

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**Start**	uint	1...9999999999	The current starting value of the interval.
3	**End**	uint	1...9999999999	The current ending value of the interval.
4	**Current**	uint	1...9999999999	The current invoice receipt number.

Example

Binary log	
Request	01 30 30 32 3A 50 30 30 34 32 05 30 31 3E 37 03
Answer	01 30 30 35 30 50 30 30 34 32 30 09 31 30 30 30 30 32 35 09 31 30 30 30 30 30 30 30 35 37 09 31 30 30 30 30 32 39 09 04 80 80 80 80 86 9A 80 80 05 30 3A 3F 39 03
Human oriented log	
Request	
Answer	0[\t]1000025[\t]1000000057[\t]1000029[\t]

4.28. Command 68 (44h) Number of remaining entries for Z-reports in FM

Request (no parameters)					
	Name	Type	Opt	Value	Description

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**ReportsLeft**	uint	1...3650	The number of remaining entries for Z-reports in FM.

Example	
Binary log	
Request	01 30 30 32 3A A5 30 30 34 34 05 30 32 33 3E 03
Answer	01 30 30 33 3A A5 30 30 34 34 30 09 33 35 38 31 09 04 80 80 88 80 86 9A 80 80 05 30 37 37 3E 03
Human oriented log	
Request	
Answer	0[\t]3644[\t]

4.29. Command 69 (45h) Reports

4.29.1. Report X and report Z

Request

	Name	Type	Opt	Value	Description
1	**ReportType**	char		X or Z	Report type: • **'X'** – X-report; • **'Z'** – Z-report;

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**nRep**	uint	1...3650	Number of report.
3	**TotA**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum accumulated by TAX group A – sell operations.
4	**TotB**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum accumulated by TAX group B – sell operations.
5	**TotC**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum accumulated by TAX group C – sell operations.
6	**TotD**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum accumulated by TAX group D – sell operations.
7	**TotE**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum accumulated by TAX group E – sell operations.
8	**TotF**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum accumulated by TAX group F – sell operations.
9	**TotG**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum accumulated by TAX group G – sell operations.
10	**TotH**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum accumulated by TAX group H – sell operations.
11	**StorA**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum accumulated by TAX group A – storno operations.
12	**StorB**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum accumulated by TAX group B – storno operations.
13	**StorC**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum accumulated by TAX group C – storno operations.
14	**StorD**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum accumulated by TAX group D – storno operations.
15	**StorE**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum accumulated by TAX group E – storno operations.
16	**StorF**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum accumulated by TAX group F – storno operations.
17	**StorG**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum accumulated by TAX group G – storno operations.
18	**StorH**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum accumulated by TAX group H – storno operations.

Example	
Binary log	
Request	01 30 30 32 3C 53 30 30 34 35 5A 09 05 30 32 35 32 03
Answer	01 30 30 38 3A 53 30 30 34 35 30 09 37 09 32 32 2E 34 30 09 31 32 37 2E 32 32 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 04 80 80 80 80 86 9A 80 80 05 31 33 3A 36 03
Human oriented log	
Request	Z[\t]
Answer	0[\t]7[\t]22.40[\t]127.22[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]

4.29.2. Report D, report G

Request					
	Name	Type	Opt	Value	Description
1	**ReportType**	char		D or G	Report type: • **'D'** – Departments report; • **'G'** – Item groups report;

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3C 54 30 30 34 35 44 09 05 30 32 33 3D 03
Answer	01 30 30 33 35 54 30 30 34 35 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 34 37 03
Human oriented log	
Request	D[\t]
Answer	0[\t]

4.29.3. Report P – print the periodical report

Request					
	Name	Type	Opt	Value	Description
1	**ReportType**	char		P	Report type: • **'P' – Periodical** report;
2	**ReportSubType**	uint		1...3	Report sub-type: • **1 - by payments;** • **2 - by departments;** • **3 - by items;**
2	**StartDate**	char	●	See remark: [*27]	Default: Date of fiscalization
3	**EndDate**	char	●	See remark: [*27]	Default: Current date

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 34 30 25 30 30 34 35 50 09 31 09 30 31 2D 30 31 2D 32 30 09 31 33 2D 30 32 2D 32 30 09 05 30 35 35 3C 03
Answer	01 30 30 33 35 25 30 30 34 35 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 31 38 03
Human oriented log	
Request	P[\t]1[\t]01-01-20[\t]13-02-20[\t]
Answer	0[\t]

4.30. Command 70 (46h) Cash in and Cash out operations

Request					
	Name	Type	Opt	Value	Description
1	**Type**	uint		0...3	Type of operation: • **'0'** – cash in; • **'1'** – cash out; • **'2'** – cash in - (foreign currency); • **'3'** – cash out - (foreign currency);
2	**Amount**	uint		0.00...9999999.99 or 0...999999999[*7]	If Amount is 0, only answer is returned, and receipt is not printed.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**CashSum**	uint	0.00...9999999.99 or 0...999999999[*7]	Cash in safe sum.
3	**CashIn**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum of cash in operations.
4	**CashOut**	uint	0.00...9999999.99 or 0...999999999[*7]	Total sum of cash out operations.

Example	
Binary log	
Request	01 30 30 32 3F 25 30 30 34 36 30 09 35 30 09 05 30 32 36 3C 03
Answer	01 30 30 34 3D 25 30 30 34 36 30 09 35 39 39 2E 35 39 09 31 30 35 30 2E 30 30 09 2D 31 30 30 30

	2E 30 30 09 04 80 80 80 80 86 9A 80 80 05 30 3A 35 30 03
Human oriented log	
Request	0[\t]50[\t]
Answer	0[\t]599.59[\t\1050.00[\t]-1000.00[\t]

4.31. Command 71 (47h) General information, modem test

4.31.1. General information, modem test

Request

	Name	Type	Opt	Value	Description
1	**InfoType**	uint	●	0, 1, 3, 4, 6, 9, 10, 11	Type of the information printed. Default: 0; • **'0'** – general diagnostic information about the device; • **'1'** – test of the modem with connection to the NRA server; • **'3'** – print information about the connection with NRA server; • **'4'** – test of the LAN interface if present; • **'6'** – test of the SD card performance; • **'9'** – setup of the BLE module (if present); • **'10'** – test of the modem without PPP connection; • **'11'** – send all unsent documents (command execution is accepted only once in every 5 minutes);

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example

Binary log	
Request	01 30 30 32 3C 27 30 30 34 37 31 09 05 30 31 3F 3F 03
Answer	01 30 30 33 35 27 30 30 34 37 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 31 3C 03
Human oriented log	
Request	1[\t]
Answer	0[\t]

4.31.2. Information about the connection with NRA server

Request

	Name	Type	Opt	Value	Description
1	**InfoType**	uint		2	Information about the connection with NRA server

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**LastDate**	char	See remark: [*27]	Last connection to the server.
3	**NextDate**	char	See remark: [*27]	Next connection to the server.
4	**Zrep**	uint	1...3650	Last send Z report.
5	**ZErrnReport**	uint	1...3650	Number of Z report with error.
6	**ZErrCnt**	uint	0...65535	Sum of all errors for Z reports.
7	**ZErrStatus**	int	-99...0	Error number from the server.
8	**SellErrnDoc**	uint	1...9999999	Number of sell document with error.
9	**SellErrCnt**	uint	0...65535	Sum of all errors for sell documents.
10	**SellErrStatus**	int	-99...0	Error number from the server.
11	**SellNumber**	uint	1...9999999	Last received document number from the server.
12	**SellDate**	char	See remark: [*27]	The date and time of last received document from the server.
13	**LastErr**	uint	See error code table	Last error from the server.
14	**RemMinutes**	uint	0...255	Remaining minutes until next GetDeviceInfo request.

Example

Binary log

Request	01 30 30 32 3C 88 30 30 34 37 32 09 05 30 32 36 31 03
Answer	01 30 30 38 3A 88 30 30 34 37 30 09 30 34 2D 30 33 2D 32 30 32 30 20 32 30 3A 33 36 3A 34 30 09 32 31 2D 30 33 2D 32 30 32 30 20 31 35 3A 35 36 3A 33 31 09 32 33 32 09 30 09 30 09 30 09 30 09 30 09 30 09 34 35 37 34 09 32 39 2D 31 31 2D 32 30 31 39 20 31 34 3A 32 30 3A 32 34 09 30 09 35 09 04 80 80 80 80 86 9A 80 80 05 31 34 3F 35 03

Human oriented log

Request	2[\t]
Answer	0[\t]04-03-2020 20:36:40[\t]21-03-2020 15:56:31[\t]232[\t]0[\t]0[\t]0[\t]0[\t]0[\t]4574[\t]29-11-2019 14:20:24[\t]0[\t]5[\t]

4.32. Command 72 (48h) Fiscalization

Request					
	Name	Type	Opt	Value	Description
1	**SerialNumber**	char		Up to 8 symbols	Serial Number: Two letters and six digits: XX123456.
2	**TAXnumber**	char		Up to 13 symbols	TAX number.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 33 3D 27 30 30 34 38 44 54 36 33 36 35 39 31 09 30 30 30 37 31 33 33 39 31 09 05 30 35 37 38
Answer	01 30 30 33 35 27 30 30 34 38 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 31 3D 03
Human oriented log	
Request	DT636591[\t]000713391[\t]
Answer	0[\t]

4.33. Command 74 (4Ah) Reading the Status

Request (syntax #1, no parameters)

	Name	Type	Opt	Value	Description

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**StatusBytes**	uint	80h...FFh	Status Bytes (See the description of the status bytes: 6).

Example

Binary log	
Request	01 30 30 32 3A 31 30 30 34 3A 05 30 31 3D 30 03
Answer	01 30 30 33 3E 31 30 30 34 3A 30 09 80 80 80 80 86 9A 80 80 09 04 80 80 80 80 86 9A 80 80 05 30 3A 35 3B 03
Human oriented log	
Request	
Answer	0[\t]ЂЂЂЂ†ЉЂЂ[\t]

Request (syntax #2)

	Name	Type	Opt	Value	Description
1	**Option**	char		0	Current receipt status.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**PrintBufferStatus**	uint	0...1	• **'0'** – buffer is not empty; • **'1'** – empty buffer, no lines pending;
3	**ReceiptStatus**	uint	0...9	Status of the current receipt: • **'0'** – Receipt is closed; • **'1'** – Normal receipt is open; • **'2'** – Storno receipt is open. Reason "mistake by operator"; • **'3'** – Storno receipt is open. Reason "refund"; • **'4'** – Storno receipt is open. Reason "tax base reduction"; • **'5'** – Standard non-fiscal receipt is open; • **'6'** – Debit invoice is open; • **'7'** – Credit invoice is open. Reason "mistake by operator"; • **'8'** – Credit invoice is open. Reason "refund"; • **'9'** – Credit invoice is open. Reason "tax base reduction";
4	**Number**	uint	1...9999999	The number of the current or the last receipt.
5	**QRamount**	uint	0.00...9999999.99 or 0...999999999[*7]	Fiscal QR code – the amount of the last fiscal receipt.
6	**QRnumber**	uint	1...9999999	Fiscal QR code – the slip number of the last fiscal receipt.
7	**QRdatetime**	char	See remark: [*27]	Fiscal QR code – the date and time of the last fiscal receipt.

Example	
Binary log	
Request	01 30 30 32 3C 30 30 30 34 3A 30 09 05 30 32 30 3A 03
Answer	01 30 30 35 39 30 30 30 34 3A 30 09 31 09 30 09 35 31 37 09 32 39 2E 35 30 09 35 31 31 09 31 35 2D 30 35 2D 31 39 20 30 39 3A 32 37 3A 31 35 09 04 80 80 80 80 86 9A 80 80 05 30 3C 35 32 03
Human oriented log	
Request	0[\t]
Answer	0[\t]1[\t]0[\t]517[\t]29.50[\t]511[\t]15-05-19 09:27:15[\t]

Request (syntax #2)

	Name	Type	Opt	Value	Description
1	**Option**	char		1	Fiscal QR code string – the contents of the QR code printed in the last fiscal document.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**QRCodeString**	char	Up to 48 symbols	Fiscal QR code string.

Example	
Binary log	
Request	01 30 30 32 3C 23 30 30 34 3A 31 09 05 30 31 3F 3E 03
Answer	01 30 30 35 3F 23 30 30 34 3A 30 09 30 32 36 33 36 35 31 38 2A 30 30 31 36 38 36 33 2A 32 30 31 39 2D 30 38 2D 31 39 2A 31 35 3A 30 30 3A 32 36 2A 31 2E 30 30 09 04 80 80 80 80 86 9A 80 80 05 30 3E 33 38 03
Human oriented log	
Request	1[\t]
Answer	0[\t]**02636518*0016863*2019-08-19*15:00:26*1.00**[\t]

4.34. Command 76 (4Ch) Status of the fiscal transaction

Request (no parameters)					
	Name	Type	Opt	Value	Description

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**IsOpen**	uint	0...9	• **'0'** – Receipt is closed; • **'1'** – Normal receipt is open; • **'2'** – Storno receipt is open. Reason "mistake by operator"; • **'3'** – Storno receipt is open. Reason "refund"; • **'4'** – Storno receipt is open. Reason "tax base reduction"; • **'5'** – Standard non-fiscal receipt is open; • **'6'** – Debit invoice is open; • **'7'** – Credit invoice is open. Reason "mistake by operator"; • **'8'** – Credit invoice is open. Reason "refund"; • **'9'** – Credit invoice is open. Reason "tax base reduction";
3	**Number**	uint	1...9999999	The number of the current or the last receipt.
4	**Items**	uint	1...9999999	Number of sales registered on the current or the last fiscal receipt.
5	**Amount**	uint	0.00...9999999.99 or 0...999999999[*7]	The sum from the current or the last fiscal receipt.
6	**Payed**	uint	0.00...9999999.99 or 0...999999999[*7]	The sum paid for the current or the last receipt.

Example	
Binary log	
Request	01 30 30 32 3A 32 30 30 34 3C 05 30 31 3D 33 03
Answer	01 30 30 34 37 32 30 30 34 3C 30 09 30 09 35 31 37 09 30 09 30 2E 30 30 09 30 2E 30 30 09 04 80 80 80 80 80 86 9A 80 80 05 30 38 3D 35 03
Human oriented log	
Request	
Answer	0[\t]0[\t]517[\t]0[\t]0.00[\t]0.00[\t]

4.35. Command 80 (50h) Play sound

Request					
	Name	Type	Opt	Value	Description
1	**Hz**	uint		0...65535	Frequency.
2	**mSec**	uint		0...65535	Time in milliseconds.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 33 33 35 30 30 35 30 32 35 30 09 31 31 35 30 09 05 30 33 33 35 03
Answer	01 30 30 33 35 35 30 30 35 30 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 32 34 03
Human oriented log	
Request	250[\t]1150[\t]
Answer	0[\t]

4.36. Command 83 (53h) Programming of VAT rates

Request

	Name	Type	Opt	Value	Description
1	**TaxA**	uint		0.00...100.00	• **0.00**...**99.99** – enabled; • **100.00** – disabled;
2	**TaxB**	uint		0.00...100.00	• **0.00**...**99.99** – enabled; • **100.00** – disabled;
3	**TaxC**	uint		0.00...100.00	• **0.00**...**99.99** – enabled; • **100.00** – disabled;
4	**TaxD**	uint		0.00...100.00	• **0.00**...**99.99** – enabled; • **100.00** – disabled;
5	**TaxE**	uint		0.00...100.00	• **0.00**...**99.99** – enabled; • **100.00** – disabled;
6	**TaxF**	uint		0.00...100.00	• **0.00**...**99.99** – enabled; • **100.00** – disabled;
7	**TaxG**	uint		0.00...100.00	• **0.00**...**99.99** – enabled; • **100.00** – disabled;
8	**TaxH**	uint		0.00...100.00	• **0.00**...**99.99** – enabled; • **100.00** – disabled;
9	**decimal_point**	uint		0 or 2	• **'0'** - work with integer prices; • **'2'** - work with fractional prices;

Answer

	Name	Type	Opt	Value	Description
1	**ErrorCode**	int		-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**RemainingChanges**	int		1...30	Number of remaining changes.

Example

Binary log

Request	01 30 30 35 38 3A 30 30 35 33 30 2E 30 30 09 32 30 2E 30 30 09 32 30 2E 30 30 09 32 30 2E 30 30 09 39 2E 30 30 09 31 30 2E 30 30 09 35 2E 30 30 09 30 2E 30 30 09 32 09 05 30 39 31 3C 0301 30 30 33 38 3A 30 30 35 33 30 09 32 38 09 04 80 80 80 80 86 9A 80 80 05 30 36 3A 32 03
Answer	01 30 30 33 38 3A 30 30 35 33 30 09 32 38 09 04 80 80 80 80 86 9A 80 80 05 30 36 3A 32 03

Human oriented log

Request	0.00[\t]20.00[\t]20.00[\t]20.00[\t]9.00[\t]10.00[\t]5.00[\t]0.00[\t]2[\t]
Answer	0[\t]28[\t]

When changing decimal_point is necessary to restart the device so the correct values indicate on the client display

4.37. Command 84 (54h) Printing of barcode

Request

	Name	Type	Opt	Value	Description
1	**Type**	uint		1...7	Type of barcode: • **'1'** - EAN8 barcode. **Data** must contain only 8 digits; • **'2'** - EAN13 barcode. **Data** must contain only 13 digits; • **'3'** - Code128 barcode. **Data** must contain symbols with ASCII codes between 32 and 127. **Data** length is between 3 and 31 symbols; • **'4'** - QR code. **Data** must contain symbols with ASCII codes between 32 and 127. **Data** length is between 3 and 279 symbols; • **'5'** - Interleave 2of5 barcode. **Data** must contain only digits, from 3 to 22 chars; • **'6'** - PDF417 truncated **Data** must contain symbols with ASCII codes between 32 and 127. **Data** length is between 3 and 400 symbols; • **'7'** - PDF417 normal **Data** must contain symbols with ASCII codes between 32 and 127. **Data** length is between 3 and 400 symbols;
2	**Data**	char		Up to 400 symbols	Data of the barcode. Length of **Data** depends on the type of the barcode.
3	**QRcodeSize**	uint	●	3...10	Dots multiplier for QR barcodes and PDF417 barcodes. Default: 4;

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0

Example

Binary log	
Request	01 30 30 33 35 3F 30 30 35 34 31 09 31 32 33 34 35 36 37 38 09 05 30 33 3B 3C 03
Answer	01 30 30 33 35 3F 30 30 35 34 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 33 32 03
Human oriented log	
Request	1[\t]12345678[\t]
Answer	0[\t]

4.38. Command 86 (56h) The date of the last record in the fiscal memory

Request (no parameters)					
	Name	Type	Opt	Value	Description

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**DateTime**	char	See remark: [*27]	Date and time of last fiscal record.

Example	
Binary log	
Request	01 30 30 32 3A 89 30 30 35 36 05 30 32 32 35 03
Answer	01 30 30 34 39 89 30 30 35 36 30 09 30 37 2D 30 33 2D 32 30 32 30 20 31 36 3A 31 30 3A 35 32 09 04 80 80 80 80 86 9A 80 80 05 30 3A 33 37 03
Human oriented log	
Request	
Answer	0[\t]07-03-2020 16:10:52[\t]

4.39. Command 87 (57h) Get item groups information

Request

	Name	Type	Opt	Value	Description
1	**ItemGroup**	uint	●	1...99	Number of item group. If parameter **ItemGroup** is empty – item group report is printed.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**TotSales**	uint	0...99999999	Number of sales for this item group for day.
3	**TotSum**	uint	0.00...9999999.99 or 0...999999999[*7]	Accumulated sum for this item group for day.
4	**Name**	char	Up to 32 symbols	Name of item group.

Example

Binary log	
Request	01 30 30 32 3C 54 30 30 35 37 31 09 05 30 32 32 3D 03 16
Answer	01 30 30 34 38 54 30 30 35 37 30 09 30 2E 30 30 30 09 30 2E 30 30 09 C3 D0 D3 CF C0 20 31 09 04 80 80 80 80 86 9A 80 80 05 30 3C 35 3B 03
Human oriented log	
Request	1[\t]
Answer	0[\t]0.000[\t]0.00[\t]ГРУПА 1[\t]

4.40. Command 88 (58h) Get department information

Request

	Name	Type	Opt	Value	Description
1	**Departament**	int	●	1...99	Number of department. If parameter **Department** is empty – department report is printed.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**TaxGr**	uint	0...99999999	Tax group of department.
3	**Price**	uint	0.00...9999999.99 or 0...999999999[*7]	Default price.
4	**TotSales**	uint	0...99999999	Number of sales for this department for day.
5	**TotSum**	uint	0.00...9999999.99 or 0...999999999[*7]	Accumulated sum for this department for day.
6	**STotSales**	uint	0...99999999	Number of storno operations for this department for day.
7	**STotSum**	uint	0.00...9999999.99 or 0...999999999[*7]	Accumulated sum from storno operations for this department for day.
8	**Name**	char	Up to 72 symbols	Name of the department.

Example

Binary log	
Request	01 30 30 32 3C 55 30 30 35 38 31 09 05 30 32 32 3F 03
Answer	01 30 30 35 37 55 30 30 35 38 30 09 32 09 31 2E 30 30 09 30 2E 30 30 30 09 30 2E 30 30 09 30 2E 30 30 30 09 30 2E 30 30 09 C4 CF 20 31 09 04 80 80 80 80 86 9A 80 80 05 30 3C 3B 3C 03
Human oriented log	
Request	1[\t]
Answer	0[\t]2[\t]1.00[\t]0.000[\t]0.00[\t]0.000[\t]0.00[\t]ДП 1[\t]

4.41. Command 89 (59h) Test of the fiscal memory

Request

	Name	Type	Opt	Value	Default	Description
1	**Write**	uint	●	0...1	0	• **'0'** - Read test; • **'1'** - Write and read test;

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**Records**	uint	0...16	Number of records left.

Example

Binary log	
Request	01 30 30 32 3C 59 30 30 35 39 30 09 05 30 32 33 33 03
Answer	01 30 30 33 3A 59 30 30 35 39 30 09 30 30 31 35 09 04 80 80 80 80 86 9A 80 80 05 30 37 32 35 03
Human oriented log	
Request	0[\t]
Answer	0[\t]0015[\t]

4.42. Command 90 (5Ah) Diagnostic information

Request

	Name	Type	Opt	Value	Description
1	**option**	chat	●	empty or 1	Diagnostic information without firmware checksum.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**Name**	char	Up to 32 symbols	Device name.
3	**FwRev**	char	6 symbols	Firmware version.
4	**FwDate**	char	See remark: [*27]	Firmware date.
5	**FwTime**	char	See remark: [*27]	Firmware time.
6	**Checksum**	char	4 symbols	Firmware checksum. (empty if option from request is empty)
7	**Sw**	char	8 symbols	Switch from Sw1 to Sw8.
8	**SerialNumber**	char	Two letters and six digits: XX123456	Serial Number.
9	**FMNumber**	char	8 digits	Fiscal memory number.

Example

Binary log

Request	01 30 30 32 3C 51 30 30 35 3A 31 09 05 30 32 32 3D 03
Answer	01 30 30 37 30 51 30 30 35 3A 30 09 57 50 2D 35 30 58 09 32 36 31 32 31 36 09 31 32 4D 61 72 31 39 09 31 36 33 31 09 31 34 32 36 09 30 30 30 30 30 30 30 30 09 44 54 36 33 36 35 35 35 09 30 32 36 33 36 35 35 35 09 04 80 80 80 80 86 9A 80 80 05 31 31 3C 3F 03

Human oriented log

Request	1[\t]
Answer	0[\t]WP-50X[\t]261216[\t]12Mar19[\t]1631[\t]1426[\t]00000000[\t]DT636555[\t]02636555[\t]

4.43. Command 91 (5Bh) Programming of Serial number and FM number

Request

	Name	Type	Opt	Value	Description
1	**SerialNumber**	char		Up to 32 symbols	Serial Number.
2	**FMnumber**	char		8 digits	Fiscal Memory Number.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**Country**	char	Up to 32 symbols	Name of the country.

