

Communication program

Txt2500

for Windows 9x/NT/2000

Version 4.02

for Euro-2500TX cash registers

Document version 4.02.00

Document revision

Version 3.01.00 changes compared to 3.00.00

- added functions for modem communication

Version 3.02.00 changes

- corrected ranges of set items ID

Version 3.03.00 changes

- changed function for deleting all set records

Version 3.04.00 changes

- corrected error codes

Version 4.00.00 changes

- communication speed of 19,200 Bd was added
- the telephone number dialing philosophy was changed
- the structure of sets and functions was changed
- ID parameters were changed

Version 4..00.01 changes

- incorrect customer name length corrected - page 18
- added description of function for execute financial report - page 56

Version 4.01.00 changes

- added legend of used variable types – page 5
- added note for report types for fiscal countries – page 26
- added note for master currency for fiscal countries – page 26
- added note pre TAX levels for fiscal countries – page 30
- added note pre identification numbers for fiscal countries – page 31
- changed read / write of RAWSTRING parameters – reading and writing from / to files is in hexadecimal strings
- added note for journal / log – page 48 / 49
- added new fiscal functions – page 52
- added description for „Set master currency, TAX levels, Reregister ECR“ - pages 71÷74
- added „Reading and Decoding of FM“ - pages 75 ÷ 77
- added description fo FM – pages 78÷79

Version 4.02.00 changes

- better error detection in programm
- removed „Decoding of FM“ - page 76

1. Free Version License

By receiving and/or using Elcom communication software, you accept the following Free Version User Agreement. This agreement is a binding legal agreement between Elcom s.r.o. and the purchasers, users or evaluators of Elcom software and products. If you do not intend to honor this agreement, remove all installed Elcom software from your computer now.

2. Free Version Distribution

This Free Version software may be freely distributed, provided that:

1. Such distribution includes only the original archive supplied by Elcom s.r.o. You may not alter, delete or add any files in the distribution archive.
2. No money is charged to the person receiving the software, beyond reasonable cost of packaging and other overhead.

For information about redistribution of Elcom communication...

WWW: <http://www.elcom.sk>

E-mail: elcom@elcom.sk

3. Free Version User Agreement

3.1 Usage and distribution restrictions

The evaluator/user/buyer/owner may not use or distribute Elcom software for any unlawful purpose. The evaluator/user/buyer/owner is not allowed to attempt to disassemble or decompile Elcom Software and products.

3.2 Copyright restriction

Elcom name and any logo or graphics file that represents our software may not be used in any way to promote products developed with our software. All parts of Elcom software and products are copyright protected.

3.3 Limitation of responsibility

The evaluator/user/buyer/owner of Elcom software will indemnify and defend Elcom against lawsuits, claims, costs associated with defense or accusations that result from the use of Elcom software.

Elcom is not responsible for any damages whatsoever, including loss of information, interruption of business, personal injury and/or any damage or consequential damage without limitation, incurred before, during or after the use of our products.

1 Program description

- It is 32-bits console application for Windows® operating system
- The program provides a communication between PC and Euro 2500TX cash registers
- It allows to read and write programmable data of ECR (sets, parameters)
- reading of sale data (PLUs, DPTs, ...)
- execution of the functions (PLU store change, reports, ...)

2 Program parameters

TXT2500.EXE <portNum> /<dataTypeId> <opType> <firstItemNum> <lastItemNum**> <dataFileName**> <headerFileName**>**

- *portNum* – serial port number which is the ECR connected to: (1 – 4)
- *dataTypeId* – data type:
 - /s - set (see chapter 4.1)
 - /p - parameter (see chapter 4.2)
 - /f - function (see chapter 4.3)
- *ID* - number of set or function
 - if *n* character is added, will be received only non zero sale items (for PLU set type only)
- *opType* – operation type:
 - 0 - programming (PC --> ECR)
 - 1 - reading (ECR --> PC)
 - 2 - erasing *
 - 3 - reading set structure *
- *firstItemNum*** – number of the first transmitted item
 - if the value is omitted or is '0', the transmitted data will begin with the first item
- *lastItemNum*** – number of the last transmitted item
 - if the value is omitted or is '0', the transmitted data will end at the last item
- *dataFileName*** – name of data file (default data.in, data.out)
- *headerFileName*** – name of header file (default head.in, head.out)

* - for set data type only

** - the parameter is not needed

3 The files structures

3.1 Configuration file structure

- The base program setting is saved in setup.dat configuration file:
 1. line – link ECR address * : (2 – 32)
 2. line – transmission speed * : (9600, 19200 or 38400)
 3. line – number of decimal places for price displaying: (0 – 4)
 4. line