4.44. Command 92 (5Ch) Printing of separating line

Request

	Name	Type	Opt	Value	Description
1	**Type**	uint		1...4	Type of the separating line: • **'1'** - Separating line with the symbol '-'; • **'2'** - Separating line with the symbols '-' and ' '; • **'3'** - Separating line with the symbol '='; • **'4'** - Print fixed text "НЕ СЕ ДЪЛЖИ ПЛАЩАНЕ";

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example

Binary log	
Request	01 30 30 32 3C 9F 30 30 35 3C 31 09 05 30 32 37 3D 03
Answer	01 30 30 33 35 9F 30 30 35 3C 30 09 04 80 80 A0 80 86 9A 80 80 05 30 36 3B 3A 03
Human oriented log	
Request	1[\t]
Answer	0[\t]

4.45. Command 94 (5Eh) Fiscal memory report by date

Request

	Name	Type	Opt	Value	Default	Description
1	**Type**	uint		0...1		• **'0'** - short; • **'1'** - detailed;
2	**Start**	char	●	See remark: [*27]	Date of fiscalization	Start date w/o time
3	**End**	char	●	See remark: [*27]	Current date	End date w/o time

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0

Example

Binary log	
Request	01 30 30 33 3E 4E 30 30 35 3E 30 09 31 37 2D 30 35 2D 31 39 09 31 37 2D 30 35 2D 31 39 09 05 30 35 36 34 03
Answer	01 30 30 33 35 4E 30 30 35 3E 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 34 3B 03
Human oriented log	
Request	0[\t]17-05-19[\t]17-05-19[\t]
Answer	0[\t]

4.46. Command 95 (5Fh) Fiscal memory report by number of Z-report

Request

	Name	Type	Opt	Value	Default	Description
1	**Type**	uint		0...1		• **'0'** - short; • **'1'** - detailed
2	**First**	uint	●	1..3650	1	First Z-report in the period. (default: 1)
3	**Last**	uint	●	1..3650	Number of last Z-report	Last Z-report in the period. (default: last Z report number)

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example

Binary log	
Request	01 30 30 33 30 4C 30 30 35 3F 30 09 31 09 32 09 05 30 32 39 36 03
Answer	01 30 30 33 35 4C 30 30 35 3F 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 34 3A 03
Human oriented log	
Request	0[\t]1[\t]2[\t]
Answer	0[\t]

4.47. Command 96 (60h) Set software password

Request (syntax #1)						
	Name	Type	Opt	Value	Default	Description
1	**SoftPassword**	char		Up to 16 symbols		Software Password.

Request (syntax #2)						
	Name	Type	Opt	Value	Default	Description
1	**OldPasw**	char		Up to 16 symbols	empty string	Value of the old password.
2	**NewPasw**	char		Up to 16 symbols		Value of the new password.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3C 4B 30 30 36 30 31 09 05 30 32 31 3E 03
Answer	01 30 30 33 35 4B 30 30 36 30 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 33 3B 03
Human oriented log	
Request	1[\t]
Answer	0[\t]

4.48. Command 98 (62h) Programming of TAX number

Request					
	Name	Type	Opt	Value	Description
1	**TAXnumber**	char		Up to 13 symbols	TAX number.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 33 34 49 30 30 36 32 30 30 30 37 31 33 33 39 31 09 05 30 33 3A 3E 03
Answer	01 30 30 33 35 49 30 30 36 32 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 33 3B 03
Human oriented log	
Request	000713391[\t]
Answer	0[\t]

4.49. Command 99 (63h) Reading the programmed TAX number

Request (no parameters)					
	Name	Type	Opt	Value	Description

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**TAXnumber**	char	Up to 13 symbols	TAX number.

Example	
Binary log	
Request	01 30 30 32 3A 43 30 30 36 33 05 30 31 3D 3D 03
Answer	01 30 30 33 3F 43 30 30 36 33 30 09 30 30 30 37 31 33 33 39 31 09 04 80 80 80 80 86 9A 80 80 05 30 38 31 31 03
Human oriented log	
Request	
Answer	0[\t]000713391[\t]

4.50. Command 100 (64h) Reading an error

Request

	Name	Type	Opt	Value	Description
1	**Code**	int		Negative number	Code of the error.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**Code**	int	-999999...0	Code of the error, to be explained.
3	**ErrorMessage**	char		Explanation of the error in **Code.**

Example

Binary log	
Request	01 30 30 33 32 42 30 30 36 34 2D 31 31 31 30 31 36 09 05 30 33 33 36 03
Answer	01 30 30 34 3B 42 30 30 36 34 30 09 2D 31 31 31 30 31 36 09 C7 E0 F2 E2 EE F0 E5 ED 20 E1 EE ED 21 09 04 80 80 80 80 86 9A 80 80 05 31 31 3C 3E 03
Human oriented log	
Request	-111016[\t]
Answer	0[\t]-111016[\t]Затворен бон![\t]

4.51. Command 101 (65h) Set operator password

Request					
	Name	Type	Opt	Value	Description
1	**OpCode**	uint		1...30	Operator number from.
2	**NewPwd**	char		1...8	Operator password, ASCII string of digits.
3	**OldPwd**	char	●	Blank if service jumper is on.	Operator old password or administrator (oper29 & oper30) password.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 33 30 41 30 30 36 35 31 09 31 09 31 09 05 30 32 38 32 03
Answer	01 30 30 33 35 41 30 30 36 35 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 33 36 03
Human oriented log	
Request	1[\t]1[\t]1[\t]
Answer	0[\t]

4.52. Command 103 (67h) Information for the current receipt

Request (no parameters)					
	Name	Type	Opt	Value	Description

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**SumVATx**	uint	0.00...9999999.99 or 0...999999999[*7]	The current accumulated sum on VATx.
3	**Inv**	uint	0...1	Invoice: • **'0'** - if it is simplified receipt; • **'1'** - if it is expanded receipt;
4	**InvNumb**	char	Up to 10 digits	Number of the next invoice.
5	**fStorno**	uint	0...1	Flag indicating type of the receipt. • **'0'** - if it is normal receipt • **'1'** - if a storno receipt is open

Example	
Binary log	
Request	01 30 30 32 3A 40 30 30 36 37 05 30 31 3D 3E 03
Answer	01 30 30 36 33 26 30 30 36 37 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 31 09 35 09 30 09 04 80 80 88 80 86 9A 80 80 05 30 3D 30 3F 03
Human oriented log	
Request	
Answer	0[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]1[\t]5[\t]0[\t]

4.53. Command 105 (69h) Print of operator's report

Request						
	Name	Type	Opt	Value	Default	Description
1	**FirstOper**	uint	●	1...30	1	First operator.
2	**LastOper**	uint	●	1...30	Maximum operator number	Last operator.
3	**Clear**	uint	●	0...1	0	Clear registers for operators: • **'0'** - Does not clear registers for operators; • **'1'** - Clear registers for operators;

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0

Example	
Binary log	
Request	01 30 30 33 30 3D 30 30 36 39 31 09 32 09 30 09 05 30 32 38 32 03
Answer	01 30 30 33 35 3D 30 30 36 39 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 33 36 03
Human oriented log	
Request	1[\t]2[\t]0[\t]
Answer	0[\t]

4.54. Command 106 (6Ah) Drawer opening

Request

	Name	Type	Opt	Value	Description
1	**mSec**	uint	●	0...65535	The length of the impulse in milliseconds.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example

Binary log	
Request	01 30 30 32 3C 3C 30 30 36 3A 30 09 05 30 32 31 38 03 16
Answer	01 30 30 33 35 3C 30 30 36 3A 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 33 36 03
Human oriented log	
Request	0[\t]
Answer	0[\t]

The command is not used on FMP-350X and FMP-55X

4.55. Command 107 (6Bh) Defining and reading items

4.55.1. Item programming – option ‘P’ - Programming item

Request (syntax #1)

	Name	Type	Opt	Value	Description
1	**Operation**	char		P	Operation type.
2	**PLU**	uint		1...100000[*8] 1...3000[*9]	Item number.
3	**TaxGr**	char		Letter 'A'...'H' or cyrillic 'A'...'З'	VAT group.
4	**Dep**	uint		0...99	Department number.
5	**Group**	uint		1...99	Item group number.
6	**PriceType**	uint		0...2	Price type: • **'0'** - fixed price; • **'1'** - free price; • **'2'** - max price;
7	**Price**	uint		0,00...9999999,99 or 0...999999999[*7]	Item price.
8	**AddQty**	char	●	Byte with value 'A'	Change of the available quantity for item
9	**Quantity**	uint		0,001...999999,999	Stock quantity.
10	**Bar1**	char	●	Up to 13 digits	Barcode 1.
11	**Bar2**	char	●	Up to 13 digits	Barcode 2.
12	**Bar3**	char	●	Up to 13 digits	Barcode 3.
13	**Bar4**	char	●	Up to 13 digits	Barcode 4.
14	**Name**	char		Up to 72 symbols	Item name.

Answer (syntax #1)

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example

Binary log

Request	01 30 30 36 3E 27 30 30 36 3B 50 09 33 09 C2 09 33 09 33 09 31 09 33 2E 30 30 09 41 09 31 30 30 30 09 31 30 30 30 30 30 34 09 32 30 30 30 30 30 34 09 33 30 30 30 30 30 34 09 34 30 30 30 30 30 34 09 D0 CE CB C5 D0 20 CC C5 D2 C0 CB 09 05 31 33 35 34 03
Answer	01 30 30 33 35 27 30 30 36 3B 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 32 32 03

Human oriented log

Request	P[\t]3[\t]B[\t]3[\t]3[\t]1[\t]3.00[\t]A[\t]1000[\t]1000004[\t]2000004[\t]3000004[\t]4000004[\t]РОЛЕР МЕТАЛ[\t]
Answer	0[\t]

Request: Syntax #2					
	Name	Type	Opt	Value	Description
1	**Operation**	char		P	Operation type.
2	**PLU**	uint		1...100000[*8] 1...3000[*9]	Item number.
3	**TaxGr**	char		Letter 'A'...'H' or cyrillic 'A'...'З'	VAT group.
4	**Dep**	uint		0...99	Department number.
5	**Group**	uint		1...99	Item group number.
6	**PriceType**	uint		0...2	Price type: • **'0'** - fixed price; • **'1'** - free price; • **'2'** - max price;
7	**Price**	uint		0,00...9999999,99 or 0...999999999 [*7]	Item price.
8	**AddQty**	char	●	Byte with value 'A'	Change of the available quantity for item
9	**Quantity**	uint		0,001...999999,999	Stock quantity.
10	**Bar1**	char	●	Up to 13 digits	Barcode 1.
11	**Bar2**	char	●	Up to 13 digits	Barcode 2.
12	**Bar3**	char	●	Up to 13 digits	Barcode 3.
13	**Bar4**	char	●	Up to 13 digits	Barcode 4.
14	**Name**	char		Up to 72 symbols	Item name.
15	**Measurement unit**	uint		0...19	By default: • **'0'** - “бр.“; • **'1'** - “кг“; • **'2'** - “м“; • **'3'** - “л“; • **'4'** - “ч“; • **'5-19'** - “ ”;

Answer: (syntax #2)				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example syntax #2	
Binary log	
Request	01 30 30 37 36 21 30 30 36 3B 50 09 31 30 09 C1 09 32 09 32 09 31 09 31 2E 30 39 09 41 09 31 30 30 30 09 31 30 30 30 31 31 31 09 32 30 30 30 31 31 31 09 33 30 30 30 31 31 31 09 34 30 30 30 31 31 31 09 C1 FA EB E3 E0 F0 F1 EA E8 20 FF E1 FA EB EA E8 09 31 09 05 31 39 37 36 03
Answer	01 30 30 33 35 21 30 30 36 3B 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 31 3C 03
Human oriented log syntax #2	
Request	P[\t]10[\t]Б[\t]2[\t]2[\t]1[\t]1.09[\t]A[\t]1000[\t]1000111[\t]2000111[\t]3000111[\t]4000111[\t]Български ябълки[\t]1[\t]
Answer	0[\t]

4.55.2. Item programming – option 'I' - Item information

Request					
	Name	Type	Opt	Value	Description
1	**Operation**	char		I	Information for item.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**Total**	uint	100000[*8] 3000[*9]	Total count of the programmable items.
3	**Prog**	uint	0...100000[*8] 0...3000[*9]	Total count of the programmed items.
4	**NameLen**	uint	72	Maximum length of item name.

Example	
Binary log	
Request	01 30 30 32 3C 2F 30 30 36 3B 49 09 05 30 32 32 35 03
Answer	01 30 30 34 31 2F 30 30 36 3B 30 09 31 30 30 30 30 30 09 35 09 37 32 09 04 80 80 80 80 86 9A 80 80 05 30 38 30 31 03
Human oriented log	
Request	I[\t]
Answer	0[\t]100000[\t]5[\t]72[\t]

4.55.3. Item programming – option 'A' - Add stock quantity for item

Request					
	Name	Type	Opt	Value	Description
1	**Operation**	char		A	Change of the available quantity for item.
2	**PLU**	uint		1...100000[*8] 1...3000[*9]	Item number.
3	**Quantity**	uint		0.001...999999.999	Stock quantity.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 33 32 32 30 30 36 3B 41 09 33 09 32 30 30 09 05 30 32 3E 3E 03
Answer	01 30 30 33 35 32 30 30 36 3B 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 32 3D 03
Human oriented log	
Request	A[\t]3[\t]200[\t]
Answer	0[\t]

4.55.4. Item programming – option ‘D’ - Item deleting

Request					
	Name	Type	Opt	Value	Description
1	**Operation**	char		D	Item deleting.
2	**firstPLU**	uint		1...100000[*8] 1...3000[*9]	Item number.
3	**lastPLU**	uint	●	1...100000[*8] 1...3000[*9]	Item number.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0

Example	
Binary log	
Request	01 30 30 33 32 34 30 30 36 3B 44 09 33 30 09 34 30 09 05 30 32 3F 35 03
Answer	01 30 30 33 35 34 30 30 36 3B 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 32 3F 03
Human oriented log	
Request	D[\t]30[\t]40[\t]
Answer	0[\t]

4.55.5. Item programming – option ‘R’ - Reading item

Request

	Name	Type	Opt	Value	Description
1	**Operation**	char		R	Reading item data.
2	**PLU**	uint		1...100000[*8] 1...3000[*9]	Item number.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**PLU**	uint	1...100000[*8] 1...3000[*9]	Item number.
3	**TaxGr**	char	Letter 'A'...'H' or cyrillic 'A'...'З'	VAT group.
4	**Dep**	uint	0...99	Department.
5	**Group**	uint	1...99	Stock group.
6	**PriceType**	uint	0...2	Price type: • **'0'** - fixed price; • **'1'** - free price; • **'2'** - max price;
7	**Price**	uint	0,00...9999999,99 or 0...999999999[*7]	Item price.
8	**Turnover**	uint	0,00...9999999,99 or 0...999999999[*7]	Accumulated amount of the item.
9	**SoldQty**	uint	0.001...999999.999	Sold out quantity.
10	**StockQty**	uint	0.001...999999.999	Current quantity.
11	**Bar1**	char	Up to 13 digits	Barcode 1.
12	**Bar2**	char	Up to 13 digits	Barcode 2.
13	**Bar3**	char	Up to 13 digits	Barcode 3.
14	**Bar4**	char	Up to 13 digits	Barcode 4.
15	**Name**	char	Up to 72 symbols	Item name.
16	**Units**	uint	0...19	Measurement unit.

Example

Binary log

Request	01 30 30 32 3E 41 30 30 36 3B 46 09 32 09 05 30 32 37 31 03
Answer	01 30 30 38 3C 41 30 30 36 3B 30 09 33 09 31 09 33 09 33 09 31 09 31 2E 30 39 09 30 2E 30 30 09 30 2E 30 30 30 09 36 34 30 30 2E 30 30 30 09 31 30 30 30 30 30 34 09 32 30 30 30 30 30 34 09 33 30 30 30 30 30 34 09 34 30 30 30 30 30 34 09 CF FA EB ED EE E7 FA F0 ED E5 F1 F2 20 F5 EB FF E1 09 30 09 04 80 80 80 80 86 9A 80 80 05 32 30 34 35 03

Human oriented log

Request	F[\t]2[\t]
Answer	0[\t]3[\t]1[\t]3[\t]3[\t]1[\t]1.09[\t]0.00[\t]0.000[\t]6400.000[\t]1000004[\t]2000004[\t]3000004[\t]4000004[\t]Пълнозърнест хляб[\t]0[\t]

4.55.6. Item programming – option ‘F’ - Data about the first found programmed item

Request

	Name	Type	Opt	Value	Description
1	**Operation**	char		F	Returns data about the first found programmed item.
2	**PLU**	uint	●	1...100000[*8] 1...3000[*9]	Item number.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**PLU**	uint	1...100000[*8] 1...3000[*9]	Item number.
3	**TaxGr**	uint	Letter 'A'...'H' or cyrillic 'A'...'З'	VAT group.
4	**Dep**	uint	0...99	Department.
5	**Group**	uint	1...99	Stock group.
6	**PriceType**	uint	0...2	Price type: • **'0'** - fixed price; • **'1'** - free price; • **'2'** - max price;
7	**Price**	uint	0,00...9999999,99 or 0...999999999[*7]	Item price.
8	**Turnover**	uint	0,00...9999999,99 or 0...999999999[*7]	Accumulated amount of the item.
9	**SoldQty**	uint	0.001...999999.999	Sold out quantity.
10	**StockQty**	uint	0.001...999999.999	Current quantity.
11	**Bar1**	char	Up to 13 digits	Barcode 1.
12	**Bar2**	char	Up to 13 digits	Barcode 2.
13	**Bar3**	char	Up to 13 digits	Barcode 3.
14	**Bar4**	char	Up to 13 digits	Barcode 4.
15	**Name**	char	Up to 72 symbols	Item name.
16	**Units**	uint	0...19	Measurement unit.

4.55.7. Item programming – option 'L' - Data about the last found programmed item

Request

	Name	Type	Opt	Value	Description
1	**Operation**	char		L	Returns data about the last found programmed item.
2	**PLU**	uint	●	1...100000[*8] 1...3000[*9]	Item number.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**PLU**	uint	1...100000[*8] 1...3000[*9]	Item number.
3	**TaxGr**	char	Letter 'A'...'H' or cyrillic 'A'...'З'	VAT group.
4	**Dep**	uint	0...99	Department.
5	**Group**	uint	1...99	Stock group.
6	**PriceType**	uint	0...2	Price type: • **'0'** - fixed price; • **'1'** - free price; • **'2'** - max price;
7	**Price**	uint	0,00...9999999,99 or 0...999999999[*7]	Item price.
8	**Turnover**	uint	0,00...9999999,99 or 0...999999999[*7]	Accumulated amount of the item.
9	**SoldQty**	uint	0.001...999999.999	Sold out quantity.
10	**StockQty**	uint	0.001...999999.999	Current quantity.
11	**Bar1**	char	Up to 13 digits	Barcode 1.
12	**Bar2**	char	Up to 13 digits	Barcode 2.
13	**Bar3**	char	Up to 13 digits	Barcode 3.
14	**Bar4**	char	Up to 13 digits	Barcode 4.
15	**Name**	char	Up to 72 symbols	Item name.
16	**Units**	uint	0...19	Measurement unit.

Example

Binary log

Request	01 30 30 32 3F 43 30 30 36 3B 4C 09 35 30 09 05 30 32 3A 3D 03
Answer	01 30 30 38 3B 43 30 30 36 3B 30 09 31 30 09 32 09 32 09 32 09 31 09 31 2E 30 39 09 31 2E 30 39 09 31 2E 30 30 30 09 39 39 39 2E 30 30 30 09 31 30 30 30 31 31 31 09 32 30 30 30 31 31 31 09 33 30 30 30 31 31 31 09 34 30 30 30 31 31 31 09 C1 FA EB E3 E0 F0 F1 EA E8 20 FF E1 FA EB EA E8 09 31 09 04 80 80 80 80 86 9A 80 80 05 31 3F 33 3A 03

Human oriented log

Request	L[\t]50[\t]
Answer	0[\t]10[\t]2[\t]2[\t]2[\t]1[\t]1.09[\t]1.09[\t]1.000[\t]1000.000[\t]1000111[\t]2000111[\t]3000111[\t]4000111[\t]Български ябълки[\t]1[\t]

4.55.8. Item programming – option 'N' - Data for the next found programmed item

Request

	Name	Type	Opt	Value	Description
1	**Operation**	char		N	Returns data for the next found programmed item.
2	**PLU**	uint		1...100000[*8] 1...3000[*9]	Item number.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**PLU**	uint	1...100000[*8] 1...3000[*9]	Item number.
3	**TaxGr**	char	Letter 'A'...'H' or cyrillic 'A'...'3'	VAT group.
4	**Dep**	uint	0...99	Department.
5	**Group**	uint	1...99	Stock group.
6	**PriceType**	uint	0...2	Price type: • **'0'** - fixed price; • **'1'** - free price; • **'2'** - max price;
7	**Price**	uint	0,00...9999999,99 or 0...999999999[*7]	Item price.
8	**Turnover**	uint	0,00...9999999,99 or 0...999999999[*7]	Accumulated amount of the item.
9	**SoldQty**	uint	0.001...999999.999	Sold out quantity.
10	**StockQty**	uint	0.001...999999.999	Current quantity.
11	**Bar1**	char	Up to 13 digits	Barcode 1.
12	**Bar2**	char	Up to 13 digits	Barcode 2.
13	**Bar3**	char	Up to 13 digits	Barcode 3.
14	**Bar4**	char	Up to 13 digits	Barcode 4.
15	**Name**	char	Up to 72 symbols	Item name.
16	**Units**	uint	0...19	Measurement unit.

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

Example

Binary log

Request	01 30 30 32 3C 44 30 30 36 3B 4E 09 05 30 32 33 3F 03
Answer	01 30 30 38 3A 44 30 30 36 3B 30 09 39 09 32 09 39 09 39 09 31 09 34 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 30 09 31 30 30 30 2E 30 30 30 09 31 30 30 30 30 31 30 09 32 30 30 30 30 31 30 09 33 30 30 30 30 31 30 09 34 30 30 30 30 31 30 09 C2 F1 E8 F7 EA EE 20 E4 F0 F3 E3 EE 20 F5 32 09 31 09 04 80 80 80 80 86 9A 80 80 05 31 3C 3B 33 03

Human oriented log

Request	N[\t]
Answer	0[\t]9[\t]2[\t]9[\t]9[\t]1[\t]4.00[\t]0.00[\t]0.000[\t]1000.000[\t]1000010[\t]2000010[\t]3000010[\t]4000010[\t]Всичко друго x2[\t]1[\t]

4.55.9. Item programming – option ‘f’ - Data about the first found item with sales on it

Request

	Name	Type	Opt	Value	Description
1	**Operation**	char		f	Returns data about the first found item with sales on it.
2	**PLU**	uint	●	1...100000[*8] 1...3000[*9]	Item number.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**PLU**	uint	1...100000[*8] 1...3000[*9]	Item number.
3	**TaxGr**	char	Letter 'A'...'H' or cyrillic 'A'...'З'	VAT group.
4	**Dep**	uint	0...99	Department.
5	**Group**	uint	1...99	Stock group.
6	**PriceType**	uint	0...2	Price type: • **'0'** - fixed price; • **'1'** - free price; • **'2'** - max price;
7	**Price**	uint	0,00...9999999,99 or 0...999999999[*7]	Item price.
8	**Turnover**	uint	0,00...9999999,99 or 0...999999999[*7]	Accumulated amount of the item.
9	**SoldQty**	uint	0.001...999999.999	Sold out quantity.
10	**StockQty**	uint	0.001...999999.999	Current quantity.
11	**Bar1**	char	Up to 13 digits	Barcode 1.
12	**Bar2**	char	Up to 13 digits	Barcode 2.
13	**Bar3**	char	Up to 13 digits	Barcode 3.
14	**Bar4**	char	Up to 13 digits	Barcode 4.
15	**Name**	char	Up to 72 symbols	Item name.
16	**Units**	uint	0...19	Measurement unit.

Example

Binary log

Request	01 30 30 32 3E 47 30 30 36 3B 66 09 32 09 05 30 32 39 37 03
Answer	01 30 30 38 3C 47 30 30 36 3B 30 09 33 09 31 09 33 09 33 09 31 09 31 2E 30 39 09 33 2E 32 37 09 33 2E 30 30 30 09 36 33 39 37 2E 30 30 30 09 31 30 30 30 30 30 34 09 32 30 30 30 30 30 34 09 33 30 30 30 30 30 34 09 34 30 30 30 30 30 34 09 CF FA EB ED EE E7 FA F0 ED E5 F1 F2 20 F5 EB FF E1 09 30 09 04 80 80 80 80 86 9A 80 80 05 32 30 36 39 03

Human oriented log

Request	f[\t]2[\t]
Answer	0[\t]3[\t]1[\t]3[\t]3[\t]1[\t]1.09[\t]3.27[\t]3.000[\t]6397.000[\t]1000004[\t]2000004[\t]3000004[\t]4000004[\t]Пълнозърнест хляб[\t]0[\t]

4.55.10. Item programming – option ‘l’ - Data about the last found item with sales on it

Request

	Name	Type	Opt	Value	Description
1	**Operation**	char		1	Returns data about the last found item with sales on it.
2	**PLU**	uint	●	1...100000[*8] 1...3000[*9]	Item number.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**PLU**	uint	1...100000[*8] 1...3000[*9]	Item number.
3	**TaxGr**	char	Letter 'A'...'H' or cyrillic 'A'...'З'	VAT group.
4	**Dep**	uint	0...99	Department.
5	**Group**	uint	1...99	Stock group.
6	**PriceType**	uint	0...2	Price type: • **'0'** - fixed price; • **'1'** - free price; • **'2'** - max price;
7	**Price**	uint	0,00...9999999,99 or 0...999999999[*7]	Item price.
8	**Turnover**	uint	0,00...9999999,99 or 0...999999999[*7]	Accumulated amount of the item.
9	**SoldQty**	uint	0.001...999999.999	Sold out quantity.
10	**StockQty**	uint	0.001...999999.999	Current quantity.
11	**Bar1**	char	Up to 13 digits	Barcode 1.
12	**Bar2**	char	Up to 13 digits	Barcode 2.
13	**Bar3**	char	Up to 13 digits	Barcode 3.
14	**Bar4**	char	Up to 13 digits	Barcode 4.
15	**Name**	char	Up to 72 symbols	Item name.
16	**Units**	uint	0...19	Measurement unit.

Example

Binary log

Request	01 30 30 32 3E 4C 30 30 36 3B 6C 09 38 09 05 30 32 3A 38 03
Answer	01 30 30 38 3A 4C 30 30 36 3B 30 09 36 09 32 09 32 09 32 09 31 09 39 2E 32 30 09 35 35 2E 32 30 09 36 2E 30 30 30 09 39 39 34 2E 30 30 30 09 31 30 30 30 30 30 37 09 32 30 30 30 30 30 37 09 33 30 30 30 30 30 37 09 34 30 30 30 30 30 37 09 D1 E2 E8 ED F1 EA E8 20 F0 E5 E1 FA F0 F6 E0 09 31 09 04 80 80 80 80 86 9A 80 80 05 31 3E 36 38 03

Human oriented log

Request	l[\t]8[\t]
Answer	0[\t]6[\t]2[\t]2[\t]2[\t]1[\t]9.20[\t]55.20[\t]6.000[\t]994.000[\t]1000007[\t]2000007[\t]3000007[\t]4000007[\t]Свински ребърца[\t]1[\t]

4.55.11. Item programming – option ‘n’ - Data for the next found programmed item with sales on it

Request

	Name	Type	Opt	Value	Description
1	**Operation**	char		n	Returns data for the next found programmed item with sales on it. 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).*

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**PLU**	uint	1...100000[*8] 1...3000[*9]	Item number.
3	**TaxGr**	char	Letter 'A'...'H' or cyrillic 'A'...'З'	VAT group.
4	**Dep**	uint	0...99	Department.
5	**Group**	uint	1...99	Stock group.
6	**PriceType**	uint	0...2	Price type: • **'0'** - fixed price; • **'1'** - free price; • **'2'** - max price;
7	**Price**	uint	0,00...9999999,99 or 0...999999999[*7]	Item price.
8	**Turnover**	uint	0,00...9999999,99 or 0...999999999[*7]	Accumulated amount of the item.
9	**SoldQty**	uint	0.001...999999.999	Sold out quantity.
10	**StockQty**	uint	0.001...999999.999	Current quantity.
11	**Bar1**	char	Up to 13 digits	Barcode 1.
12	**Bar2**	char	Up to 13 digits	Barcode 2.
13	**Bar3**	char	Up to 13 digits	Barcode 3.
14	**Bar4**	char	Up to 13 digits	Barcode 4.
15	**Name**	char	Up to 72 symbols	Item name.
16	**Units**	uint	0...19	Measurement unit.

Example

Binary log

Request	01 30 30 32 3C 4D 30 30 36 3B 6E 09 05 30 32 36 38 03
Answer	01 30 30 39 31 4D 30 30 36 3B 30 09 35 09 31 09 35 09 35 09 31 09 31 2E 38 30 09 39 2E 30 30 09 35 2E 30 30 30 09 39 39 35 2E 30 30 30 09 31 30 30 30 30 30 36 09 32 30 30 30 30 30 36 09 33 30 30 30 30 30 36 09 34 30 30 30 30 30 36 09 CF F0 FF F1 ED EE 20 EC EB FF EA EE 20 22 C2 E5 F0 E5 FF 22 2D 32 25 09 30 09 04 80 80 80 80 86 9A 80 80 05 32 31 30 35 03

Human oriented log

Request	n[\t]
Answer	0[\t]5[\t]1[\t]5[\t]5[\t]1[\t]1.80[\t]9.00[\t]5.000[\t]995.000[\t]1000006[\t]2000006[\t]3000006[\t]4000006[\t]Прясно мляко "Верея"-2%[\t]0[\t]

4.55.12. Item programming – option 'X' - Find the first not programmed item

Request					
	Name	Type	Opt	Value	Description
1	**Operation**	char		X	Find the first not programmed item.
2	**PLU**	uint	●	1...100000[*8] 1...3000[*9]	Item number.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**PLU**	uint	1...100000[*8] 1...3000[*9]	Item number.

Example	
Binary log	
Request	01 30 30 32 3E 4E 30 30 36 3B 58 09 32 09 05 30 32 39 30 03
Answer	01 30 30 33 38 4F 30 30 36 3B 30 09 31 31 09 04 80 80 80 80 86 9A 80 80 05 30 36 3B 38 03
Human oriented log	
Request	X[\t]4[\t]
Answer	0[\t]11[\t]

4.55.13. Item programming – option 'x' - Find the last not programmed item

Request					
	Name	Type	Opt	Value	Description
1	**Operation**	char		x	Find the last not programmed item.
2	**PLU**	uint	●	1...100000[*8] 1...3000[*9]	Starting Item number

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**PLU**	uint	1...100000[*8] 1...3000[*9]	Item number found.

Example	
Binary log	
Request	01 30 30 32 3E 4F 30 30 36 3B 58 09 34 09 05 30 32 39 33 03
Answer	01 30 30 33 38 4F 30 30 36 3B 30 09 31 31 09 04 80 80 80 80 86 9A 80 80 05 30 36 3B 38 03 31 09 04 80 80 80 80 86 9A 80 80 05 30 36 3B 38 03
Human oriented log	
Request	x[\t]4[\t]
Answer	0[\t]2[\t]

4.56. Command 109 (6Dh) Print duplicate copy of last fiscal receipt

Request (no parameters)					
	Name	Type	Opt	Value	Description

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3A 37 30 30 36 3D 05 30 31 3D 3B 03
Answer	01 30 30 33 35 37 30 30 36 3D 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 33 34 03
Human oriented log	
Request	
Answer	0[\t]

In order to use this command, the parameter **DublReceipts** must be set to 1.

4.57. Command 110 (6Eh) Additional daily information

4.57.1. Payments (sell operations)

Request

	Name	Type	Opt	Value	Default	Description
1	**Type**	uint	●	0	0	Type of information: • **'0'** - Payments (sell operations);

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0
2	**Cash**	uint	0.00...9999999.99 or 0...999999999[*7]	Value paid by payment cash
3	**Pay1**	uint	0.00...9999999.99 or 0...999999999[*7]	Value paid by payment 1
4	**Pay2**	uint	0.00...9999999.99 or 0...999999999[*7]	Value paid by payment 2
5	**Pay3**	uint	0.00...9999999.99 or 0...999999999[*7]	Value paid by payment 3
6	**Pay4**	uint	0.00...9999999.99 or 0...999999999[*7]	Value paid by payment 4
7	**Pay5**	uint	0.00...9999999.99 or 0...999999999[*7]	Value paid by payment 5
8	**ForeignPay**	uint	0.00...9999999.99 or 0...999999999[*7]	Value paid by foreign currency

Example

Binary log	
Request	01 30 30 32 3C C7 30 30 36 3E 30 09 05 30 32 3A 37 03
Answer	01 30 30 35 39 C7 30 30 36 3E 30 09 31 39 2E 30 34 09 31 2E 30 30 09 39 2E 30 30 09 31 2E 30 30 09 31 2E 30 30 09 31 2E 30 30 09 31 2E 30 30 09 04 80 80 80 80 86 9A 80 80 05 30 3C 38 38 03
Human oriented log	
Request	0[\t]
Answer	0[\t]19.04[\t]1.00[\t]9.00[\t]1.00[\t]1.00[\t]1.00[\t]1.00[\t]

4.57.2. Payments (storno operations)

Request

	Name	Type	Opt	Value	Default	Description
1	**Type**	uint		1		Type of information: • **'1'** - Payments (storno operations);

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0
2	**Cash**	uint	0.00...9999999.99 or 0...999999999[*7]	Value paid by payment cash
3	**Pay1**	uint	0.00...9999999.99 or 0...999999999[*7]	Value paid by payment 1
4	**Pay2**	uint	0.00...9999999.99 or 0...999999999[*7]	Value paid by payment 2
5	**Pay3**	uint	0.00...9999999.99 or 0...999999999[*7]	Value paid by payment 3
6	**Pay4**	uint	0.00...9999999.99 or 0...999999999[*7]	Value paid by payment 4
7	**Pay5**	uint	0.00...9999999.99 or 0...999999999[*7]	Value paid by payment 5
8	**ForeignPay**	uint	0.00...9999999.99 or 0...999999999[*7]	Value paid by foreign currency

Example

Binary log	
Request	01 30 30 32 3C C8 30 30 36 3E 31 09 05 30 32 3A 39 03
Answer	01 30 30 35 38 C8 30 30 36 3E 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 04 80 80 80 80 86 9A 80 80 05 30 3C 33 3C 03
Human oriented log	
Request	1[\t]
Answer	0[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]

4.57.3. Number and sum of sells

Request

	Name	Type	Opt	Value	Default	Description
1	**Type**	uint		2		Type of information: • **'2'** - number and sum of sells

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**Num**	uint	0...65535	Number of clients.
3	**Sum**	uint	0.00...9999999.99 or 0...999999999[*7]	Sum of the sells.

Example

Binary log

Request	01 30 30 32 3C C9 30 30 36 3E 32 09 05 30 32 3A 3B 03
Answer	01 30 30 33 3D C9 30 30 36 3E 30 09 32 09 33 34 2E 30 30 09 04 80 80 80 80 86 9A 80 80 05 30 38 30 38 03

Human oriented log

Request	2[\t]
Answer	0[\t]2[\t]34.00[\t]

4.57.4. Count and sum of discounts and surcharges

Request

	Name	Type	Opt	Value	Default	Description
1	**Type**	uint		3		Type of information: • **'3'** - count and sum of discounts and surcharges

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**qSur**	uint	0..999999	Count of surcharges.
3	**sSur**	uint	0.00...9999999.99 or 0...999999999[*7]	Sum of surcharges.
4	**qDis**	uint	0..999999	Count of discounts.
5	**sDis**	uint	0.00...9999999.99 or 0...999999999[*7]	Sum of discounts.

Example

Binary log

Request	01 30 30 32 3C CA 30 30 36 3E 33 09 05 30 32 3A 3D 03
Answer	01 30 30 34 33 CA 30 30 36 3E 30 09 30 09 30 2E 30 30 09 30 09 30 2E 30 30 09 04 80 80 80 80 86 9A 80 80 05 30 38 3C 37 03

Human oriented log

Request	3[\t]
Answer	0[\t]0[\t]0.00[\t]0[\t]0.00[\t]

4.57.5. Count and sum of corrections and annulled receipts

Request

	Name	Type	Opt	Value	Default	Description
1	**Type**	uint		4		Type of information: • **'4'** - count and sum of corrections and annulled receipts;

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**qVoid**	uint	0..999999	Count of operation of correction
3	**sVoid**	uint	0.00...9999999.99 or 0...999999999[*7]	Sum of operation of correction
4	**qAnul**	uint	0..999999	Count of annulled receipts
5	**sAnul**	uint	0.00...9999999.99 or 0...999999999[*7]	Sum of annulled receipts

Example

Binary log

Request	01 30 30 32 3C CB 30 30 36 3E 34 09 05 30 32 3A 3F 03
Answer	01 30 30 34 33 CB 30 30 36 3E 30 09 30 09 30 2E 30 30 09 30 09 30 2E 30 30 09 04 80 80 80 80 86 9A 80 80 05 30 38 3C 38 03

Human oriented log

Request	4[\t]
Answer	0[\t]0[\t]0.00[\t]0[\t]0.00[\t]

4.57.6. Count and sum of cash in and cash out operations

Request

	Name	Type	Opt	Value	Default	Description
1	**Type**	uint		5		Type of information: • **'5'** - Count and sum of cash in and cash out operations

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**qCashIn1**	uint	0..999999	Count of cash in operations.
3	**sCashIn1**	uint	0.00...9999999.99 or 0...999999999[*7]	Sum of cash in operations.
4	**qCashOut1**	uint	0..999999	Count of cash out operations.
5	**sCashOut1**	uint	0.00...9999999.99 or 0...999999999[*7]	Sum of cash out operations.
6	**qCashIn2**	uint	0..999999	Count of cash in operations in foreign currency.
7	**sCashIn2**	uint	0.00...9999999.99 or 0...999999999[*7]	Sum of cash in operations in foreign currency.
8	**qCashOut2**	uint	0..999999	Count of cash out operations in foreign currency.
9	**sCashOut2**	uint	0.00...9999999.99 or 0...999999999[*7]	Sum of cash out operations in foreign currency.

Example

Binary log	
Request	01 30 30 32 3C CC 30 30 36 3E 35 09 05 30 32 3B 31 03
Answer	01 30 30 35 31 CC 30 30 36 3E 30 09 30 09 30 2E 30 30 09 30 09 30 2E 30 30 09 30 09 30 2E 30 30 09 30 09 30 2E 30 30 09 04 80 80 80 80 86 9A 80 80 05 30 3A 3C 38 03
Human oriented log	
Request	5[\t]
Answer	0[\t]0[\t]0.00[\t]0[\t]0.00[\t]0[\t]0.00[\t]0[\t]0.00[\t]

4.58. Command 111 (65h) Print PLU report

Request

	Name	Type	Opt	Value	Description
1	**Type**	uint		0...3	Type of report: • **'0'** - PLU turnovers; • **'1'** - PLU turnovers with clearing; • **'2'** - PLU parameters; • **'3'** - PLU stock;
2	**FirstPLU**	uint	●	1...100000[*8] 1...3000[*9]	First PLU.
3	**LastPLU**	uint	●	1...100000[*8] 1...3000[*9]	Last PLU.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example

Binary log	
Request	01 30 30 33 30 2E 30 30 36 3F 30 09 31 09 32 09 05 30 32 37 39 03
Answer	01 30 30 33 35 2E 30 30 36 3F 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 32 3D 03
Human oriented log	
Request	0[\t]1[\t]2[\t]
Answer	0[\t]

4.59. Command 112 (70h) Information for operator

Request

	Name	Type	Opt	Value	Description
1	**Operator**	uint		1...30	Number of operator.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**Receipts**	uint	0...65535	Number of fiscal receipts, issued by the operator.
3	**Total**	uint	0.00...9999999.99 or 0...999999999[*7]	Total accumulated sum.
4	**SReceipts**	uint	0...65535	Number of storno receipts.
5	**STotal**	uint	0.00...9999999.99 or 0...999999999[*7]	Total accumulated sum from storno operations.
6	**nDiscount**	uint	0...65535	Number of discounts.
7	**Discount**	uint	0.00...9999999.99 or 0...999999999[*7]	Total accumulated sum of discounts with sign.
8	**nSurcharge**	uint	0...65535	Number of surcharges.
9	**Surcharge**	uint	0.00...9999999.99 or 0...999999999[*7]	Total accumulated sum of surcharges with sign.
10	**nVoid**	uint	0...65535	Number of corrections.
11	**Void**	uint	0.00...9999999.99 or 0...999999999[*7]	Total accumulated sum of corrections with sign.

Example

Binary log	
Request	01 30 30 32 3D 2B 30 30 37 30 33 30 09 05 30 32 33 32 03
Answer	01 30 30 35 38 84 30 30 37 30 30 09 30 09 30 2E 30 30 09 30 09 30 2E 30 30 09 30 09 30 2E 30 30 09 30 09 30 2E 30 30 09 30 09 30 2E 30 30 09 04 80 80 80 80 86 9A 80 80 05 30 3B 37 3A 03
Human oriented log	
Request	30[\t]
Answer	0[\t]0[\t]0.00[\t]0[\t]0.00[\t]0[\t]0.00[\t]0[\t]0.00[\t]0[\t]0.00[\t]0[\t]0.00[\t]

4.60. Command 116 (74h) Reading fiscal memory binary data.

Request

	Name	Type	Opt	Value	Description
1	**Operation**	uint		0	Type of operation.
2	**Address**	char		0...FFFFFF	Start address 0...FFFFFF (format ASCII-hex).
3	**nBytes**	uint		1...104	Number of bytes.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example

Binary log	
Request	01 30 30 33 35 2B 30 30 37 34 30 09 30 31 30 31 30 31 09 31 09 05 30 33 36 32 03
Answer	01 30 30 33 38 2B 30 30 37 34 30 09 46 46 09 04 80 80 80 80 86 9A 80 80 05 30 36 3B 38 03
Human oriented log	
Request	0[\t]010101[\t]1[\t]
Answer	0[\t]FF[\t]

4.61. Command 122 (7Ah) Printing of a free vertical fiscal text

Request

	Name	Type	Opt	Value	Description
1	**Text**	char		Text of 0...128 symbols	**Double-byte control codes:** (0Bh 42h) - Bolds all symbols. (0Bh 62h) - Stops the bolding of the symbols. (0Bh 55h) - Overlines all symbols. (0Bh 75h) - Stops the overlining of the symbols (0Bh 4Fh) - Underlines all symbols. (0Bh 6Fh) - Stops the underlining of the symbols. **Box drawing symbols:** (82h) - up right (84h) - up left (91h) - down right (92h) - down left (93h) - up right + down right (94h) - up left + down left (95h) - up left + down right (96h) - down left + down right (97h) - up right + down left + up left + down right (ACh) - vertical line

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

All information is automatically printed after XX executions of the command. XX=10 for 2 inch paper, XX=16 for 3 inch paper

4.62. Command 123 (7Bh) Device information

Request (1)

	Name	Type	Opt	Value	Description
1	**Option**	uint		1	Type of information to return: • **'1'** - Serial numbers, Header and Tax numbers;

Answer (1)

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**SerialNumber**	char	Up to 8 symbols	Serial number.
3	**FiscalNumber**	char	8 digits	Fiscal memory number.
4	**Headerline1**	char	Up to XX symbols	Supposed to contain 'Company name'. XX depends on print columns. • XX = 42,48,64[*11] • XX = 42[*8] • XX = 32[*12].
5	**Headerline2**	char	Up to XX symbols	Supposed to contain 'Company address'. XX depends on print columns. • XX = 42,48,64[*11] • XX = 42[*8] • XX = 32[*12].
6	**TAXnumber**	char	Up to 13 symbols	Tax number.
7	**Headerline3**	char	Up to XX symbols	Supposed to contain 'name of the business premises'. XX depends on print columns. • XX = 42,48,64[*11] • XX = 42[*8] • XX = 32[*12].
8	**Headerline4**	char	Up to XX symbols	Supposed to contain 'address of the business premises'. XX depends on print columns. • XX = 42,48,64[*11] • XX = 42[*8] • XX = 32[*12].

Example (1)

Binary log

Request	01 30 30 32 3C 52 30 30 37 3B 31 09 05 30 32 33 31 03
Answer	01 30 30 39 3E 52 30 30 37 3B 30 09 44 54 36 33 36 35 35 35 09 30 32 36 33 36 35 35 35 09 44 41 54 45 43 53 20 CE CE C4 09 D1 EE F4 E8 FF 2C 20 F3 EB 2E C4 E0 F2 E5 EA F1 20 34 09 30 30 30 37 31 33 33 39 31 09 D2 E5 F1 F2 EE E2 E8 20 28 F0 E0 E7 ED EE F1 ED E0 20 F2 FA F0 E3 EE E2 E8 FF 29 09 D1 EE F4 E8 FF 2C 20 F3 EB 2E C4 E0 F2 E5 EA F1 20 34 09 04 80 80 80 80 86 9A 80 80 05 33 3E 3E 3F 03

Human oriented log

Request	1[\t]
Answer	0[\t]DT636555[\t]02636555[\t]DATECS ООД[\t]София, ул.Датекс 4[\t]000713391[\t]Тестови (разносна търговия)[\t]София, ул.Датекс 4[\t]

Request (2)					
	Name	Type	Opt	Value	Description
1	**Option**	uint		2	Type of information to return: • **'2'** - Battery and GSM signal status;

Answer (2)				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**MainBattery**	uint	0...999999	Main Battery level in mV.
3	**RamBattery**	uint	0...999999	Ram Battery level in mV.
4	**Signal**	uint	0...100	GSM Signal level in percentage.
5	**Network**	uint	0...1	GSM network status: • **'0'** – unregistered; • **'1'** – registered;

Example (2)	
Binary log	
Request	01 30 30 32 3C 7E 30 30 37 3B 32 09 05 30 32 35 3E 03
Answer	01 30 30 34 34 7E 30 30 37 3B 30 09 38 36 36 36 09 34 31 35 37 09 39 36 09 31 09 04 80 80 80 80 86 9A 80 80 05 30 38 3E 39 03
Human oriented log	
Request	2[\t]
Answer	0[\t]8666[\t]4157[\t]96[\t]1[\t]

Request (3)					
	Name	Type	Opt	Value	Description
1	**Option**	uint		3	Type of information to return: • **'3'** - Last fiscal receipt;

Answer (3)				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**BonFiscal**	uint	1...9999	Number of last sales receipt in current Z report.
3	**DateBonFiscal**	char	[*27]	Date and time of last sales receipt.
4	**Znumber**	uint	1..3650	Number of last Z-report.
5	**Zdate**	char	[*27]	Date of last of Z-report.

Example (3)	
Binary log	
Request	01 30 30 32 3C 7F 30 30 37 3B 33 09 05 30 32 36 30 03
Answer	01 30 30 36 35 7F 30 30 37 3B 30 09 32 37 30 09 30 34 2D 30 34 2D 32 30 32 30 20 31 37 3A 33 38 3A 32 34 09 31 30 33 09 30 34 2D 30 34 2D 32 30 32 30 20 31 37 3A 33 38 3A 32 35 09 04 80 80 80 80 86 9A 80 80 05 30 3F 33 36 03
Human oriented log	
Request	3[\t]
Answer	0[\t]270[\t]04-04-2020 17:38:24[\t]103[\t]04-04-2020 17:38:25[\t]

Request (4)					
	Name	Type	Opt	Value	Description
1	**Option**	uint		4	Type of information to return: • **'4'** - Full EJ verify;

Answer (4)				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example (4)	
Binary log	
Request	01 30 30 32 3C 81 30 30 37 3B 34 09 05 30 32 36 33 03
Answer	01 30 30 33 35 81 30 30 37 3B 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 37 3D 03
Human oriented log	
Request	4[\t]
Answer	0[\t]

Request (5)					
	Name	Type	Opt	Value	Description
1	**Option**	uint		5	Type of information to return: • **'5'** - Battery level;

Answer (5)				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**MainBattery**	uint	0...999999	Main Battery level in mV.
3	**ChargeLevel**	uint	0...100	Battery charge percentage.

Example (5)	
Binary log	
Request	01 30 30 32 3C 80 30 30 37 3B 35 09 05 30 32 36 33 03
Answer	01 30 30 33 3D 80 30 30 37 3B 30 09 38 36 36 36 09 39 39 09 04 80 80 80 80 86 9A 80 80 05 30 37 3E 32 03
Human oriented log	
Request	5[\t]
Answer	0[\t]8666[\t]99[\t]

4.63. Command 124 (7Ch) Search receipt number by period

Request

	Name	Type	Opt	Value	Default	Description
1	**StartDate**	char	●	See remark: [*27]	Date and time of first document	Start date and time for searching.
2	**EndDate**	char	●	See remark: [*27]	Date and time of last document	End date and time for searching.
3	**DocType**	uint	●	0...10	0	Type of document: • **'0'** - all types; • **'1'** - fiscal receipts; • **'2'** - daily Z reports; • **'3'** - invoice receipts; • **'4'** - non-fiscal receipts; • **'5'** – paid out receipts; • **'6'** - fiscal receipts – storno; • **'7'** - invoice receipts – storno; • **'8'** - cancelled receipts (all voided); • **'9'** - daily X reports; • **'10'** - fiscal receipts, invoice receipts, fiscal receipts – storno and invoice receipts – storno;

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**StartDate**	char	See remark: [*27]	Start date for searching, see DateTime format described at the beginning of the document.
3	**EndDate**	char	See remark: [*27]	End date for searching, see DateTime format described at the beginning of the document.

4	**FirstDoc**	uint	1...9999999999	First document in the period
				• **DocType** = '0', '1', '4', '5', '6', '8', '9', '10'
				  ○ Number of document (1...9999999);
				• **DocType** = '2'
				  ○ Z report number (1...3650);
				• **DocType** = '3' or '7'
				  ○ Invoice number(1...9999999999);

5	**LastDoc**	uint	1...9999999999	Last document in the period
				• **DocType** = '0', '1', '4', '5', '6', '8', '9', '10'
				  ○ Number of document (1...9999999);
				• **DocType** = '2'
				  ○ Z report number (1...3650);
				• **DocType** = '3' or '7'
				  ○ Invoice number(1...9999999999);

Example

Binary log

Request	01 30 30 35 30 2D 30 30 37 3C 30 31 2D 30 35 2D 31 39 20 30 30 3A 30 30 3A 30 30 09 30 33 2D 30 35 2D 31 39 20 31 39 3A 30 30 3A 30 30 09 30 09 05 30 38 39 3D 03
Answer	01 30 30 36 39 2D 30 30 37 3C 30 09 30 31 2D 30 35 2D 31 39 20 30 30 3A 30 30 3A 30 30 20 44 53 54 09 30 33 2D 30 35 2D 31 39 20 31 39 3A 30 30 3A 30 30 20 44 53 54 09 34 37 31 09 34 37 39 09 04 80 80 80 80 86 9A 80 80 05 31 30 33 33 03

Human oriented log

Request	01-05-19 00:00:00{\t]03-05-19 19:00:00[\t]0[\t]
Answer	0[\t]01-05-19 00:00:00 DST[\t]03-05-19 19:00:00 DST[\t]471[\t]479[\t]

4.64. Command 125 (7Dh) Information from EJ

4.64.1. Set document to read

Request					
	Name	Type	Opt	Value	Description
1	**Option**	uint		0	Type of information: • **'0'** - Set document to read
2	**DocNum**	uint	●	1...9999999	Number of document.
3	**RecType**	uint	●	0...10	Document type: • **'0'** - all types; • **'1'** - fiscal receipts; • **'2'** - daily Z reports; • **'3'** - invoice receipts; • **'4'** – non-fiscal receipts; • **'5'** – paid out receipts; • **'6'** - fiscal receipts – storno; • **'7'** - invoice receipts – storno; • **'8'** - cancelled receipts (all voided); • **'9'** - daily X reports; • **'10'** - fiscal receipts, invoice receipts, fiscal receipts – storno and invoice receipts – storno;

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**DocNumber**	uint	1...9999999	Global number of document.
3	**RecNumber**	uint	1...9999999999	• **DocType** = '0', '1', '4', '5', '6', '8', '9', '10' ○ Number of document (1...9999999); • **DocType** = '2' ○ Z report number (1...3650); • **DocType** = '3' or '7' ○ Invoice number(1...9999999999);
4	**Date**	char	See remark: [*27]	Date of document
5	**Type**	uint	0...9	Type of document: • **'0'** - all types; • **'1'** - fiscal receipts; • **'2'** - daily Z reports; • **'3'** - invoice receipts; • **'4'** - non-fiscal receipts; • **'5'** – paid out receipts; • **'6'** - fiscal receipts – storno; • **'7'** - invoice receipts – storno; • **'8'** - cancelled receipts (all voided); • **'9'** - daily X reports;
6	**Znumber**	uint	1...3650	Number of Z report.

4.64.2. Read one line as text.

Must be called multiple times to read the whole document

Request					
	Name	Type	Opt	Value	Description
1	**Option**	uint		1	Type of information: Read one line as text
2	**DocNum**	uint	●	1...9999999	Number of document (see option ‘0’)
3	**RecType**	uint	●	0...10	Document type (see option ‘0’)

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**TextData**	char	Up to 64 symbols	Document text.

4.64.3. Read as data

Must be called multiple times to read the whole document

Request					
	Name	Type	Opt	Value	Description
1	**Option**	uint		2	Type of information: Read as data
2	**DocNum**	uint	●	1...9999999	Number of document (see option ‘0’)
3	**RecType**	uint	●	0...10	Document type (see option ‘0’)

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**Data**	char		Document data, structured information in base64 format. Detailed information in other document.

4.64.4. Print document

Request					
	Name	Type	Opt	Value	Description
1	**Option**	uint		3	Type of information: Print document
2	**DocNum**	uint	●	1...9999999	Number of document (see option ‘0’)
3	**RecType**	uint	●	0...10	Document type (see option ‘0’)

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

4.64.5. Set document to read in CSV formatted data

Request					
	Name	Type	Opt	Value	Description
1	**Option**	uint		9	Type of information: Set document to read
2	**FirstDoc**	uint		1...99999999	First document in the period. Number received in response to command 124.
3	**LastDoc**	uint		1...99999999	Last document in the period. Number received in response to command 124.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**DocNumber**	uint	1...9999999	Global number of document.
3	**RecNumber**	uint	1...9999999999	• **DocType** = '0', '1', '4', '5', '6', '8', '9', '10' ○ Number of document (1...9999999); • **DocType** = '2' ○ Z report number (1...3650); • **DocType** = '3' or '7' ○ Invoice number(1...9999999999);
4	**Date**	char	See remark: [*27]	Date of document, see DateTime format described at the beginning of the document.
5	**Type**	uint	0...9	Type of document: • **'0'** - all types; • **'1'** - fiscal receipts; • **'2'** - daily Z reports; • **'3'** - invoice receipts; • **'4'** - non-fiscal receipts; • **'5'** – paid out receipts; • **'6'** - fiscal receipts – storno; • **'7'** - invoice receipts – storno; • **'8'** - cancelled receipts (all voided); • **'9'** - daily X reports;
6	**Znumber**	uint	1...3650	Number of Z report.

4.64.6. *Read CSV formatted data*

Must be called multiple times to read the whole document

Request (syntax 3)					
	Name	Type	Opt	Value	Description
1	**Option**	uint		8	Type of information: Read as data

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**CSV_Col_1**	char	Up to 8 symbols	Identification number of FD.
3	**CSV_Col_2**	uint	ФБ, Разширен ФБ, Сторно ФБ, Разширен сторно ФБ	вид на ФБ – ФБ, Разширен ФБ, Сторно ФБ или Разширен сторно ФБ.
4	**CSV_Col_3**	char	1...9999999	номер на ФБ.
5	**CSV_Col_4**	char	"LLDDDDDD-CCCC-DDDDDDD", L[A-Z], C[0-9A-Za-z], D[0-9]	уникален номер на продажба (УНП) - в случай, че ФУ е от типа "Фискален принтер" или работи в такъв режим.
6	**CSV_Col_5**	char	Up to 72 symbols	стока/услуга – наименование.
7	**CSV_Col_6**	uint	0.01...999999.999[*4]	стока/услуга – единична цена.
8	**CSV_Col_7**	uint	0.01...999999.999[*4]	стока/услуга – количество.
9	**CSV_Col_8**	uint	0.01...999999.999[*4]	стока/услуга – стойност.
10	**CSV_Col_9**	uint	0.00...9999999.99 or 0...999999999[*7]	обща сума на ФБ/Сторно ФБ или Разширен ФБ/Разширен сторно ФБ.
11	**CSV_Col_10**	uint	1...9999999	номер на фактура/кредитно известие – в случай че записът е за Разширен ФБ или съответно – за Разширен сторно ФБ.
12	**CSV_Col_11**	char	8...13 symbols	ЕИК на получател – в случай че записът е за разширен ФБ или Разширен сторно ФБ.
13	**CSV_Col_12**	uint	1...9999999	номер на сторниран ФБ – в случай че записът се отнася за Сторно ФБ или Разширен сторно ФБ.
14	**CSV_Col_13**	uint	1...9999999	номер на сторнирана фактура – в случай че записът се отнася за Разширен сторно ФБ.
15	**CSV_Col_14**	char	Up to 48 symbols	причина за издаване – в случай че записът се отнася за Сторно ФБ или Разширен сторно ФБ.

Example	
Binary log	
Request	01 30 30 33 30 56 30 30 37 3D 30 09 31 09 30 09 05 30 32 39 3E 03
Answer	01 30 30 34 3F 56 30 30 37 3D 30 09 31 09 31 09 30 31 2D 30 31 2D 30 30 20 30 31 3A 33 36 3A 30 39 09 34 09 31 09 04 80 80 80 80 86 9A 80 80 05 30 3A 39 36 03
Human oriented log	
Request	0[\t]1[\t]0[\t]
Answer	0[\t]1[\t]1[\t]01-01-00[\t]01:36:09[\t]4[\t]1[\t]

4.65. *Command 126 (7Eh) Fiscal memory-structured information*

4.65.1. Ask for non-empty and max records

Request					
	Name	Type	Opt	Value	Description
1	**Type**	uint		0...9	Type of record: • **'0'** – Z reports; • **'1'** - ID number

					• **'2'** - Fiscal memory number • **'3'** - Open the fiscal memory • **'4'** - TAX number • **'5'** - VAT rates changes • **'6'** - Memory resetting events • **'7'** - NRA registered • **'8'** - NRA unregistered • **'9'** - EJ (KLEN) open/close
2	**Record**	uint		Empty	Empty parameter

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**NonEmpty**	int	0...9999	Count of occupied (non-empty) records according requested type.
3	**MaxRecords**	int	0...9999	Max count of records according requested type.

Example

Binary log	
Request	01 30 30 32 3D 6A 30 30 37 3E 30 09 09 05 30 32 35 35 03
Answer	01 30 30 33 3E 6A 30 30 37 3E 30 09 31 32 34 09 33 36 35 30 09 04 80 80 80 80 86 9A 80 80 05 30 37 3E 39 03
Request	0[\t][\t]
Answer	0[\t]124[\t]3650[\t]

4.65.2. Ask for Z reports structured information.

Request					
	Name	Type	Opt	Value	Description
1	**Type**	uint		0	Type of record**:** Z reports;
2	**Record**	uint		1...NonEmpty	According information about non-empty

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**Number**	int	1...9999	Record number
3	**DateTime**	char	See remark: [*30]	The date and time of record creation
4	**TaxA**	uint	0.00...9999999.99 or 0...999999999[*7]	Daily turnover by vat A.
5	**TaxB**	uint	0.00...9999999.99 or 0...999999999[*7]	Daily turnover by vat B.
6	**TaxC**	uint	0.00...9999999.99 or 0...999999999[*7]	Daily turnover by vat C.
7	**TaxD**	uint	0.00...9999999.99 or 0...999999999[*7]	Daily turnover by vat D.
8	**TaxE**	uint	0.00...9999999.99 or 0...999999999[*7]	Daily turnover by vat E.
9	**TaxF**	uint	0.00...9999999.99 or 0...999999999[*7]	Daily turnover by vat F.
10	**TaxG**	uint	0.00...9999999.99 or 0...999999999[*7]	Daily turnover by vat G.
11	**TaxH**	uint	0.00...9999999.99 or 0...999999999[*7]	Daily turnover by vat H.
12	**TaxTotal**	uint	0.00...9999999.99 or 0...999999999[*7]	Total of daily turnovers.
13	**StornoTaxA**	uint	0.00...9999999.99 or 0...999999999[*7]	Storno daily turnover by vat A.
14	**StornoTaxB**	uint	0.00...9999999.99 or 0...999999999[*7]	Storno daily turnover by vat B.
15	**StornoTaxC**	uint	0.00...9999999.99 or 0...999999999[*7]	Storno daily turnover by vat C.
16	**StornoTaxD**	uint	0.00...9999999.99 or 0...999999999[*7]	Storno daily turnover by vat D.
17	**StornoTaxE**	uint	0.00...9999999.99 or 0...999999999[*7]	Storno daily turnover by vat E.
18	**StornoTaxF**	uint	0.00...9999999.99 or 0...999999999[*7]	Storno daily turnover by vat F.
19	**StornoTaxG**	uint	0.00...9999999.99 or 0...999999999[*7]	Storno daily turnover by vat G.
20	**StornoTaxH**	uint	0.00...9999999.99 or 0...999999999[*7]	Storno daily turnover by vat H.
21	**StornoTotal**	uint	0.00...9999999.99 or 0...999999999[*7]	Total of storno daily turnovers.
22	**Hash**	char	40 symbols	Daily hash code
23	**LastDoc**	uint	1…9999999	Last document number
24	**nKlen**	uint	1.99	Last Klen number

Example	
Binary log	
Request	01 30 30 33 30 68 30 30 37 3E 30 09 31 32 34 09 05 30 32 3D 3E 03
Answer	01 30 30 3E 30 68 30 30 37 3E 30 09 30 31 32 34 09 31 33 2D 31 31 2D 32 30 31 39 20 30 39 3A 32 35 09 32 33 36 2E 30 30 09 31 35 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 33 38 36 2E 30 30 09 33 38 2E 30 30 09 38 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 30 2E 30 30 09 31 31 38 2E 30 30 09 45 39 33 32 39 45 38 43 35 42 30 34 35 38 45 33 38 45 34 35 37 42 32 46 34 33 42 34 33 44 37 43 45 45 33 46 41 44 34 35 09 31 37 32 31 35 09 33 09 04 80 80 80 80 86 9A 80 80 05 32 34 3F 39 03
Request	0[\t]124[\t]
Answer	0[\t]0124[\t]13-11-2019 09:25[\t]236.00[\t]150.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00 [\t]0.00[\t]386.00[\t]38.00[\t]80.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00[\t]0.00 [\t]118.00[\t]E9329E8C5B0458E38E457B2F43B43D7CEE3FAD45[\t]17215[\t]3[\t]

4.65.3. Ask for device ID number

Request					
	Name	Type	Opt	Value	Description
1	**Type**	uint		1	Type of record**:** ID number
2	**Record**	uint		1..NonEmpty	According information about non-empty

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**IDnumber**	char	8 symbols	The device's id number
3	**DateTime**	char	See remark: [*30]	The date and time of record creation

Example	
Binary log	
Request	01 30 30 32 3E 70 30 30 37 3E 31 09 31 09 05 30 32 38 3E 03
Answer	01 30 30 34 3F 70 30 30 37 3E 30 09 44 54 36 33 36 35 31 38 09 30 37 2D 30 31 2D 32 30 31 39 20 31 34 3A 32 38 09 04 80 80 80 80 86 9A 80 80 05 30 3B 37 38 03
Request	1[\t]1[\t]
Answer	0[\t]DT636518 [\t]07-01-2019[\t]14:28[\t]

4.65.4. Ask for fiscal memory number

Request					
	Name	Type	Opt	Value	Description
1	**Type**	uint		2	Type of record**:** Fiscal memory
2	**Record**	uint		1...NonEmpty	According information about non-empty

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**FMNumber**	char	8 symbols	The fiscal memory number
3	**DateTime**	char	See remark: [*30]	The date and time of record creation

Example	
Binary log	
Request	01 30 30 32 3E 6E 30 30 37 3E 32 09 31 09 05 30 32 38 3D 03
Answer	01 30 30 34 3F 6E 30 30 37 3E 30 09 30 32 36 33 36 35 31 38 09 30 37 2D 30 31 2D 32 30 31 39 20 31 34 3A 32 38 09 04 80 80 80 80 86 9A 80 80 05 30 3B 34 30 03
Request	2[\t]1[\t]
Answer	0[\t]02636518[\t]07-01-2019 14:28[\t]

4.65.5. Ask for date of fiscalization

Request

	Name	Type	Opt	Value	Description
1	**Type**	uint		3	Type of record**:** The date of fiscalization
2	**Record**	uint		1...NonEmpty	According information about non-empty

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**DateTime**	char	See remark: [*30]	The date and time of record creation

Example

Binary log	
Request	01 30 30 32 3E 75 30 30 37 3E 33 09 31 09 05 30 32 39 35 03
Answer	01 30 30 34 36 75 30 30 37 3E 30 09 30 37 2D 30 31 2D 32 30 31 39 20 31 34 3A 32 38 09 04 80 80 80 80 86 9A 80 80 05 30 39 39 36 03
Request	3[\t]1[\t]
Answer	0[\t]07-01-2019[\t]14:28[\t]

4.65.6. Ask for TAX number changes

Request

	Name	Type	Opt	Value	Description
1	**Type**	uint		4	Type of record**:** TAX number changes
2	**Record**	uint		1...NonEmpty	According information about non-empty

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**TAXnumber**	char	Up to 13 symbols	TAX number
3	**DateTime**	char	See remark: [*30]	The date and time of record creation

Example

Binary log	
Request	01 30 30 32 3D 77 30 30 37 3E 34 09 09 05 30 32 36 36 03
Answer	01 30 30 33 39 77 30 30 37 3E 30 09 31 09 31 09 04 80 80 80 80 86 9A 80 80 05 30 36 3E 3E 03
Request	4[\t]1[\t]
Answer	0[\t]000713391[\t]07-01-2019[\t]14:28[\t]

4.65.7. Ask for vat rate changes

Request

	Name	Type	Opt	Value	Description
1	**Type**	uint		5	Type of record**:** Vat rates
2	**Record**	uint		1...NonEmpty	According information about non-empty

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
1	**TaxA**	uint	0.00...100.00	• **0.00**...**99.99** – enabled; • **100.00** – disabled;
2	**TaxB**	uint	0.00...100.00	• **0.00**...**99.99** – enabled; • **100.00** – disabled;
3	**TaxC**	uint	0.00...100.00	• **0.00**...**99.99** – enabled; • **100.00** – disabled;
4	**TaxD**	uint	0.00...100.00	• **0.00**...**99.99** – enabled; • **100.00** – disabled;
5	**TaxE**	uint	0.00...100.00	• **0.00**...**99.99** – enabled; • **100.00** – disabled;
6	**TaxF**	uint	0.00...100.00	• **0.00**...**99.99** – enabled; • **100.00** – disabled;
7	**TaxG**	uint	0.00...100.00	• **0.00**...**99.99** – enabled; • **100.00** – disabled;
8	**TaxH**	uint	0.00...100.00	• **0.00**...**99.99** – enabled; • **100.00** – disabled;
9	**nZrep**	uint	0...9999	Z report number from which they are active.
10	**decimal_point**	uint	0 or 2	• **'0'** - work with integer prices; • **'2'** - work with fractional prices;
11	**DateTime**	char	See remark: [*30]	The date and time of record creation

Example

Binary log

Request	01 30 30 32 3E 7A 30 30 37 3E 35 09 31 09 05 30 32 39 3C 03
Answer	01 30 30 37 3C 7A 30 30 37 3E 30 09 30 2E 30 30 09 32 30 2E 30 30 09 32 30 2E 30 30 09 39 2E 30 30 09 31 30 30 2E 30 30 09 31 30 30 2E 30 30 09 31 30 30 2E 30 30 09 31 09 32 09 30 37 2D 30 31 2D 32 30 31 39 20 31 34 3A 32 38 09 04 80 80 80 80 86 9A 80 80 05 31 32 34 32 03

Request	5[\t]1[\t]
Answer	0[\t]0.00[\t]0.00[\t]20.00[\t]9.00[\t]100.00[\t]100.00[\t]100.00[\t]100.00[\t]1[\t]2[\t]07-01-2019 14:28[\t]

4.65.8. Ask for memory resetting events

Request					
	Name	Type	Opt	Value	Description
1	**Type**	uint		6	Type of record**:** memory resetting event
2	**Record**	uint		1...NonEmpty	According information about non-empty

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**nZrep**	uint	0...9999	Z report number when the event occurred
3	**DateTime**	char	See remark: [*30]	The date and time of record creation

Example	
Binary log	
Request	01 30 30 32 3E 7C 30 30 37 3E 36 09 31 09 05 30 32 39 3F 03
Answer	01 30 30 34 39 7C 30 30 37 3E 30 09 31 31 09 32 35 2D 30 31 2D 32 30 31 39 20 31 34 3A 30 30 09 04 80 80 80 80 86 9A 80 80 05 30 3A 30 31 03
Request	6[\t]1[\t]
Answer	0[\t]11[\t]25-01-2019 14:00[\t]

4.65.9. Ask for NRA registrations events

Request					
	Name	Type	Opt	Value	Description
1	**Type**	uint		7	Type of record**:** NRA registrations events
2	**Record**	uint		1...NonEmpty	According information about non-empty

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**RegID**	char	Up to 20 symbols	NRA registration number
3	**DateTime**	char	See remark: [*30]	The date and time of record creation

Example	
Binary log	
Request	01 30 30 32 3E 7E 30 30 37 3E 37 09 31 09 05 30 32 3A 32 03
Answer	01 30 30 34 3D 7E 30 30 37 3E 30 09 31 34 36 36 38 32 09 30 37 2D 30 31 2D 32 30 31 39 20 31 34 3A 33 32 09 04 80 80 80 80 86 9A 80 80 05 30 3A 3E 35 03
Request	7[\t]1[\t]
Answer	0[\t]146682[\t]07-01-2019 14:32[\t]

4.65.10. Ask for NRA unregistrations events

Request					
	Name	Type	Opt	Value	Description
1	**Type**	uint		8	Type of record**:** NRA unregistrations events
2	**Record**	uint		1...NonEmpty	According information about non-empty

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
3	**DateTime**	char	See remark: [*30]	The date and time of record creation

4.65.11. Ask for EJ (KLEN) changes

Request					
	Name	Type	Opt	Value	Description
1	**Type**	uint		9	Type of record**:** EJ (KLEN) changes
2	**Record**	uint		1...NonEmpty	According information about non-empty

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**lastRecNumAtOpen**	int	0...9999999	Last receipt number when EJ opened
3	**lastZRepAtOpen**	int	0...9999	Last Z number when EJ opened
4	**DateTime**	char	See remark: [*30]	The date and time of record creation
5	**lastRecNumAtClose**	int	0...9999999	Last receipt number when EJ closed or empty if EJ is not closed
6	**lastZRepAtClose**	int	0...9999	Last Z number when EJ closed or empty if EJ is not closed
7	**DateTime**	char	See remark: [*30]	The date and time of record creation or empty if EJ is not closed

Example	
Binary log	
Request	01 30 30 32 3E 83 30 30 37 3E 39 09 32 09 05 30 32 3A 3A 03
Answer	01 30 30 36 39 83 30 30 37 3E 30 09 31 36 31 35 38 09 33 39 09 30 39 2D 30 37 2D 32 30 31 39 20 30 39 3A 35 35 09 31 36 38 37 33 09 39 39 09 33 30 2D 30 39 2D 32 30 31 39 20 31 36 3A 30 36 09 04 80 80 80 80 86 9A 80 80 05 30 3F 3E 37 03
Request	9[\t]2[\t]
Answer	0[\t]16158[\t]39[\t]09-07-2019 09:55[\t]16873[\t]99[\t]30-09-2019 16:06[\t]

4.66. Command 127 (7Fh) Stamp operations [*32]

Request					
	Name	Type	Opt	Value	Description
1	**Type**	uint		0...1	Type of operation: • **'0'** - Print stamp; • **'1'** - Rename loaded stamp with command 203;
2	**Name**	char		Up to 12 symbols	Name of stamp as filename in format 8.3

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 33 30 26 30 30 37 3F 30 09 73 73 73 09 05 30 33 35 3F 03
Answer	01 30 30 33 3B 26 30 30 37 3F 2D 31 30 30 30 30 34 09 04 80 80 80 80 86 9A 80 80 05 30 37 34 3E 03
Human oriented log	
Request	0[\t]sss[\t]
Answer	-100004[\t]

4.67. Command 135 (87h) Modem information

4.67.1. Modem information – option ‘s’ - Read IMEI of the modem

Request					
	Name	Type	Opt	Value	Description
1	**Option**	char		s	Read the IMEI of the modem. *Answer(1)*

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**IMEI**	char	Up to 15 symbols	IMEI number of the modem.

4.67.2. Modem information – option ‘i’ - Read the IMSI of the SIM card

Request					
	Name	Type	Opt	Value	Description
1	**Option**	char		i	Read the IMSI of the SIM card. *Answer(2)*

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**IMSI**	char	Up to 15 symbols	IMSI number of the SIM card.

4.67.3. Modem information – option ‘M’ - Modem status

Request

	Name	Type	Opt	Value	Description
1	**Option**	char		M	Modem status. Returns the last state of the modem. *Answer(3)*

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**SignalLevel**	uint	0...100	GSM Signal level in percentage.
3	**IMEI**	char	Up to 15 symbols	IMEI number of the modem.
4	**IMSI**	char	Up to 15 symbols	IMSI number of the SIM card.
5	**MobileOperatorName**	char		Mobile operator name.

Example

Binary log	
Request	01 30 30 32 3C 57 30 30 38 37 4D 09 05 30 32 34 3F 03
Answer	01 30 30 36 35 57 30 30 38 37 30 09 36 34 09 38 36 38 39 39 37 30 33 36 32 37 35 30 30 34 09 32 38 34 30 31 33 39 31 31 35 32 33 36 37 31 09 4D 6F 62 69 6C 74 65 6C 20 45 41 44 09 04 80 80 80 80 86 9A 80 80 05 31 31 32 32 03
Human oriented log	
Request	M[\t]
Answer	0[\t]64[\t]868997036275004[\t]284013911523671[\t]Mobiltel EAD[\t]

4.68. Command 140 (8Ch) Defining and reading clients

4.68.1. Clients programming – option 'I' - Clients information

Request

	Name	Type	Opt	Value	Description
1	**Option**	char		I	Clients information.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**Total**	uint	1000	Total count of the programmable clients.
3	**Prog**	uint	1...1000	Total count of the programmed clients.
4	**NameLen**	uint	36	Maximum length of client name.

Example

Binary log	
Request	01 30 30 32 3C 58 30 30 38 3C 49 09 05 30 32 35 31 03
Answer	01 30 30 33 3F 58 30 30 38 3C 30 09 31 30 30 30 09 30 09 33 36 09 04 80 80 80 80 86 9A 80 80 05 30 37 3D 35 03
Human oriented log	
Request	I[\t]
Answer	0[\t]1000[\t]0[\t]36[\t]

4.68.2. Clients programming – option ‘P’ - Programming clients

Request					
	Name	Type	Opt	Value	Description
1	**Option**	char		P	Clients programming.
2	**FIRM**	uint		1...1000	Client number, index of record.
3	**Name**	char		Up to 36 symbols	Client's name.
4	**TypeTAXN**	uint		0...3	Type of TAXN: • **'0'** - BULSTAT; • **'1'** - EGN; • **'2'** - LNCH; • **'3'** - service number;
5	**TAXN**	char		9...13 symbols	Client's tax number.
6	**RecName**	char		Up to 36 symbols	Receiver's name.
7	**VATN**	char		Up to 14 symbols	VAT number of the client.
8	**Addr1**	char		Up to 36 symbols	Client's address – line 1.
9	**Addr2**	char		Up to 36 symbols	Client's address – line 2.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 35 3B 9C 30 30 38 3C 50 09 31 09 4E 41 4D 45 09 31 09 39 35 30 33 32 31 36 36 31 36 09 52 45 43 45 49 56 45 52 20 4E 41 4D 45 09 09 41 44 44 52 31 09 41 44 44 52 32 09 05 30 3C 39 3F 03
Answer	01 30 30 33 35 9C 30 30 38 3C 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 39 3A 03
Human oriented log	
Request	P[\t]1[\t]NAME[\t]1[\t]9503216616[\t]RECEIVER NAME[\t][\t]ADDR1[\t]ADDR2[\t]
Answer	0[\t]

4.68.3. Clients programming – option ‘D’ - Client deleting

Request					
	Name	Type	Opt	Value	Description
1	**Option**	char		D	Client deleting.
2	**firstFIRM**	uint		1...1000	First client to delete. If this parameter has value 'A', all clients will be deleted (lastFIRM must be empty).
3	**lastFIRM**	uint	●	1...1000	Last client to delete.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 32 3F 9D 30 30 38 3C 44 09 31 09 09 05 30 32 3D 37 03
Answer	01 30 30 33 35 9D 30 30 38 3C 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 39 3B 03
Human oriented log	
Request	D[\t]1[\t][\t]
Answer	0[\t]

4.68.4. Clients programming – option ‘R’ - Reading client data

Request

	Name	Type	Opt	Value	Description
1	**Option**	char		R	Reading client data.
2	**FIRM**	uint		1...1000	Client number.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**FIRM**	uint	1...1000	Client number, index of record.
3	**TAXN**	char	9...13 symbols	Client's tax number.
4	**TypeTAXN**	uint	0...3	Type of TAXN: • **'0'** - BULSTAT; • **'1'** - EGN; • **'2'** - LNCH; • **'3'** - service number;
5	**VATN**	char	Up to 14 symbols	VAT number of the client.
6	**Name**	char	Up to 36 symbols	Client's name.
7	**RecName**	char	Up to 36 symbols	Receiver's name.
8	**Addr1**	char	Up to 36 symbols	Client's address – line 1.
9	**Addr2**	char	Up to 36 symbols	Client's address – line 2.

Example

Binary log	
Request	01 30 30 32 3E 9F 30 30 38 3C 52 09 31 09 05 30 32 3D 3D 03
Answer	01 30 30 37 32 9F 30 30 38 3C 30 09 31 09 39 35 30 33 32 31 36 36 31 36 09 31 09 20 20 20 20 20 20 20 20 20 20 20 20 20 20 09 4E 41 4D 45 09 52 45 43 45 49 56 45 52 20 4E 41 4D 45 09 41 44 44 52 31 09 41 44 44 52 32 09 04 80 80 80 80 86 9A 80 80 05 31 32 35 3F 03
Human oriented log	
Request	R[\t]1[\t]
Answer	0[\t]1[\t]9503216616[\t]1[\t][\t]NAME[\t]RECEIVER NAME[\t]ADDR1[\t]ADDR2[\t]

4.68.5. Clients programming – option ‘F’ - Data about the first found programmed client

Request

	Name	Type	Opt	Value	Description
1	**Option**	char		I	Returns data about the first found programmed client.
2	**FIRM**	uint	●	1...1000	Seek from given client number (Default : 1)

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**FIRM**	uint	1...1000	Client number, index of record.
3	**TAXN**	char	9...13 symbols	Client's tax number.
4	**TypeTAXN**	uint	0...3	Type of TAXN: • **'0'** - BULSTAT; • **'1'** - EGN; • **'2'** - LNCH; • **'3'** - service number;
5	**VATN**	char	Up to 14 symbols	VAT number of the client.
6	**Name**	char	Up to 36 symbols	Client's name.
7	**RecName**	char	Up to 36 symbols	Receiver's name.
8	**Addr1**	char	Up to 36 symbols	Client's address – line 1.
9	**Addr2**	char	Up to 36 symbols	Client's address – line 2.

Example

Binary log

Request	01 30 30 32 3E A4 30 30 38 3C 46 09 31 09 05 30 32 3D 36 03
Answer	01 30 30 37 32 A4 30 30 38 3C 30 09 31 09 39 35 30 33 32 31 36 36 31 36 09 31 09 20 20 20 20 20 20 20 20 20 20 20 20 20 20 09 4E 41 4D 45 09 52 45 43 45 49 56 45 52 20 4E 41 4D 45 09 41 44 44 52 31 09 41 44 44 52 32 09 04 80 80 80 80 86 9A 80 80 05 31 32 36 34 03

Human oriented log

Request	F[\t]1[\t]
Answer	0[\t]1[\t]9503216616[\t]1[\t][\t]NAME[\t]RECEIVER NAME[\t]ADDR1[\t]ADDR2[\t]

4.68.6. Clients programming – option 'L' - Data about the last found programmed client

Request

	Name	Type	Opt	Value	Description
1	**Option**	char		L	Returns data about the last found programmed client.
2	**FIRM**	uint	●	1...1000	Seek from given client number.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**FIRM**	uint	1...1000	Client number, index of record.
3	**TAXN**	char	9...13 symbols	Client's tax number.
4	**TypeTAXN**	uint	0...3	Type of TAXN: • **'0'** - BULSTAT; • **'1'** - EGN; • **'2'** - LNCH; • **'3'** - service number;
5	**VATN**	char	Up to 14 symbols	VAT number of the client.
6	**Name**	char	Up to 36 symbols	Client's name.
7	**RecName**	char	Up to 36 symbols	Receiver's name.
8	**Addr1**	char	Up to 36 symbols	Client's address – line 1.
9	**Addr2**	char	Up to 36 symbols	Client's address – line 2.

Example

Binary log

Request	01 30 30 32 3E A5 30 30 38 3C 4C 09 31 09 05 30 32 3D 3D 03
Answer	01 30 30 37 32 A5 30 30 38 3C 30 09 31 09 39 35 30 33 32 31 36 36 31 36 09 31 09 20 20 20 20 20 20 20 20 20 20 20 20 20 20 09 4E 41 4D 45 09 52 45 43 45 49 56 45 52 20 4E 41 4D 45 09 41 44 44 52 31 09 41 44 44 52 32 09 04 80 80 80 80 86 9A 80 80 05 31 32 36 35 03

Human oriented log

Request	L[\t]
Answer	0[\t]1[\t]9503216616[\t]1[\t][\t]NAME[\t]RECEIVER NAME[\t]ADDR1[\t]ADDR2[\t]

4.68.7. Clients programming – option ‘N’ - Data for the next found programmed client

Request

	Name	Type	Opt	Value	Description
1	**Option**	char		N	The same command with option 'F' or 'L' must be executed first. This determines whether to get next('F') or previous ('L') client.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**FIRM**	uint	1...1000	Client number, index of record.
3	**TAXN**	char	9...13 symbols	Client's tax number.
4	**TypeTAXN**	uint	0...3	Type of TAXN: • **'0'** - BULSTAT; • **'1'** - EGN; • **'2'** - LNCH; • **'3'** - service number;
5	**VATN**	char	Up to 14 symbols	VAT number of the client.
6	**Name**	char	Up to 36 symbols	Client's name.
7	**RecName**	char	Up to 36 symbols	Receiver's name.
8	**Addr1**	char	Up to 36 symbols	Client's address – line 1.
9	**Addr2**	char	Up to 36 symbols	Client's address – line 2.

Example

Binary log

Request	01 30 30 32 3C AB 30 30 38 3C 4E 09 05 30 32 3A 39 03
Answer	01 30 30 37 34 AB 30 30 38 3C 30 09 32 09 39 32 30 32 32 35 32 32 31 32 09 31 09 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 09 4E 41 4D 45 20 32 09 52 45 43 45 49 56 45 52 20 4E 41 4D 45 09 41 44 44 52 31 09 41 44 44 52 32 09 04 80 80 80 80 86 9A 80 80 05 31 32 3B 34 03

Human oriented log

Request	N[\t]
Answer	0[\t]2[\t]9202252212[\t]1[\t][\t]NAME 2[\t]RECEIVER NAME[\t]ADDR1[\t]ADDR2[\t]

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

4.68.8. Clients programming – option ‘T’ - Find a client by tax number

Request

	Name	Type	Opt	Value	Description
1	**Option**	char		T	Find a client by tax number.
2	**TAXN**	char		9...13	Client's tax number.

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**FIRM**	uint	1...1000	Client number, index of record.
3	**TAXN**	char	9...13 symbols	Client's tax number.
4	**TypeTAXN**	uint	0...3	Type of TAXN: • **'0'** - BULSTAT; • **'1'** - EGN; • **'2'** - LNCH; • **'3'** - service number;
5	**VATN**	char	Up to 14 symbols	VAT number of the client.
6	**Name**	char	Up to 36 symbols	Client's name.
7	**RecName**	char	Up to 36 symbols	Receiver's name.
8	**Addr1**	char	Up to 36 symbols	Client's address – line 1.
9	**Addr2**	char	Up to 36 symbols	Client's address – line 2.

Example

Binary log

Request	01 30 30 33 37 AC 30 30 38 3C 54 09 39 35 30 33 32 31 36 36 31 36 09 05 30 34 3B 3C 03
Answer	01 30 30 37 32 AC 30 30 38 3C 30 09 31 09 39 35 30 33 32 31 36 36 31 36 09 31 09 20 20 20 20 20 20 20 20 20 20 20 20 20 20 09 4E 41 4D 45 09 52 45 43 45 49 56 45 52 20 4E 41 4D 45 09 41 44 44 52 31 09 41 44 44 52 32 09 04 80 80 80 80 86 9A 80 80 05 31 32 36 3C 03

Human oriented log

Request	T[\t]
Answer	0[\t]1[\t]9503216616[\t]1[\t][\t]NAME[\t]RECEIVER NAME[\t]ADDR1[\t]ADDR2[\t]

4.68.9. Clients programming – option ‘X’ - Find the first not programmed client

Request

	Name	Type	Opt	Value	Description
1	**Option**	char		X	Find the first not programmed client.
2	**FIRM**	uint	●	1...1000	Seek from given client number. (Default : 1)

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**FIRM**	uint	1...1000	Client number.

Example

Binary log	
Request	01 30 30 32 3D AD 30 30 38 3C 58 09 09 05 30 32 3B 3F 03
Answer	01 30 30 33 37 AD 30 30 38 3C 30 09 33 09 04 80 80 80 80 86 9A 80 80 05 30 36 3E 39 03
Human oriented log	
Request	X[\t][\t]
Answer	0[\t]3[\t]

4.68.10. Clients programming – option ‘x’ - Find the last not programmed client

Request

	Name	Type	Opt	Value	Description
1	**Option**	char		x	Operation type.
2	**FIRM**	uint	●	1...1000	Seek from given client number.(Default : 1)

Answer

	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**FIRM**	uint	1...1000	Client number.

Example

Binary log

Request	01 30 30 32 3D AE 30 30 38 3C 78 09 09 05 30 32 3E 30 03
Answer	01 30 30 33 3A AE 30 30 38 3C 30 09 31 30 30 30 09 04 80 80 80 80 86 9A 80 80 05 30 37 37 3B 03

Human oriented log

Request	x[\t][\t]
Answer	0[\t]1000[\t]

4.69. Command 202 (CAh) Customer graphic logo loading. [*32]

Request					
	Name	Type	Opt	Value	Description
1	**Parameter**	char		Up to 72 symbols	Type of operation.

4.69.1. Parameters description

Request					
	Name	Type	Opt	Value	Description
1	**START**				Preparation for data loading.
2	**STOPP**				End of data.
3	**YmFzZTY0ZGF0YQ==**				Base64 coded data of the graphic logo.
4	**POWEROFF**				Shutting down the device.
5	**RESTART**				Device restarting.

Answer (1)				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Answer (2)				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**ChechSum**	char		Sum of decoded base64 data.

Example	
Binary log	
Request	01 30 30 33 32 25 30 30 3C 3A 52 45 53 54 41 52 54 09 05 30 33 3F 33 03
Answer	01 30 30 33 35 25 30 30 3C 3A 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 32 35 03
Human oriented log	
Request	RESTART[\t]
Answer	0[\t]

4.70. Command 203 (CAh) Stamp image loading [*32]

Request					
	Name	Type	Opt	Value	Description
1	**Parameter**	char		Up to 72 symbols	Type of operation.

4.70.1. Parameters description

Request					
	Name	Type	Opt	Value	Description
1	**START**				Preparation for data loading.
2	**STOPP**				End of data.
3	**YmFzZTY0ZGF0YQ==**				Base64 coded data of the graphic logo.

Answer (1)				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Answer (2)				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**CheckSum**	char		Sum of decoded base64 data.

Example	
Binary log	
Request	01 30 30 33 30 86 30 30 3C 3B 53 54 4F 50 50 09 05 30 33 3C 34 03
Answer	01 30 30 33 3D 86 30 30 3C 3B 30 09 30 30 34 30 33 46 37 30 04 80 80 80 80 86 9A 80 80 05 30 38 33 33 03
Human oriented log	
Request	STOPP[\t]
Answer	0[\t]00403F70[\t]

4.71. Command 253 (FDh) Service operations

4.71.1. Entering service password

Request					
	Name	Type	Opt	Value	Description
1	**Option**	char		‘0’	Type of operation
2	**Value**	char		8 digits	Service technician password (blank string disables service mode)

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

4.71.2. Change service password

Request					
	Name	Type	Opt	Value	Description
1	**Option**	char		‘1’	Type of operation
2	**OldPassword**	char		8 digits	Old technician password
3	**NewPassword**	char		8 digits	New technician password

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

4.71.3. Close the current EJ

Request					
	Name	Type	Opt	Value	Description
1	**Option**	char		‘2’	Type of operation - service jumper is needed

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

4.71.4. Factory setting of configuration parameters

Request					
	Name	Type	Opt	Value	Description
1	**Option**	char		‘3’	Type of operation - service jumper is needed

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

4.71.5. Clear errors from NRA server communication. Unblock the blocked device.

Request					
	Name	Type	Opt	Value	Description
1	**Option**	char		‘4’	Type of operation - service jumper is needed

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

4.71.6. Send all unsent documents to the NRA servers

Request					
	Name	Type	Opt	Value	Description
1	**Option**	char		‘5’	Type of operation – possible execution of the command - once every 5 minutes

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

4.72. Command 255 (FFh) Programming

4.72.1. Read parameters

Request					
	Name	Type	Opt	Value	Description
1	**Name**	char		Up to 72 symbols	Parameter name.
2	**Index**	uint		0...9999	Used for index if variable is array. For variable that is not array can be left blank. Default: 0;
3	**Value**	char/ uint/ int		According parameter	This parameter is blank.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.
2	**Value**	char/uint/ int	According parameter	Current value of the variable.

Example	
Binary log	
Request	01 30 30 33 39 28 30 30 3F 3F 41 75 74 6F 50 6F 77 65 72 4F 66 66 09 09 09 05 30 36 3B 33 03
Answer	01 30 30 33 37 28 30 30 3F 3F 30 09 31 09 04 80 80 80 80 86 9A 80 80 05 30 36 36 3C 03
Human oriented log	
Request	AutoPowerOff[\t][\t][\t]
Answer	0[\t]1[\t]

4.72.2. Write parameters

Request					
	Name	Type	Opt	Value	Description
1	**Name**	char		Up to 72 symbols	Parameter name.
2	**Index**	uint		0...9999	Used for index if variable is array. For variable that is not array can be left blank. Default: 0;
3	**Value**	char/ uint/ int		According parameter	The value to be set.

Answer				
	Name	Type	Value	Description
1	**ErrorCode**	int	-999999...0	Indicates an error code. If command passed, **ErrorCode** is 0.

Example	
Binary log	
Request	01 30 30 33 3A 2F 30 30 3F 3F 41 75 74 6F 50 6F 77 65 72 4F 66 66 09 09 32 09 05 30 36 3E 3D 03
Answer	01 30 30 33 35 2F 30 30 3F 3F 30 09 04 80 80 80 80 86 9A 80 80 05 30 36 33 37 03
Human oriented log	
Request	AutoPowerOff[\t][\t]2[\t]
Answer	0[\t]

4.72.3. Parameters description

Parameters description						
Name	Type	Value	Default	Index	Read only	Description
FpComBaudRate	uint	0...9[*17]	9	N	N	Baud rate of COM port for communication with PC.
AutoPaperCutting[*16]	uint	0...1	1	N	N	Permission/rejection of the automatic cutting of paper after each receipt: • **'0'** – rejected; • **'1'** – permitted;
PaperCuttingType[*16]	uint	0...1	0	N	N	Paper cutting type: • **'0'** – full; • **'1'** – partial;
BarCodeHeight	uint	1...10	1	N	N	Barcode height from '1' (7mm) to '10' (70mm).
BarcodeName	uint	0...1	1	N	N	Enable/Disable printing of the barcode data: • **'0'** – disabled; • **'1'** – enabled;
ComPortDevice	uint	0...3[*24]	0	0...1	N	Assign peripheral device to COM port: • Index=0 – COM1; • Index=1 – COM2;
ComPortBaudRate	uint	0...9[*17]	0	0...1	N	Baud rate of COM port that has peripheral device assigned: • Index=0 – COM1; • Index=1 – COM2;
ComPortProtocol[*26]	uint	0...2[*25]	0	0...1	N	Protocol for communication with peripheral device assigned COM port: • Index=0 – COM1; • Index=1 – COM2;
MainInterfaceType[*9]	uint	0...4	0	N	N	PC interface type: • **'0'** - auto select; • **'1'** – RS232; • **'2'** – BLUETOOTH; • **'3'** – USB; • **'4'** – LAN;
TimeOutBeforePrintFlush	uint	1...999999999	200	N	N	Time out between commands before start auto print (in milliseconds).
WorkBatteryIncluded	uint	0..1	0	N	N	Device works with battery on main supply: • **'0'** - disable; • **'1'** - enable;
Dec2xLineSpacing	uint	0...5	3	N	N	Decrease the space between text lines. Greater values causes less line spacing.
Dec2xLineSpacingVerical	uint	0...5	3	N	N	Decrease the space between vertical text lines. Greater values causes less line spacing.
PrintFontType[*11]	uint	0...1	0	N	N	Printer font type: • **'0'** - coarser with a small line spacing; • **'1'** - smaller, with greater spacing between rows;
FooterEmptyLines	uint	0...10	According device model	N	N	Number of blank lines for proper paper cutting.
HeaderMinLines	uint	0...10	According device model	N	N	Minimum number of lines from the header after printing the footer.
LogoPrintAfterFooter	uint	0...1	0	N	N	Print the logo after rows to push the paper: • **'0'** – no; • **'1'** – yes;
EnableNearPaperEnd	uint	0...1	1	N	N	Handling of near paper end: • **'0'** – handling; • **'1'** - no handling;
DateFromNAPServDisable	uint	0...1	0	N	N	Synchronize date/time from the NRA server: • **'0'** – sync; • **'1'** - does not sync;

Parameters description						
AutoPowerOff	uint	0...240	10	N	N	Minutes to automatically turn off device if it is idle and working on battery: • **'0'** – disable; • **'1-240'** - time in minutes;
BkLight_AutoOff	uint	0...240	1	N	N	Minutes to automatically turn off backlight of the display if device is idle and working on battery: • **'0'** – disable; • **'1-240'** – time in minutes;
PinpadComPort	uint	0,1,2,4	0	N	N	Number of COM port for communication with pinpad: • **'0'** - Not connected; • **'1'** – COM1; • **'2'** – COM2; • **'4'** – Bluetooth;
PinpadComBaudRate	uint	0...9[*17]	0	N	N	Baud rate of COM port that has pinpad device assigned.
PinpadType	uint	0...3	0	N	N	Type of pinpad: • **'0'** – unknown; • **'1'** – BORICA; • **'2'** – UBB; • **'3'** – DSK;
PinpadConnectionType	uint	0...1	0	N	N	Type of connection between cash register and bank server: • **'0'** – GPRS; • **'1'** – LAN;
PinpadReceiptCopies	uint	0...3	0	N	N	Copies of the receipt from pinpad.
PinpadReceiptInfo	uint	0...1	0	N	N	Where to print pinpad receipt: • **'0'** - separate slip; • **'1'** - included in fiscal slip;
PinpadPaymentMenu[*10]	uint	0...1	0	N	N	Function of PY2 key in registration. Works only if PinpadType = 1. • **'0'** - payment with card with pinpad; • **'1'** - menu for payment with pinpad (card and loyalty scheme);
PinpadLoyaltyPayment[*10]	uint	0...1	0	N	N	Function of PY4 key. Works only if PinpadType = 1. • **'0'** - payment PY4; • **'1'** - payment with pinpad with loyalty scheme;
PinpadShortRec	uint	0...1	0	N	N	Receipt type from pinpad: • **'0'** - normal; • **'1'** - short;
BthEnable	uint	0...1	0	N	N	Turn on/off Bluetooth module: • **'0'** - off; • **'1'** - on;
BthDiscoverability	uint	0...1	0	N	N	Turn on/off Bluetooth device discoverability: • **'0'** - non-discoverable; • **'1'** - discoverable;
BthPairing	uint	0...2	0	N	N	Bluetooth pairing type: • **'0'** – unsecure; • **'1'** – reset and save; • **'2'** – reset;
BthPinCode	char	Up to 16 symbols	0000	N	N	Pin code for Bluetooth pairing.
BthVersion	char	Up to 16 symbols	According device model	N	Y	Version of the Bluetooth module.
BthAddress	char	Up to 16 symbols	According device model	N	Y	Bluetooth module address.
EcrLogNumber[*10]	uint	1...99999	1	N	N	Logical number in the workplace.
EcrExtendedReceipt	uint	0...1	0	N	N	Enable extended printout of the fiscal receipts: • **'0'** – disable; • **'1'** – enable;

Parameters description						
EcrDoveriteli	uint	0...1	0	N	N	Work with business partners. When is enabled, in one receipt only one business partners can exist. • **'0'** – disable; • **'1'** – enable;
EcrWithoutPasswords[*10]	uint	0...1	0	N	N	Work without passwords: • **'0'** - disable; • **'1'** - enable;
EcrAskForPassword[*10]	uint	0...1	0	N	N	Require password after each receipt: • **'0'** – disable; • **'1'** - enable;
EcrAskForVoidPassword[*10]	uint	0...1	0	N	N	Require password for void operations: • **'0'** - disable; • **'1'** - enable;
EcrConnectedOperReport	uint	0...1	0	N	N	When making Z-report, automatically make "Operator report". • **'0'** – disable; • **'1'** – enable;
EcrConnectedDeptReport	uint	0...1	0	N	N	When making Z-report, automatically make "Report by Departments": • **'0'** – disable; • **'1'** – enable;
EcrConnectedPluSalesReport	uint	0...1	0	N	N	When making Z-report, automatically make "Report by PLU with turnovers": • **'0'** – disable; • **'1'** – enable;
EcrConnectedGroupsReport	uint	0...1	0	N	N	When making Z-report, automatically make "Group report": • **'0'** - disable; • **'1'** - enable;
EcrConnectedCashReport	uint	0...1	0	N	N	When making Z-report, automatically make "ECR report": • **'0'** - disable; • **'1'** - enable;
EcrUserPeriodReports	uint	0...1	0	N	N	Periodic reports: • **'0'** - disable; • **'1'** - enable;
EcrPluDailyClearing	uint	0...1	0	N	N	When making Z-report, automatically clear PLU turnovers: • **'0'** – disable; • **'1'** – enable;
EcrSafeOpening	uint	0...1	0	N	N	Open drawer on every total: • **'0'** - disable; • **'1'** - enable;
EcrScaleBarMask[*10]	char	Up to 10 symbols	0	N	N	Text up to 10 symbols. If second number of the weight barcode not match any of the symbols in this string, barcode will be interpreted as normal barcode.
EcrNumberBarcode[*10]	uint	1...4	1	N	N	Count of used barcodes for each programmed article.
RegModeOnIdle[*10]	uint	1...2147483647	10000	N	N	Time to clear display after last receipt in milliseconds.
FlushAtEndOnly[*10]	uint	0...1	0	N	N	The receipt is printed after last payment.
EcrMidnightWarning[*10]	uint	0...240	0	N	N	Minutes before midnight, when device starts showing warning for Z report.
EcrMandatorySubtotal[*10]	uint	0...1	0	N	N	The operator must press STL key before payment: • **'0'** - disable; • **'1'** - enable:
Seller[*10]	char	0...36	empty	N	N	Name of the seller. Used in invoices.
AutoMonthReport[*10]	uint	0...1	1	N	N	Flag for a monthly report suggesting: • **'0'** – disable; • **'1'** - enable;
AutoMonthReportDubl[*10]	uint	0...1	1	N	N	Flag for monthly report dublicate suggesting. ECR only. • **'0'** - no;

Parameters description						
						• **'1'** - yes:
EcrUnsentWarning[*10]	uint	0...24	0	N	N	Warning for unsent documents from XX hours. The value must be set in hours before device will be blocked. • **'0'** - no warnings messages • **'1-24'** - warning will appear before the value of EcrUnsentWarning of device blocking.
CurrNameLocal	char	Up to 3 chars	'JIB'	N	N	Local currency name.
CurrNameForeign	char	Up to 3 chars	'EUR'	N	N	Foreign currency name.
ExchangeRate	uint	1... 999999999	'195583'	N	N	Exchange rate (decimal point is before last five digits).
Unit_name	char	Up to 6 symbols	See remark: [*15]	0...19	N	An array of unit names. Index 0 is for line 1...Index 19 is for line 20.
Header	char	Up to XX symbols	See remark: [*14]	0...9	N	An array of header lines. Index 0 is for line 1, Index 9 is for line 10. XX depends on print columns. • XX = 42,48,64[*11] • XX = 42[*8] • XX = 32[*12].
Footer	char	Up to XX symbols	Empty	0...9	N	An array of footer lines. Index 0 is for line 1, Index 9 is for line 10. XX depends on print columns. • XX = 42,48,64[*11] • XX = 42[*8] • XX = 32[*12].
RecText	char	Up to XX-2 symbols	Empty	0...17	N	An array of additional text lines printed after the footer, with a '#' sign on both sides of the line. The lines are printed only on the fiscal receipts. Index 0 is for line 1, Index 17 is for line 18. XX depends on print columns. • XX = 42,48,64[*11] • XX = 42[*8] • XX = 32[*12].
OperName	char	Up to 32 symbols	'ОПЕРАТОР'	0...29	N	An array of operator names. Index 0 is for operator 1, Index 29 is for operator 30.
OperPasw	char	Up to 8 symbols (digits only)	'1'…'30'[*1] '0000'[*1]	0...29	N	An array of operator passwords. Index 0 is for operator 1, Index 29 is for operator 30.
PayName	char	Up to 16 symbols	See remark: [*13]	0...5	N	An array of payment names.
Payment_forbidden	uint	0...1	0	0...5	N	Forbid the payment: • **'0'** - not forbidden; • **'1'** - forbidden;
DPxx_PluCode[*10]	uint	0...99999	0	0...6[*18] 0...8[*19] 0...36[*20]	N	Number of PLU assigned to shortcut key: • **'0'** - Key is disabled; • **'1-99999'** - assigning PLU;
KeyNDB_value[*10]	uint	0...999999999	0	N	N	Value for surcharge by sum. Value is in cents.
KeyNDB_percentage[*10]	uint	0... 9999	0	N	N	Percentage for percentage surcharge. Value is in hundredths (0.01) of a percent
KeyOTS_value[*10]	uint	0...999999999	0	N	N	Value for value discount. Value is in cents.
KeyOTS_percentage[*10]	uint	0...9999	0	N	N	Percentage for percentage discount. Value is in hundredths (0.01) of a percent.
KeyNDB_forbidden[*10]	uint	0...1	0	N	N	Forbid the surcharge key: • **'0'** – not forbidden; • **'1'** - forbidden;
KeyOTS_forbidden[*10]	uint	0...1	0	N	N	Forbid the discount key: • **'0'** – not forbidden; • **'1'** - forbidden;

Parameters description						
ServPasw	char	Up to 8 symbols	30	N	N	Password of the Service man.
ServMessage	char	Up to value of parameter PrintColumns	empty	0...9	N	An array of text lines. Index 0 is for line 1, Index 9 is for line 10. Message that will be printed when "ServDate" is reached.
ServiceDate	char	See remark: [*27]	empty	N	N	Service date.
PrnQuality	uint	0...20	10	N	N	Contrast of Printing.
PrintColumns	uint	42,48,64[*11] 42[*8] 32[*12]	48[*11] 42[*8] 32[*12]	N	N	Number of printer columns.
EmptyLineAfterTotal	uint	0...1	0	N	N	Print empty line after TOTAL line in fiscal receipts: • **'0'** - disable; • **'1'** - enable;
DblHeigh_totalinreg	uint	0...1	0	N	N	Print TOTAL line in fiscal receipts with double height: • **'0'** - disable; • **'1'** - enable;
Bold_payments	uint	0...1	0	N	N	Bold print of the payment names in fiscal receipt: • **'0'** - disable; • **'1'** - enable;
DublReceipts	uint	0...1	0	N	N	Print receipt dublicate: • **'0'** - disable; • **'1'** - enable;
IntUseReceipts	uint	0...9	0	N	N	Number of internal receipts.
BarcodePrint	uint	0...1	0	N	N	Print PLU barcode in the receipt: • **'0'** - disable; • **'1'** - enable;
LogoPrint	uint	0...1	0	N	N	Print the logo in the receipt: • **'0'** - disable; • **'1'** - enable;
DoveritelPrint	uint	0...1	0	N	N	Print the department name at the beginning of the receipt: • **'0'** - disable; • **'1'** - enable;
ForeignPrint	uint	0...2	0	N	N	Print total sum in foreign currency: • **'0'** - disable; • **'1'** - enable; • **'2'** - enable and print exchange rate;
VatPrintEnable	uint	0...1	0	N	N	Print VAT rates in the receipt: • **'0'** - disable; • **'1'** - enable;
EnableNearPaperEnd	uint	0...1	1	N	N	Handling of near paper end: • **'0'** - no handling; • **'1'** - handling;
DsblKeyZreport	uint	0...1	0	N	N	Disable Z report generating from the keyboard: • **'0'** - enabled; • **'1'** - disabled;
DsblKeyXreport	uint	0...1	0	N	N	Disable X report generating from the keyboard: • **'0'** - enabled; • **'1'** - disabled;
DsblKeyDiagnostics	uint	0...1	0	N	N	Disable diagnostic info: • **'0'** - enabled; • **'1'** - disabled;
DsblKeyFmReports	uint	0...1	0	N	N	Disable fiscal memory reports: • **'0'** - enabled; • **'1'** - disabled;
DsblKeyJournal	uint	0...1	0	N	N	Disable electronic journal menu:

Parameters description						
						• **'0'** - enabled • **'1'** - disabled
DsblKeyDateTime	uint	0...1	0	N	N	Disable changing the date and time: • **'0'** - enabled; • **'1'** - disabled;
DsblKeyCloseReceipt	uint	0...1	0	N	N	Disable manually closing of the receipt: • **'0'** - enabled; • **'1'** - disabled;
DsblKeyCancelReceipt	uint	0...1	0	N	N	Disable manually cancellation of the receipt: • **'0'** - enabled; • **'1'** - disabled;
ModemModel	uint	0...3	According device model	N	N	Model of the modem: • **'0'** - Quectel M72; • **'1'** - Quectel UC20; • **'2'** - Quectel M66; • **'3'** - Quectel UG96;
SimPin	char	Up to 16 symbols	empty	N	N	PIN code of SIM card.
SimICCID	char	Up to 31 symbols	empty	N	Y	ICC number of the SIM card.
SimIMSI	char	Up to 16 symbols	empty	N	Y	IMSI number of the SIM card.
SimTelNumber	char	Up to 16 symbols	empty	N	Y	MSISDN number of the SIM card.
IMEI	char	Up to 16 symbols	empty	N	Y	IMEI of the modem.
LanMAC	char	Up to 12 symbols	empty	N	N	MAC address of the LAN controller.
DHCPenable	uint	0...1	1	N	N	Enable use of DHCP: • **'0'** - disabled; • **'1'** - enabled;
LAN_IP	char	Up to 15 symbols	empty	N	N	IP address when DHCP is disabled.
LAN_NetMask	char	Up to 15 symbols	empty	N	N	Net mask when DHCP is disabled.
LAN_Gateway	char	Up to 15 symbols	empty	N	N	Default gateway when DHCP is disabled.
LAN_PriDNS	char	Up to 15 symbols	empty	N	N	Primary DNS when DHCP is disabled.
LAN_SecDNS	char	Up to 15 symbols	empty	N	N	Second DNS when DHCP is disabled.
LANport_fpCommands	uint	1...99999	4999	N	N	The number of listening port for PC connection (only for devices with LAN).
WLAN_Enable	uint	0...1	0	N	N	Enable use of WLAN (only for devices with WLAN): • **'0'** - disabled; • **'1'** - enabled;
WLAN_DHCPenable	uint	0...1	1	N	N	Enable use of DHCP: • **'0'** - disabled; • **'1'** - enabled;
WLAN_IP	char	Up to 15 symbols	empty	N	N	IP address when DHCP is disabled.
WLAN_NetMask	char	Up to 15 symbols	empty	N	N	Net mask when DHCP is disabled.
WLAN_PriDNS	char	Up to 15 symbols	empty	N	N	Primary DNS

Parameters description						
WLAN_SecDNS	char	Up to 15 symbols	empty	N	N	Secondary DNS
WLAN_AP_SSID	char	Up to 32 symbols	empty	0...2	N	SSID of WLAN Access point
WLAN_AP_Password	char	Up to 32 symbols	empty	0...2	N	Password of WLAN Access point
WLAN_AP_Security	uint	0, 1, 3	0	0...2	N	Encryption type of WLAN Access point • 0 – Open; • 1 – WEP; • 3 – WPA, WPA2;
LAN_Gateway	char	Up to 15 symbols	empty	N	N	Default gateway when DHCP is disabled.
LAN_PriDNS	char	Up to 15 symbols	empty	N	N	Primary DNS when DHCP is disabled.
LAN_SecDNS	char	Up to 15 symbols	empty	N	N	Second DNS when DHCP is disabled.
WLAN_AP_SSID	char	Up to 32 symbols	empty	0...9	N	SSID of WLAN Access point.
WLAN_AP_Password	char	Up to 32 symbols	empty	0...9	N	Password of WLAN Access point.
NapBlockDateTime	char	See remark: [*27]	empty	N	Y	The date and time after which the device will be blocked due to a lack of connection with the NRA server.
nZreport	uint	1...3650	1	N	Y	Number of current Z-report.
nReset	uint	0...200	0	N	Y	Number of current memory failure.
nVatChanges	uint	0...30	0	N	Y	Number of current VAT change.
nIDnumberChanges	uint	0...1	0	N	Y	Number of current SN changes: • **'0'** - not programmed; • **'1'** - programmed;
nFMnumberChanges	uint	0...1	0	N	Y	Number of current FM number changes: • **'0'** - not programmed; • **'1'** - programmed;
nTAXnumberChanges	uint	0...1	0	N	Y	Number of current TAX number changes: • **'0'** - not programmed; • **'1'** - programmed;
valVat	uint	0...9999 – vat enabled 10000 – vat disabled	0	0...7	Y	Current value of VAT: • Index=0 – vat rate A; • Index=1 – vat rate Б; • Index=2 – vat rate B; • Index=3 – vat rate Г; • Index=4 – vat rate Д; • Index=5 – vat rate E; • Index=6 – vat rate Ж; • Index=7 – vat rate З;
FMDeviceID	uint	0...255	0	0...3	Y	ID of the fiscal memory.
IDnumber	char	2 letters and 6 digits	empty	N	Y	Serial number of the ECR.
FMnumber	char	8 digits	empty	N	Y	Number of FM.
TAXnumber	char	Up to 13 symbols	empty	N	Y	TAX number.
FmWriteDateTime	char	DD-MM-YY HH:MM:SS	empty	N	Y	Date and time of last for writing block in FM.
LastValiddate	char	DD-MM-YY HH:MM:SS	empty	N	Y	Last valid date (written on FM or EJ).
TAXlabel	char	Up to 10 symbols	“ЕИК:”	N	N	TAX number label.

Parameters description						
UNP	char	21 symbols	empty	N	Y	Last printed unique sale number (21 chars "LLDDDDDD-CCCC-DDDDDDD", L[A-Z], C[0-9A-Za-z], D[0-9]);
StornoUNP	char	21 symbols	empty	N	Y	Last printed unique sale number in storno document (21 chars "LLDDDDDD-CCCC-DDDDDDD", L[A-Z], C[0-9A-Za-z], D[0-9]).
Fiscalized	uint	0...1	0	N	Y	Flag that shows if device is fiscalized: • **'0' -** not fiscalized; • **'1'** - fiscalized;
DFR_needed	uint	0...1	0	N	Y	Shows if fiscal receipt is issued after last Z-report: • **'0' -** Z-report is not needed; • **'1' -** Z-report is needed;
DecimalPoint	uint	0 or 2	2	N	Y	Number of symbols after decimal point.
nBon	uint	1...9999999	1	N	Y	Global number of next receipt.
nFBon	uint	1...9999999	1	N	Y	Global number of next fiscal receipt.
nInvoice	uint	0... 9999999999	0	N	Y	Number of next invoice.
InvoiceRangeBeg	uint	0... 9999999999	0	N	Y	Start of the invoice range.
InvoiceRangeEnd	uint	0... 9999999999	0	N	Y	End of the invoice range.
nFBonDailyCount	uint	0... 9999	0	N	Y	Number of fiscal receipts for the day.
nLastFiscalDoc	uint	1...9999999	1	N	Y	Last number of fiscal receipt.
CurrClerk	uint	1...30	1	N	Y	Number of current operator.
EJNewJurnal	uint			N	Y	New EJ.
EJNumber	uint	0...20	0	N	Y	Number of current EJ.
DateLastSucceededSent	char	DD-MM-YY HH:MM:SS	empty	N	Y	Date/time of last connection to the NRA server.
NapRegistered	uint	0...1	0	N	Y	ECR is registered on the NRA server: • **'0' -** not registered; • **'1' -** registered;
DeregOnServer	uint	0...1	0	N	Y	ECR is deregistered on the NRA server: • **'0' -** not deregistered; • **'1' -** deregistered;
ItemGroups_name	char	Up to 32 symbols	“ГРУПА nn”	0...99	N	Name of item group.
Dept_name	char	Up to 72 symbols	“ДП nn”	0...99	N	Name of department.
Dept_price	uint	0... 999999999	0	0...99	N	Programmed price of department.
Dept_vat	uint	1...8	2	0...99	N	VAT group of department.
DHL_Algo[*22]	uint	0...1	0	N	N	Flag that tells if the entered way-bill has to be checked with DHL's algorithm.
EIK_validation[*22]	uint	0...1	1	N	N	Flag that tells if the entered EIK number has to be valid.
EGN_validation[*22]	uint	0...1	1	N	N	Flag that tells if the entered EGN number has to be valid.
Bonuses[*22]	char	Up to 64 symbols	empty	0...31	N	Description of the bonus.
TextReducedVAT[*22]	char	Up to 42 symbols	See remark: [*23]	0...4	N	Free text lines describing reason for reduced VAT.
Config901	uint	0...1	0	N	N	DHL/InTime-whether to request invoice number and date when issuing a simple fiscal receipt. • **'0' –** DHL – ask for invoice number; • **'1' –** InTime – do not ask for invoice number;
LastDocDateBlock24h	char	See remark:	empty	N	Y	The date and time of the most recently issued document without

Parameters description						
		[*27]				Z report being made.
HideSingleItemInfoOnSells	uint	0...1	0	N	N	Hide "quantity * price" information when selling with a single quantity • **'0'** – do not hide; • **'1'** – hide info;
DisableMotorOverheatingAlarm	uint	0...1	0	N	N	Prohibit „Printer Overheating" alarm. • **'0'** – alarm is enabled; • **'1'** – disable the alarm;
ClearOperatorsInZreport	uint	0...1	1	N	N	Clearing the operator registers for each Z report • **'0'** – do not clear registers; • **'1'** – clear all operator registers;
BlockOnAutoZreport	uint	0...1	1[*9] 0[*8]	N	N	• **'0'** - An automatic Z report is generated as needed • **'1'** - Blocks the device until Z report is printed
PrintAutoZreport	uint	0...1	1	N	N	• **'0'** - The automatic Z report is not printed • **'1'** - The automatic Z report is printed
ModemConnMode	uint	0...2	0	N	N	Modem connection mode (Valid for 3G modems only): • **'0'** – auto mode; • **'1'** – 2G only; • **'2'** - 3G only
ModemConnOper	uint	0...1	0	N	N	GSM operator selection mode: • **'0'** – auto mode; • **'1'** – manual according IMSI number;
StornoCashControl	uint	0...1	0	N	N	Checking cash in safe at a storno with operator mistake reason: • '0' - no • '1' - yes
PYxx_Pgm[*8]	uint	1...6[*31]	1...6	0...5	N	Payment shortcut
PYxx_Server	uint	0...10	1...6	0...5	N	Payment shortcut for NRA server fields (see: "AI_PaymentParameters_X.pdf")
PYxx_FPmode	uint	1...6[*31]	1...6	0...5	N	Crossreference to payments in command 53 options each index corresponds to option in cmd 53, The value means payment type [*31]
MobOperName	char	Up to 64 symbols	empty	N	Y	MobileOperatorName
OnlyPCcontrol[*8]	uint	0...3	0	N	N	The device is controlled only from remote PC • 0 - no restrictions • 1- 'R' mode is disabled • 2- 'R' and 'Z' modes are disabled • 3- 'R','Z' and 'P' modes are disabled
EcrSrv_port	uint	1...65535	4000,5000,6000,7000	0...3	N	Ports for EcrSrv comunication protocol
AdditionalSoundsLevel	uint	0...2	0 1[*21]	N	N	Additional sounds: • '0' – no additional sounds • '1' – only on power off; • '2' – on power off and on interface change
AskMobileNetworkChecking	uint	0...1	0	N	N	Check connection to NRA server: • '0' – after Z report – postponed • '1' – before Z report
EcrSellOnlineOnly[*10]	uint	0...1	0	N	N	Work from keyboard only if EcrSrv is used: • '0' – disabled • '1' – enabled
Online_MsgSeqUN[*10]	uint	0...999999999	0	N	N	Sequence number of the message on EcrSrv comminication mode
Online_TrnN	uint	0...999999999	0	N	N	Sequence number of the transaction on EcrSrv comminication mode

5. Remarks

[*1] WP-500X, WP-50X, DP-25X, WP-25X, DP-05C: the default password for each operator is equal to the corresponding number (for example, for Operator 1 the password is "1"). FMP-350X, FMP-55X, FP-700X, FP-700XE: the default password for each operator is “0000”
[*2] If DiscountType is zero or empty, parameter DiscountValue must be empty.
[*3] The option is not used on FMP-350X and FMP-55X
[*4] Max value of Price * Quantity is *9999999.99.
[*5] If DiscountType is zero or empty, this parameter must be empty. Void operations are made by placing '-' before PluCode. In order to make void operation the Price parameter must be the same as the price at which the item was sold.
[*6] "Index" = 0 for current values, "Index" = 1 for saved values after successful registration/change on the NRA server;
[*7] Depending on decimal point position.
[*8] Valid for WP-500X, WP-50X, DP-25X, WP-25X, DP-05C.
[*9] Valid for FP-700X, FP-700XE, FMP-350X, FMP-55X.
[*10]Used in WP-500X, WP-50X, DP-25X, WP-25X, DP-05C only.
[*11]Valid for FP-700X, FP-700XE, FMP-350X.
[*12]Valid for FMP-55X.
[*13]0-“В БРОЙ”, 1-“КРЕДИТ“, 2-“ДЕБ.КАРТА“, 3-“ЧЕК“, 4-“ВАУЧЕР“, 5-“КУПОН“.
[*14]0-”ИМЕ НА ФИРМА”, 1-”АДРЕС НА ФИРМА”, 2-”ИМЕ НА ОБЕКТ”, 3-”АДРЕС НА ОБЕКТ”, 4...9-“”.
[*15]0-“бр.”, 1-“кг“, 2-“м“, 3-“л“, 4-“ч“, 5...19-“”.
[*16]Used in FP-700X and FP-700XE only.
[*17]0-1200, 1-2400, 2-4800, 3-9600, 4-14400, 5-19200, 6-38400, 7-56000, 8-57600, 9-115200.
[*18]Valid for WP-50X, DP-05C.
[*19]Valid for DP-25X, WP-25X.
[*20]Valid for WP-500X.
[*21]Valid for FMP-55X only.
[*22]Valid for DP-05C only.
[*23]0-“ДДС СТАВКА 20%.....[ ]”, 1-“ДДС СТАВКА 0%:”, 2-“Съгл. чл. 22 ал. 2 от ЗДДС.....[ ]”, 3-“ Съгл. чл. 30 от ЗДДС.....[ ]”, 4-“ПОДПИС:.....”.
[*24]0 - “none”, 1 - “Barcode”, 2 - “Scale”, 3 - “External display”.
[*25]0 - “datecs”, 1 - “cas”, 2 - “atlas”.
[*26]Valid only if ComPortDevice=2.
[*27]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;
If it is only time the format is hh:mm:ss.
If it is only date the format is DD-MM-YY.
[*28]0.01-9999999.99 for sum operations, 0.01-99.99 for percentage operations.
[*29]This option is possible to be used only if device is registered with FDType = 11 or 21!
[*30]DateTime – Date and time in format: "DD-MM-YY hh:mm";
DD – Day; MM – Month, YY – Year, hh – Hour, mm – Minute;
[*31]1-“КРЕДИТ“, 2-“ДЕБ.КАРТА“, 3-“ЧЕК“, 4-“ВАУЧЕР“, 5-“КУПОН“, 6-“EUR“,
[*32]see “AI_Example_of_using_commands_202 _and_203.7z”

6. Status bits

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

Status byte 0: General purpose			
Name	Number	Mark	Description
Byte 0 Bit 7	**0.7**		Always 1
Byte 0 Bit 6	**0.6**	#	Cover is open
Byte 0 Bit 5	**0.5**		General error - this is OR of all errors marked with #
Byte 0 Bit 4	**0.4**		Failure in printing mechanism.
Byte 0 Bit 3	**0.3**		Always 0
Byte 0 Bit 2	**0.2**		The real time clock is not synchronized
Byte 0 Bit 1	**0.1**	#	Command code is invalid
Byte 0 Bit 0	**0.0**	#	Syntax error

Status byte 1: General purpose			
Name	Number	Mark	Description
Byte 1 Bit 7	**1.7**		Always 1
Byte 1 Bit 6	**1.6**		Always 0
Byte 1 Bit 5	**1.5**		Always 0
Byte 1 Bit 4	**1.4**		Always 0
Byte 1 Bit 3	**1.3**		Always 0
Byte 1 Bit 2	**1.2**		Always 0
Byte 1 Bit 1	**1.1**	#	Command is not permitted
Byte 1 Bit 0	**1.0**	#	Overflow during command execution

Status byte 2: General purpose			
Name	Number	Mark	Description
Byte 2 Bit 7	**2.7**		Always 1
Byte 2 Bit 6	**2.6**		Always 0
Byte 2 Bit 5	**2.5**		Non-fiscal receipt is open
Byte 2 Bit 4	**2.4**		EJ nearly full
Byte 2 Bit 3	**2.3**		Fiscal receipt is open
Byte 2 Bit 2	**2.2**		EJ is full
Byte 2 Bit 1	**2.1**		Near paper end
Byte 2 Bit 0	**2.0**	#	End of paper

Status byte 3: Not used			
Name	Number	Mark	Description
Byte 3 Bit 7	**3.7**		Always 1
Byte 3 Bit 6	**3.6**		Always 0
Byte 3 Bit 5	**3.5**		Always 0
Byte 3 Bit 4	**3.4**		Always 0
Byte 3 Bit 3	**3.3**		Always 0
Byte 3 Bit 2	**3.2**		Always 0
Byte 3 Bit 1	**3.1**		Always 0
Byte 3 Bit 0	**3.0**		Always 0

Status byte 4: Fiscal memory			
Name	Number	Mark	Description
Byte 4 Bit 7	**4.7**		Always 1
Byte 4 Bit 6	**4.6**		Fiscal memory is not found or damaged
Byte 4 Bit 5	**4.5**		OR of all errors marked with '*' from Bytes 4 и 5
Byte 4 Bit 4	**4.4**	*	Fiscal memory is full
Byte 4 Bit 3	**4.3**		There is space for less then 60 reports in Fiscal memory
Byte 4 Bit 2	**4.2**		Serial number and number of FM are set
Byte 4 Bit 1	**4.1**		Tax number is set
Byte 4 Bit 0	**4.0**	*	Error when trying to access data stored in the FM

Status byte 5: General purpose			
Name	Number	Mark	Description
Byte 5 Bit 7	**5.7**		Always 1
Byte 5 Bit 6	**5.6**		Always 0
Byte 5 Bit 5	**5.5**		Always 0
Byte 5 Bit 4	**5.4**		VAT are set at least once
Byte 5 Bit 3	**5.3**		Device is fiscalized
Byte 5 Bit 2	**5.2**		Always 0
Byte 5 Bit 1	**5.1**		FM is formatted
Byte 5 Bit 0	**5.0**		Always 0

Status byte 6: Not used			
Name	Number	Mark	Description
Byte 6 Bit 7	**6.7**		Always 1
Byte 6 Bit 6	**6.6**		Always 0
Byte 6 Bit 5	**6.5**		Always 0
Byte 6 Bit 4	**6.4**		Always 0
Byte 6 Bit 3	**6.3**		Always 0
Byte 6 Bit 2	**6.2**		Always 0
Byte 6 Bit 1	**6.1**		Always 0
Byte 6 Bit 0	**6.0**		Always 0

Status byte 7: Not used			
Name	Number	Mark	Description
Byte 7 Bit 7	**7.7**		Always 1
Byte 7 Bit 6	**7.6**		Always 0
Byte 7 Bit 5	**7.5**		Always 0
Byte 7 Bit 4	**7.4**		Always 0
Byte 7 Bit 3	**7.3**		Always 0
Byte 7 Bit 2	**7.2**		Always 0
Byte 7 Bit 1	**7.1**		Always 0
Byte 7 Bit 0	**7.0**		Always 0

7. Error codes

Error code	Error name	Description
(100000 - 100100) GENERIC ERRORS - FISCAL DEVICES		
-100001	ERR_IO	General error in fiscal device: In - out error(cannot read or write)
-100002	ERR_CHECKSUM	General error in fiscal device: Wrong checksum
-100003	ERR_END_OF_DATA	General error in fiscal device: No more data
-100004	ERR_NOTFOUND	General error in fiscal device: The element is not found
-100005	ERR_NO_RECORDS	General error in fiscal device: There are no records found
-100006	ERR_ABORTED	General error in fiscal device: The operation is aborted
-100007	ERR_WRONG_MODE	Wrong mode(standart, training...) is selected.
-100008	ERR_NOT_READY	General error in fiscal device: Device is not ready
-100009	ERR_NOTHING_TO_PRINT	General error in fiscal device: Nothing to print
(100100 - 100254) FISCAL MEMORIES		
-100100	ERR_FM_BUSY	Fiscal memory error: Fiscal memory is busy
-100101	ERR_FM_FAILURE	Fiscal memory error: Fiscal memory failure. Could not read or write
-100102	ERR_FM_WRITE_PROTECTED	Fiscal memory error: Forbidden write in fiscal memory
-100103	ERR_FM_WRONG_ADDRESS	Fiscal memory error: Wrong address in fiscal memory
-100104	ERR_FM_WRONG_SIZE	Fiscal memory error: Wrong size in fiscal memory
-100105	ERR_FM_NOT_CONNECTED	Fiscal memory error: Fiscal memory is not connected
-100106	ERR_FM_WRONG_CHECK_SUM	Fiscal memory error: Wrong checksum in fiscal memory(invalid data)
-100107	ERR_FM_BLOCK_IS_EMPTY	Fiscal memory error: Empty block in fiscal memory
-100108	ERR_FM_MAX_NUMBER	Fiscal memory error: Maximum number of block in fiscal memory
-100109	ERR_FM_WRONG_RANGE	Fiscal memory error: Wrong range in fiscal memory
-100110	ERR_FM_EMPTY_RANGE	Fiscal memory error: Empty range in fiscal memory
-100111	ERR_FM_NEW_MODULE	Fiscal memory error: New module in fiscal memory
-100112	ERR_FM_NOT_EMPTY	Fiscal memory error: Fiscal memory is not empty
-100113	ERR_FM_NOT_EQUAL	Fiscal memory error: Fiscal memory is not equal
-100114	ERR_FM_FULL	Fiscal memory error: Fiscal memory is full
-100115	ERR_FM_NEED_UPDATE	Fiscal memory error: Fiscal memory needs update
-100116	ERR_FM_BLOCKED	Fiscal memory error: Fiscal memory is blocked
(100400 - 100499) PRINTER DRIVER ERRORS		
-100400	ERR_LTP_VCCERR	Line thermal printer mechanism error: Power supply error (3,3 V)
-100401	ERR_LTP_SVPERR	Line thermal printer mechanism error: Power supply error (24V or 8V)
-100402	ERR_LTP_STHERR	Line thermal printer mechanism error: Head overheating
-100403	ERR_LTP_PESENS	Line thermal printer mechanism error: Paper end
-100404	ERR_LTP_HDSENS	Line thermal printer mechanism error: Cover is open
-100405	ERR_LTP_NESENS	Line thermal printer mechanism error: Near paper end
-100406	ERR_LTP_MKSENS	Line thermal printer mechanism error: Mark sensor - not used
-100407	ERR_LTP_CUTERR	Line thermal printer mechanism error: Cutter error
-100408	ERR_LTP_PR_ERR	Line thermal printer mechanism error: Not used
-100409	ERR_LTP_PR_BUSY	Line thermal printer mechanism error: Not used

-100410	ERR_LTP_BZLPDEC	Line thermal printer mechanism error: Not used
-100411	ERR_LTP_BZLCLMP	Line thermal printer mechanism error: Not used
-100412	ERR_LTP_CHARGE_MODE	Line thermal printer mechanism error: Not used
-100413	ERR_LTP_INZERR_MODE	Line thermal printer mechanism error: Not used
-100414	ERR_LTP_MOTOR_OVERRUN	Printer on time is overrun.
(100500 - 100999) SYSTEM ERRORS		
-100500	ERR_PROGRAM_SELF_CHECK_ERROR	System error: Memory structure error
-100501	ERR_SRAM_ERROR	System error: Error in RAM
-100502	ERR_FLASH_ERROR	System error: Flash memory error
-100503	ERR_SDCARD_ERROR	System error: SD card error
-100504	ERR_INVALID_MSG_FILE	System error: Invalid message file
-100505	ERR_FM_ERROR	System error: Fiscal memory error(could not write or read)
-100506	ERR_NO_RAM_BATTERY	System error: No RAM battery
-100507	ERR_SAM_ERROR	System error: SAM module error
-100508	ERR_RTC_ERROR	System error: Real time clock error
-100509	ERR_PROGRAM_EXRAM_CHECK_ERROR	System error: Memory error
-100510	ERR_SDCARD_WRONG_SIZE	System error: The size of SD card is wrong.
-100511	ERR_TPM_ERROR	System error: TPM module error
(101000 - 101499) COMMON LOGICAL ERRORS		
-101000	ERR_NO_HEAP_MEMORY	Common logical error: No heap memory(cannot allocate memory for operation)
-101001	ERR_FILE_MANIPULATE	Common logical error: File manipulate error
-101003	ERR_REJECTED	Common logical error: Operation is rejected
-101004	ERR_BAD_INPUT	Common logical error: Bad input. Some of the data or parameters are incorrect
-101005	ERR_IAP	Common logical error: In Application Programming error
-101006	ERR_NOT_POSSIBLE	Common logical error: The execution of the operation is not possible
-101007	ERR_TMOUT	Common logical error: Timeout. The time for waiting execution is out
-101007	ERR_TIMEOUT	Common logical error: Timeout. The time for waiting execution is out
-101008	ERR_INVALID_TIME	Common logical error: Invalid time
-101009	ERR_CANCELLED	Common logical error: The operation is cancelled
-101010	ERR_INVALID_FORMAT	Common logical error: Invalid format
-101011	ERR_INVALID_DATA	Common logical error: Invalid data
-101012	ERR_PARSE_ERROR	Common logical error: Data parsing error
-101013	ERR_HARDWARE_CONFIGURATION	Common logical error: Hardware configuration error
-101014	ERR_ACCESS_DENIED	ERR_ACCESS_DENIED
-101015	ERR_BAD_DATA_LENGTH	Wrong data length
-101016	ERR_VERIFY_Z	Error during verification of Z reports
-101017	ERR_NO_PERMISSION	Common logical error: No permission
(102000 - 102999) GENERAL ERRORS		
-102000	ERR_LOW_BATTERY	Battery error: Low battery
-102001	ERR_LOW_BATTERY_WARNING	Battery error: Low battery warning
-102002	ERR_OPER_WRONG_PASSWORD	Operator error: Wrong operator password
-102003	ERR_IDNUMBER_IS_EMPTY	ECR error: ID number is empty

-102004	ERR_NOT_FOUND_BLUETOOTH	Bluetooth error: Bluetooth is not found
-102005	ERR_DISPLAY_DISCONNECTED	Display error: Display is not connected
-102006	ERR_PRINTER_DISCONNECTED	Printer error: Printer is not connected
-102007	ERR_SD_NOT_PRESENT	SD card error: SD card not present
-102008	ERR_SD2_NOT_PRESENT	SD card error: SD2 card not present
-102009	ERR_VAT_RATES_ARE_EMPTY	ECR error: VAT rates is not set.
-102010	ERR_HEADER_IS_EMPTY	ECR error: Header lines are empty.
-102011	ERR_ZDDS_NUM_IS_EMPTY	User is registered by VAT, but number of the user is not entered.
-102012	ERR_FMNUMBER_IS_EMPTY	ECR error: FM number is empty
-102013	ERR_SERVICEMAN_NAME_IS_EMPTY	ECR error: Serviceman name is empty
-102014	ERR_SERVICEMAN_ID_IS_EMPTY	ECR error: Serviceman ID is empty
-102015	ERR_TAXOFFICE_ID_IS_EMPTY	ECR error: Tax office ID is empty!
-102016	ERR_WRONG_FORMAT	ECR error: Wrong format
-102017	ERR_TAXNUMBER_IS_EMPTY	ECR error: TAX number is empty
-102018	ERR_WRONG_IDNUMBER	ECR error: ID number is wrong
-102019	ERR_DATETIME_EARLIER_THAN_PREV_Z	ECR error: Date and time are earlier than date and time of previous Z report.
-102020	ERR_NEED_SOFTWARE_PASSWORD	ECR error: The software password is not entered
(103000 - 103999) PLU DATABASE		
-103000	ERR_PLUDB_NOT_FOUND	PLU database error: PLU database is not found
-103001	ERR_PLUDB_PLUCODE_EXISTS	PLU database error: PLU code already exists
-103002	ERR_PLUDB_BARCODE_EXISTS	PLU database error: Barcode already exists
-103003	ERR_PLUDB_FULL	PLU database error: PLU database is full
-103004	ERR_P_HAVE_TURNOVER	PLU database error: PLU has turnover
-103005	ERR_PLUDB_NAME_EXISTS	PLU database error: In the PLU base has an article with same name.
-103006	ERR_PLUDB_NAMES_NOT_UNIQUE	PLU database error: PLU name is not unique.
-103007	ERR_PLUDB_FORMAT_INCOMPATIBLE	PLU database error: Database format is not compatible.
-103008	ERR_PLUDB_CAN_NOT_OPEN	Can't open the PLU database file
(104000 - 104999) SERVICE OPERATIONS		
-104000	ERR_NEED_Z_REPORT	Service operation error: Z report is needed for this operation
-104001	ERR_NEED_SERVICE_JUMPER	Service operation error: Service jumper is needed for this operation
-104002	ERR_NEED_SERVICE_PASSWORD	Service operation error: Service password is needed for this operation
-104003	ERR_FORBIDEN	Service operation error: The operation is forbidden
-104004	ERR_NEED_SERVICE_INTERVENTION	Service operation error: Service intervention is needed
-104005	ERR_NEED_ALL_CLEARING_REPORTS	Service operation error: All clearing report is needed.
-104006	ERR_Z_REPORT_CLOSED	Service operation error: Z report closed.
-104007	ERR_NEED_MONTH_REPORT	Service operation error: Montly report needed.
-104008	ERR_NEED_YEAR_REPORT	Service operation error: Year report needed.
-104009	ERR_NEED_BACKUP	Service operation error: Backup needed.
-104010	ERR_NEED_ALL_PAIDOUT	ERR_NEED_ALL_PAIDOUT
-104011	ERR_NEED_OPERATOR_Z_REPORT	Clearing report for operator is needed.
-104012	ERR_NEED_GROUP_Z_REPORT	Clearing report for item group is needed.
-104013	ERR_NEED_VAT_CHANGES	VAT changes is needed.

(105000 - 105999) EJ - ERRORS		
-105000	ERR_EJ_NO_RECORDS	EJ error: No records in EJ
-105001	ERR_CANNOT_ADD_TO_EJ	EJ error: Cannot add to EJ
-105002	ERR_EJ_WRONG_MAC_RECORD	EJ error: SAM module signature error
-105003	ERR_EJ_IMMPOSSIBLE_TO_CHK_MAC_RECORD	EJ error: Signature key version is changed -> impossible check
-105004	ERR_EJ_BAD_RECORDS	EJ error: Bad record in EJ
-105005	ERR_EJ_CAN_NOT_GENERATE_MAC	EJ error: Generate signature error(cannoct generate signature)
-105006	ERR_EJ_WRONG_TYPE_TO_SIGN	EJ error: Wrong type of document to sign
-105007	ERR_EJ_ALREADY_SIGNED	EJ error: Document is already signed
-105008	ERR_EJ_NOT_FROM_THIS_DEVICE	EJ error: EJ is not from this device
-105009	ERR_EJ_NEAR_FULL	EJ error: EJ is almost full
-105010	ERR_EJ_FULL	EJ error: EJ is full
-105011	ERR_EJ_WRONG_FORMAT	EJ error: Wrong format of EJ
-105012	ERR_EJ_NOT_READY	The electronic journal is not ready.
-105013	ERR_EJ_NEED_NEW	Error in EJ structure. Create new one.
-105014	ERR_EJ_CANNOT_CREATE	Cannot create file
-105015	ERR_EJ_CANNOT_VERIFY	Cannot verify data in EJ
-105016	ERR_EJ_CANNOT_OPEN	Cannot open a file
-105017	ERR_EJ_CANNOT_CLOSE	Cannot close file
-105018	ERR_EJ_CANNOT_READ	Cannot read data from a file
-105019	ERR_EJ_CANNOT_WRITE	Cannot write data in file
(106000 - 106999) CLIENTS DATABASE ERRORS		
-106000	ERR_R_FIRM_NOTEXIST	Client database error: Firm does not exist
-106001	ERR_FIRMDB_FIRMCODE_EXISTS	Client database error: Firmcode already exists
-106002	ERR_FIRMDB_EIK_EXISTS	Client database error: EIK already exists
-106003	ERR_FIRMDB_FULL	Client database error: Firm database is full
-106004	ERR_FIRMDB_NOT_FOUND	Client database error: Firm database is not found
(110100 - 110199) EXT FISCAL DEVICE ERRORS		
-110100	ERR_DEVICE_COMM_ERROR	Device error: Communication error
-110101	ERR_DEVICE_WRONG_STRUCT	Device error: Wrong struct format
-110102	ERR_DEVICE_STFLAG_ACTIVE	Device error: ST flag is active
-110103	ERR_DEVICE_INVALID_DATA	Device error: Invalid data
-110104	ERR_DEVICE_NOT_FISCALIZED	Device error: Device is not fiscalized
-110105	ERR_DEVICE_ALREADY_FISCALIZED	Device error: Device is already fiscalized
-110106	ERR_DEVICE_IN_SERVICE_MODE	Device error: Device is in service mode
-110107	ERR_DEVICE_PASSED_SERVICE_DATE	Device error: Service date is passed
-110108	ERR_DEVICE_DAY_IS_OPEN	Device error: Day(shift) is open
-110109	ERR_DEVICE_DAY_IS_CLOSED	Device error: Day(shift) is closed
-110110	ERR_DEVICE_WRONG_NUMBERS	Device error: Z-report number and shift number are not equal
-110111	ERR_DEVICE_ADMIN_ONLY	Device error: Only admin has permition
-110112	ERR_DEVICE_UNFISCALIZED	Device error: Fiscal memory is closed
(110200 - 110299) NAP SERVER		

ERRORS		
-110200	ERR_NAP_OPEN_SESSION	NAP server error: Error open session
-110201	ERR_NAP_PREPARE_DATA	NAP server error: Error preparing data for server
-110202	ERR_NAP_SEND_DATA	NAP server error: There is unsent data
-110203	ERR_NAP_RECV_DATA	NAP server error: Receiving data error
-110204	ERR_NAP_EMPTY_DATA	NAP server error: Empty data
-110205	ERR_NAP_NEGATIVE_ANSWER	NAP server error: Server negative answer
-110206	ERR_NAP_WRONG_ANSWER_FORMAT	NAP server error: Wrong answer format
-110207	ERR_NAP_HOSTDI_ZERRO	NAP server error: Server HOSTDI is zerro
-110208	ERR_NAP_EXCEPTION	NAP server error: Server exception
-110209	ERR_NAP_NOTPERSONALIZED	NAP server error: Not registered on server
-110209	ERR_NAP_NOTREGISTERED	NAP server error: Not registered on server
-110210	ERR_NAP_BLOCKED_72H	NAP server error: Communication with NAP server is blocked
-110211	ERR_NAP_BLOCKED_NO_MODEM_LAN	NAP server error: Modem error
-110212	ERR_NAP_BUSY	NAP server error: NAP is busy
-110213	ERR_NAP_REGISTERED	NAP server error: Already registered
-110214	ERR_NAP_WRONG_PSTYPE	NAP server error: Wrong PS type
-110215	ERR_NAP_DEREG_ON_SERVER	NAP server error: Deregistered in NAP
-110216	ERR_NAP_WRONG_IMSI	NAP server error: Wrong IMSI number
-110217	ERR_NAP_BLOCKED_MAX_ZERRORS	NAP server error: Device is blocked(maximum Z-reports)
-110218	ERR_NAP_WRONG_FDTYPE	NAP server error: Wrong FD(Fiscal device) type
-110219	ERR_NAP_BLOCKED_BY_SERVER	NAP server error: The ECR is blocked by server
-110220	ERR_NAP_BLOCKED_ERROR_FROM_SERVER	NAP server error: The ECR is blocked - server error
-110221	ERR_NAP_NO_SERVER_ADDRESS	NAP server error: No server address
-110222	ERR_NAP_NO_REGISTRATIONS_POSSIBLE	NAP server error: Max. registrations reached.
-110223	ERR_NAP_INVALID_OPERATOR_INN	Invalid INN of the cashier
-110224	ERR_NAP_INVALID_SERVER_INN	Invalid INN of the server
-110225	ERR_NAP_BLOCKED_MAX_SELLERRORS	NAP server error: Device is blocked(unsent sales documents)
-110226	ERR_NAP_BLOCKED_24H	NAP server error: Communication with NAP server is blocked. More than 24 hours from last sent receipt.
(110300 - 110399) WORK_INVALID		
-110300	ERR_WORK_INVALID_FILE	Working error: Invalid file
-110301	ERR_WORK_INVALID_PARAM	Working error: Invalid parameters
(110500 - 110599) MODEM ERRORS		
-110500	ERR_MODEM_CTRL	Modem error: error in communication between device and modem
-110501	ERR_MODEM_NO_SIM	Modem error: No SIM card
-110502	ERR_MODEM_PIN	Modem error: Wrong PIN of SIM
-110503	ERR_MODEM_ATTACH	Modem error: Cannot register to mobile network
-110504	ERR_MODEM_PPP	Modem error: No PPP connection(cannot connect)
-110505	ERR_MODEM_CONFIG	Modem error: Wrong modem configuration(for example - no programmed apn)
-110506	ERR_MODEM_WAIT_INIT	Modem error: Modem initializing
-110507	ERR_MODEM_NOTREADY	Modem error: Modem is not ready
-110508	ERR_MODEM_REMOVE_SIM	Modem error: Remove SIM card
-110509	ERR_MODEM_CELL_FOUND	Modem error: Modem found a cell

-110510	ERR_MODEM_CELL_NOTFOUND	Modem error: Modem does not find a cell
-110511	ERR_MODEM_LOT_DAYS_FAIL	Modem error: Failed lot days
-110512	ERR_MODEM_PPP_DIFFERENT_APN	Modem error: Already connected to a different APN when a PPP connection is started
(110600 - 110699) WIFI ERRORS		
-110601	ERR_MODEM_CONNECT_AP	Modem error: Device is not connected to AP(access point)
(110700 - 110799) NETWORK ERRORS		
-110700	ERR_NET_DNS_RESOLVE	Network error: Cannot resolve address
-110701	ERR_NET_SOCKET	Network error: Cannot open socket for communication with server
-110702	ERR_NET_CONNECTION	Network error: Connection error(cannot connect to a server)
-110703	ERR_NET_CONFIG	Network error: Config error(for example: no server address)
-110704	ERR_NET_SOCKET_CONNECTED	Network error: Connection socket is already opened
-110705	ERR_NET_SSL_ERROR	Network error: SSL communication error(something went wrong in cryptographic protocol)
-110706	ERR_NET_HTTP_ERROR	Network error: HTTP communication error(something went wrong in http protocol)
(110800 - 110899) TAX_TERMINAL_ERRORS		
-110800	ERR_DT_OK	Tax terminal error: No error
-110801	ERR_DT_UNKNOWN_ID	Tax terminal error: Unknown ID
-110802	ERR_DT_INVALID_TOKEN	Tax terminal error: Invalid token(key from the server)
-110803	ERR_DT_PROTOCOL_ERROR	Tax terminal error: Protocol error
-110804	ERR_DT_UNKNOWN_COMMAND	Tax terminal error: The command is unknown
-110805	ERR_DT_UNSUPPORTED_COMMAND	Tax terminal error: The command is not supported
-110806	ERR_DT_INVALID_CONFIGURATION	Tax terminal error: Invalid configuration
-110807	ERR_DT_SSL_IS_NOT_ALLOWED	Tax terminal error: SSL is not allowed
-110808	ERR_DT_INVALID_REQUEST_NUMBER	Tax terminal error: Invalid request number
-110809	ERR_DT_INVALID_RETRY_REQUEST	Tax terminal error: Invalid retry request
-110810	ERR_DT_CANT_CANCEL_TICKET	Tax terminal error: Cannot cancel ticket
-110811	ERR_DT_OPEN_SHIFT_TIMEOUT_EXPIRED	Tax terminal error: More than 24 hours from shift opening
-110812	ERR_DT_INVALID_LOGIN_PASSWORD	Tax terminal error: Invalid login name or password
-110813	ERR_DT_INCORRECT_REQUEST_DATA	Tax terminal error: Incorrect request data
-110814	ERR_DT_NOT_ENOUGH_CASH	Tax terminal error: Not enough cash
-110815	ERR_DT_BLOCKED	Tax terminal error: Blocked from server
-110854	ERR_DT_SERVICE_TEMPORARILY_UNAVAILABLE	Tax terminal error: Service temporarily unavailable
-110855	ERR_DT_UNKNOWN_ERROR	Tax terminal error: Unknown error
(111000 - 111499) REGMODE ERRORS		
-111000	ERR_R_CLEAR	Registration mode error: Common error, followed by deliting all data for the command
-111001	ERR_R_NOCLEAR	Registration mode error: Common error, followed by partly deliting data for the command
-111002	ERR_R_SYNTAX	Registration mode error: Syntax error. Check the parameters of the command
-111003	ERR_R_NPOSSIBLE	Registration mode error: Cannot do operation
-111004	ERR_R_PLU_NOTEXIST	Registration mode error: PLU code was not found

-111005	ERR_R_PLU_VAT_DISABLE	Registration mode error: Forbidden VAT
-111006	ERR_R_PLU_QTY_PRC	Registration mode error: Overflow in multiplication of quantity and price
-111007	ERR_R_PLU_NO_PRC	Registration mode error: PLU has no price
-111008	ERR_R_PLU_GRP_RANGE	Registration mode error: Group is not in range
-111009	ERR_R_PLU_DEP_RANGE	Registration mode error: Department is not in range
-111010	ERR_R_BAR_NOTEXIST	Registration mode error: BAR code does not exist
-111011	ERR_R_OVF_TOTAL	Registration mode error: Overflow of the PLU turnover
-111012	ERR_R_OVF_QTY	Registration mode error: Overflow of the PLU quantity
-111013	ERR_R_ECR_OVR	Registration mode error: ECR daily registers overflow
-111014	ERR_R_BILL_TL_OVR	Registration mode error: Bill total register overflow
-111015	ERR_R_OPEN_BON	Registration mode error: Receipt is opened
-111016	ERR_R_CLOSED_BON	Registration mode error: Receipt is closed
-111017	ERR_R_PAY_NOCASH	Registration mode error: No cash in ECR
-111018	ERR_R_PAY_STARTED	Registration mode error: Payment is initiated
-111019	ERR_R_OVF_TRZ_BUFF	Registration mode error: Maximum number of sales in receipt
-111020	ERR_R_NO_TRANSACTIONS	Registration mode error: No transactions
-111021	ERR_R_NEGATIVE_SUMVAT	Registration mode error: Possible negative turnover
-111022	ERR_R_PYFOREIGN_HAVERESTO	Registration mode error: Foreign payment has change
-111023	ERR_R_TRZ_NOT_EXIST	Registration mode error: Transaction is not found in the receipt
-111024	ERR_R_END_OF_24_HOUR_PERIOD	Registration mode error: End of 24 hour blocking
-111025	ERR_R_NO_VALID_INVOICE	Registration mode error: Invalid invoice range
-111026	ERR_R_POS_TERM_CANCELED	Registration mode error: Operation is cancelled by operator
-111027	ERR_R_POS_TERM_APPROVED	Registration mode error: Operation approved by POS
-111028	ERR_R_POS_TERM_NOT_APPROVED	Registration mode error: Operation is not approved by POS
-111029	ERR_R_POS_TERM_CONN_ERR	Registration mode error: POS terminal communication error
-111030	ERR_R_PLU_QTY_PRC_TOO_LOW	Registration mode error: Multiplication of quantity and price is 0
-111031	ERR_R_VALUE_TOO_BIG	Registration mode error: Value is too big
-111032	ERR_R_VALUE_BAD	Registration mode error: Value is bad
-111033	ERR_R_PRICE_TOO_BIG	Registration mode error: Price is too big
-111034	ERR_R_PRICE_BAD	Registration mode error: Price is bad
-111035	ERR_R_ALL_VOID_SELECTED	Registration mode error: Operation all void is selected to be executed
-111036	ERR_R_ONLY_ALL_VOID_IS_POSSIBLE	Registration mode error: Only all void operation is permitted
-111040	ERR_R_REST_NOFREESPC_SELLS	Registration mode error: Restaurant: There is no free space for other purchases
-111041	ERR_R_REST_NOFREESPCFORNEWACNT	Registration mode error: Restaurant: There is no free space for new acount
-111042	ERR_R_REST_ACCOUNT_IS_OPENED	Registration mode error: Restaurant: Account is already opened
-111043	ERR_R_REST_WRONG_INDEX	Registration mode error: Restaurant: Wrong index
-111044	ERR_R_REST_ACNT_IS_NOTFOUND	Registration mode error: Restaurant: Account is not found
-111045	ERR_R_REST_NOT_PERMITTED	Registration mode error: Restaurant: Not permitted(only for admins)
-111046	ERR_R_OPEN_NONFISCALBON	Registration mode error: non-fiscal receipt is opened
-111047	ERR_R_OPEN_FISCALBON	Registration mode error: fiscal receipt is opened
-111048	ERR_R_BUYERS_TIN_IS_ENTERED	Registration mode error: Buyers TIN is already entered
-111049	ERR_R_BUYERS_TIN_IS_NOT_ENTERED	Registration mode error: Buyers TIN is not entered
-111050	ERR_R_PAY_NOT_STARTED	Registration mode error: Payment is not initiated
-111051	ERR_R_BON_TYPE_MISMATCH	Registration mode error: Reeipt type mismatch
-111052	ERR_R_REACH_BON_TL_LIMIT	Registration mode error: Receipt total limit is reached

-111053	ERR_R_CASH_NO_MULT_MIN_COIN	Registration mode error: Sum cannot be divided by the minimum coin
-111054	ERR_R_PAY_BIG_AMOUNT	Registration mode error: Sum must be <= payment amount
-111055	ERR_R_PAY_VOUCHER_NEED_INPUT_SUM	Registration mode error: Sum of voucher must be entered when paying with voucher
-111056	ERR_R_PAY_VOUCHER_NEED_SURCHARGE	Registration mode error: Value surcharge of the difference between voucher sum and total must be done when paying with voucher and sum > total
-111057	ERR_R_PAY_FOREIGN_DISABLED	Registration mode error: Payment with foreign currency is disabled
-111058	ERR_R_PAY_FOREIGN_IMPOSSIBLE	Registration mode error: Payment with foreign currency is impossible
-111059	ERR_R_PAY_FOREIGN_SMALL_AMOUNT	Registration mode error: Sum must be bigger or equal to payment amount
-111060	ERR_R_SAFE_OPEN_DISABLED	Registration mode error: Safe opening is disabled
-111061	ERR_R_PAY_FORBIDDEN	Registration mode error: Forbidden payment
-111062	ERR_R_PERC_KEY_FORBIDDEN	Registration mode error: Forbidden key for surcharge/discount
-111063	ERR_R_AMOUNT_BIGGER_BILLAMOUNT	Registration mode error: Entered sum is bigger than receipt sum
-111064	ERR_R_AMOUNT_SMALLER_BILLAMOUNT	Registration mode error: Entered sum is smaller than receipt sum
-111065	ERR_R_ZERO_BILLAMOUNT	Registration mode error: Fiscal printer: Sum of receipt is 0. Operation 'void' is needed
-111066	ERR_R_ALL_VOID_EXECUTED	Registration mode error: Fiscal printer: Operation 'void' is executed. Close receipt is needed
-111067	ERR_R_OPEN_STORNOBON	Registration mode error: Storno receipt is opened
-111068	ERR_R_PAY_ZERO_AMOUNT	Registration mode error: Sum is not entered
-111069	ERR_R_PLU_PRICETYPE_RANGE	Registration mode error: Price type is invalid
-111070	ERR_R_PLU_PRICETYPE_LINKED	Registration mode error: Linked surcharge is forbidden
-111071	ERR_R_PLU_PRICETYPE_NEGATIVE	Registration mode error: Negative price is forbidden
-111072	ERR_R_MORE_THAN_ONE_VAT	Registration mode error: More than 1 VAT in one receipt is not allowed
-111073	ERR_R_PINPAD	Registration mode error: Pinpad error
-111074	ERR_R_WRONG_BUYERS_DATA	Registration mode error: Buyer data is wrong
-111075	ERR_R_VAT_SYSTEM_DISABLE	Registration mode error: Vat system disable.
-111076	ERR_R_OPER_NOT_LOGGED_IN	Operator not logged in.
-111077	ERR_R_WRONG_DATE_FM	The receipt date is early on last date in fiscal memory.
-111078	ERR_R_CORR_DATA_NOT_ENTERED	Correction receipt data is not entered!
-111079	ERR_R_FRACTIONAL_QTY	Fractional quantity!
-111080	ERR_R_OUT_OF_STOCK	Registration mode error: Registration mode error: Out of stock
-111081	ERR_R_STL_NEEDED	Registration mode error: Must pushing of the STL before TL.
-111082	ERR_R_PACK_NOTEXIST	Package does not exist
-111083	ERR_R_PLU_UNIT_NOTEXIST	Measuring unit not found
-111084	ERR_R_PLU_CATEGORY_NOTEXIST	Category not found in the data base
-111085	ERR_R_DEP_WRONG_NAME	Invalid department name
-111086	ERR_R_BANK_TERM_NOT_CONFIGURED	Bank terminal not configured
-111087	ERR_R_SIGN_PAY_INCORECT	Disallowed 'признак расчета' (Russia)
-111088	ERR_R_SIGN_INCORRECT	Forbidden признак товара
-111089	ERR_R_PLU_OVER_MAX_PRC	Entered price is bigger than the programmed
-111090	ERR_R_PLU_FIX_PRC	Fix PLU's price
-111091	ERR_R_SIGN_AGENT_INCORECT	Incorect sign agent.
-111092	ERR_R_PAY_VOUCHER_RESTO	Voucher payment cannot have change
-111093	ERR_R_PAY_ADVANCE_BIG	Sum for advance payment is bigger than the sum of article
-111094	ERR_R_PAY_STORNO_RESTO	Payment in storno can not have change
-111095	ERR_R_NOT_EXCISE_PLU_WITH_EXCISE_STA	Invalid parameter - PLU is not defined as excise PLU

	MP	
-111096	ERR_R_EXCISE_PLU_WITHOUT_EXCISE_STAMP	Excise stamp of an excise PLU is not entered
-111097	ERR_R_EXCISE_PLU_FORBIDDEN	SALE FORBIDDEN (excise stamp is not valid)
-111098	ERR_R_WAIT_UNTIL_MIDNIGHT	Can't open new day. PLease wait until midnight.
(111500 - 111799) PINPAD ERRORS		
-111500	ERR_PINPAD_NONE	Pinpad error: No error from pinpad
-111501	ERR_PINPAD_GENERAL	Pinpad error: General unicreditbulbank icon error
-111502	ERR_PINPAD_INVALID_COMMAND	Pinpad error: Not valid command or sub command code
-111503	ERR_PINPAD_INVALID_PARAM	Pinpad error: Invalid parameter
-111504	ERR_PINPAD_INVALID_ADDRESS	Pinpad error: The address is outside limits
-111505	ERR_PINPAD_INVALID_VALUE	Pinpad error: The value is outside limits
-111506	ERR_PINPAD_INVALID_LENGTH	Pinpad error: The length is outside limits
-111507	ERR_PINPAD_NOT_PERMIT	Pinpad error: The action is not permited in current state
-111508	ERR_PINPAD_NO_DATA	Pinpad error: There is no data to be returned
-111509	ERR_PINPAD_TIMEOUT	Pinpad error: Timeout occurs
-111510	ERR_PINPAD_INVALID_KEY_NUMBER	Pinpad error: Invalid key number
-111511	ERR_PINPAD_INVALID_KEY_ATTRIBUTES	Pinpad error: Invalid key attributes(usage)
-111512	ERR_PINPAD_INVALID_DEVICE	Pinpad error: Calling of non-existing device
-111513	ERR_PINPAD_NOT_SUPPORT	Pinpad error: (Not used in this FW version)
-111514	ERR_PINPAD_PIN_LIMIT	Pinpad error: Pin entering limit exceed
-111515	ERR_PINPAD_FLASH	Pinpad error: General error in flash commands
-111516	ERR_PINPAD_HARDWARE	Pinpad error: General hardware unicreditbulbank error
-111517	ERR_PINPAD_INVALID_CRC	Pinpad error: Invalid code check (Not used in this FW version)
-111518	ERR_PINPAD_CANCEL	Pinpad error: The button 'CANCEL' is pressed
-111519	ERR_PINPAD_INVALID_SIGNATURE	Pinpad error: Invalid signature
-111520	ERR_PINPAD_INVALID_HEADER	Pinpad error: Invalid data in header
-111521	ERR_PINPAD_INVALID_PASSWORD	Pinpad error: Incorrect password
-111522	ERR_PINPAD_INVALID_KEY_FORMAT	Pinpad error: Invalid key format
-111523	ERR_PINPAD_SCR	Pinpad error: General unicreditbulbank error in smart card reader
-111524	ERR_PINPAD_HAL	Pinpad error: Error code returned from HAL functions
-111525	ERR_PINPAD_INVALID_KEY	Pinpad error: Invalid key (may not be present)
-111526	ERR_PINPAD_NO_PIN_DATA	Pinpad error: The PIN length is less than 4 or bigger than 12
-111527	ERR_PINPAD_INVALID_REMINDER	Pinpad error: Issuer or ICC key invalid remainder length
-111528	ERR_PINPAD_NOT_INIT	Pinpad error: Not initialized (Not used in this FW version)
-111529	ERR_PINPAD_LIMIT	Pinpad error: Limit is reached (Not used in this FW version)
-111530	ERR_PINPAD_INVALID_SEQUENCE	Pinpad error: Invalid sequence (Not used in this FW version)
-111531	ERR_PINPAD_NO_PERMITION	Pinpad error: The action is not permitted
-111532	ERR_PINPAD_NO_TMK	Pinpad error: TMK is not loaded. The action cannot be executed
-111533	ERR_PINPAD_INVALID_KEK	Pinpad error: Wrong key format
-111534	ERR_PINPAD_DUPLICATE_KEY	Pinpad error: Duplicated key
-111535	ERR_PINPAD_KEYBOARD	Pinpad error: General keyboard error
-111536	ERR_PINPAD_KEYBOARD_NOT_CALIBRATED	Pinpad error: The keyboard is no calibrated.
-111537	ERR_PINPAD_KEYBOARD_FAILED	Pinpad error: Keyboard bug detected.
-111538	ERR_PINPAD_DEVICE_BUSY	Pinpad error: The device is busy, try again
-111539	ERR_PINPAD_TAMPERED	Pinpad error: Device is tampered

-111540	ERR_PINPAD_EMSR	Pinpad error: Error in encrypted head
-111541	ERR_PINPAD_ACCEPT	Pinpad error: The button 'OK' is pressed
-111542	ERR_PINPAD_INVALID_PAN	Pinpad error: Wrong PAN
-111543	ERR_PINPAD_NOT_ENOUGH_MEMORY	Pinpad error: Out of memory
-111544	ERR_PINPAD_EMV	Pinpad error: EMV error
-111545	ERR_PINPAD_CRYPTOGRAPHY	Pinpad error: Cryptographic error
-111546	ERR_PINPAD_COMMUNICATION	Pinpad error: Communication error
-111547	ERR_PINPAD_INVALID_VERSION	Pinpad error: Invalid firmware version
-111548	ERR_PINPAD_NOPAPER	Pinpad error: Printer is out of paper
-111549	ERR_PINPAD_OVERHEATED	Pinpad error: Printer is overheated
-111550	ERR_PINPAD_NOT_CONNECTED	Pinpad error: Device is not connected
-111551	ERR_PINPAD_USE_CHIP	Pinpad error: Use the chip reader
-111552	ERR_PINPAD_END_DAY	Pinpad error: End the day first
-111554	ERR_PINPAD_BOR_ERR	Pinpad error: Error from Borica
-111555	ERR_PINPAD_NO_CONN	Pinpad error: No connection with pinpad
-111556	ERR_PINPAD_ECR	Pinpad error: Success in pinpad, unsuccess in ECR
-111557	ERR_PINPAD_NOT_CONF	Pinpad error: Not configured connection between fiscal device and PinPad
-111558	ERR_PINPAD_SAME_TRANS	Pinpad error: The last transactions are equals or connection is interrupted - try again.
-111559	ERR_PINPAD_RECEIPT	Pinpad error: Payment type: debit/credit card via PinPad. In the fiscal receipt is allowed only one payment with such type.
-111560	ERR_PINPAD_FP_TRANS	Pinpad error: Unknown result of the transaction between fiscal device and PinPad
-111561	ERR_PINPAD_NOT_CONF_TYPE	Pinpad error: Pinpad type not configured
-111700	ERR_PINPAD_INV_AMOUNT	Pinpad error: Invalid ammount.
-111701	ERR_PINPAD_TRN_NOT_FOUND	Pinpad error: Transaction not found.
-111702	ERR_PINPAD_FILE_EMPTY	Pinpad error: The file is empty.
-111703	ERR_PINPAD_MAX_CASHBACK	Entered cashback is bigger than cashback limit.
(111800 - 111899) SCALE REMOTE CONTROL		
-111800	ERR_SCALE_NOT_RESPOND	ERR_SCALE_NOT_RESPOND
-111801	ERR_SCALE_NOT_CALCULATED	ERR_SCALE_NOT_CALCULATED
-111802	ERR_SCALE_WRONG_RESPONSE	ERR_SCALE_WRONG_RESPONSE
-111803	ERR_SCALE_ZERO_WEIGHT	ERR_SCALE_ZERO_WEIGHT
-111804	ERR_SCALE_NEGATIVE_WEIGHT	ERR_SCALE_NEGATIVE_WEIGHT
-111805	ERR_SCALE_T_WRONG_INTF	ERR_SCALE_T_WRONG_INTF
-111806	ERR_SCALE_T_CONNECT	ERR_SCALE_T_CONNECT
-111807	ERR_SCALE_SEND	ERR_SCALE_SEND
-111808	ERR_SCALE_RECEIVE	ERR_SCALE_RECEIVE
-111809	ERR_SCALE_FILE_GENERATE	ERR_SCALE_FILE_GENERATE
-111810	ERR_SCALE_NOT_CONFIG	ERR_SCALE_NOT_CONFIG
(112000 - 112099) FP_MODE ERRORS		
-112000	ERR_FP_INVALID_COMMAND	Fiscal printer error: Fiscal printer invalid command
-112001	ERR_FP_INVALID_SYNTAX	Fiscal printer error: Fiscal printer command invalid syntax
-112002	ERR_FP_COMMAND_NOT_PERMITTED	Fiscal printer error: Command is not permitted
-112003	ERR_FP_OVERFLOW	Fiscal printer error: Register overflow

-112004	ERR_FP_WRONG_DATE_TIME	Fiscal printer error: Wrong date/time
-112005	ERR_FP_NEEDED_MODE_PC	Fiscal printer error: PC mode is needed
-112006	ERR_FP_NO_PAPER	Fiscal printer error: No paper
-112007	ERR_FP_COVER_IS_OPEN	Fiscal printer error: Cover is open
-112008	ERR_FP_PRINTER_FAILURE	Fiscal printer error: Printing mechanism error
(112100 - 112199) FP_MODE ERRORS BY SYNTAX		
-112100	_ERR_FP_SYNTAX_PARAM_BEGIN	_ERR_FP_SYNTAX_PARAM_BEGIN
-112101	ERR_FP_SYNTAX_PARAM_1	Invalid syntax of parameter 1.
-112102	ERR_FP_SYNTAX_PARAM_2	Invalid syntax of parameter 2.
-112103	ERR_FP_SYNTAX_PARAM_3	Invalid syntax of parameter 3.
-112104	ERR_FP_SYNTAX_PARAM_4	Invalid syntax of parameter 4.
-112105	ERR_FP_SYNTAX_PARAM_5	Invalid syntax of parameter 5.
-112106	ERR_FP_SYNTAX_PARAM_6	Invalid syntax of parameter 6.
-112107	ERR_FP_SYNTAX_PARAM_7	Invalid syntax of parameter 7.
-112108	ERR_FP_SYNTAX_PARAM_8	Invalid syntax of parameter 8.
-112109	ERR_FP_SYNTAX_PARAM_9	Invalid syntax of parameter 9.
-112110	ERR_FP_SYNTAX_PARAM_10	Invalid syntax of parameter 10.
-112111	ERR_FP_SYNTAX_PARAM_11	Invalid syntax of parameter 11.
-112112	ERR_FP_SYNTAX_PARAM_12	Invalid syntax of parameter 12.
-112113	ERR_FP_SYNTAX_PARAM_13	Invalid syntax of parameter 13.
-112114	ERR_FP_SYNTAX_PARAM_14	Invalid syntax of parameter 14.
-112115	ERR_FP_SYNTAX_PARAM_15	Invalid syntax of parameter 15.
-112116	ERR_FP_SYNTAX_PARAM_16	Invalid syntax of parameter 16.
-112199	_ERR_FP_SYNTAX_PARAM_END	_ERR_FP_SYNTAX_PARAM_END
(112200 - 112299) FP_MODE ERRORS BY VALUE		
-112200	_ERR_FP_BAD_PARAM_BEGIN	_ERR_FP_BAD_PARAM_BEGIN
-112201	ERR_FP_BAD_PARAM_1	Bad value of parameter 1.
-112202	ERR_FP_BAD_PARAM_2	Bad value of parameter 2.
-112203	ERR_FP_BAD_PARAM_3	Bad value of parameter 3.
-112204	ERR_FP_BAD_PARAM_4	Bad value of parameter 4.
-112205	ERR_FP_BAD_PARAM_5	Bad value of parameter 5.
-112206	ERR_FP_BAD_PARAM_6	Bad value of parameter 6.
-112207	ERR_FP_BAD_PARAM_7	Bad value of parameter 7.
-112208	ERR_FP_BAD_PARAM_8	Bad value of parameter 8.
-112209	ERR_FP_BAD_PARAM_9	Bad value of parameter 9.
-112210	ERR_FP_BAD_PARAM_10	Bad value of parameter 10.
-112211	ERR_FP_BAD_PARAM_11	Bad value of parameter 11.
-112212	ERR_FP_BAD_PARAM_12	Bad value of parameter 12.
-112213	ERR_FP_BAD_PARAM_13	Bad value of parameter 13.
-112214	ERR_FP_BAD_PARAM_14	Bad value of parameter 14.
-112215	ERR_FP_BAD_PARAM_15	Bad value of parameter 15.
-112216	ERR_FP_BAD_PARAM_16	Bad value of parameter 16.
-112299	_ERR_FP_BAD_PARAM_END	_ERR_FP_BAD_PARAM_END

(114000 - 114997) POS TERMINAL ERRORS		
-114000	ERR_POS_TERM_CHAN_CLOSED	POS- terminal error: Communication channel is closed
(118000 - 118999) ONLINE ERRORS		
-118000	ERR_ECRSRV_NO_SOCKET_OPENED	ECR server error: The connection socket is not open
-118001	ERR_ECRSRV_SET_IS_NOT_TAKEN	ECR server error: The set for this command is not opened
-118002	ERR_ECRSRV_WRONG_PARAM	ECR server error: Wrong parameter
-118003	ERR_ECRSRV_NOT_SEND	ECR server error: Socket send error. Could not send data to server
-118004	ERR_ECRSRV_RECV_TMOUT	ECR server error: Receiving timeout. No data is receivec on time
-118005	ERR_ECRSRV_SOCKET_CLOSED	ECR server error: Socket is closed
-118006	ERR_ECRSRV_UNKNOWN	ECR server error: Unknown state
-118007	ERR_ECRSRV_FORBIDEN	ECR server error: Forbidden operation
(120000 - 120999) PROGRAMMING ERROR		
-120000	ERR_PGM_NAME_NOT_UNIQUE	Programming: Name is not unique!
-120001	ERR_PGM_OPER_PASS_NOT_UNIQUE	Programming: Operator password is not unique!
-120002	ERR_PGM_DATETIME_OUT_OF_RANGE_MIN	Programming: Date and time is under the range.
-120003	ERR_PGM_DATETIME_OUT_OF_RANGE_MAX	Programming: Date and time is under the range.
(121000 - 121099) COURIER ERRORS		
-121000	ERR_SCANNER_GENERAL	Barcode scanner reading error!
-121001	ERR_COURIER_EIK_INVALID	Invalid EIK/EGN number!
(170000 - 170999) PENDRIVE ERRORS		
-170000	ERR_USB_HOST_INIT	USB error: Host init error
-170001	ERR_USB_NO_DEVICE	USB error: No device
-170002	ERR_USB_NO_FILESYSTEM	USB error: No filesystem
-170003	ERR_USB_FILE_OPEN	USB error: File open error
-170004	ERR_USB_FILE_COPY	USB error: File copy error
-170005	ERR_FILE_UNPACK	USB error: File unpack error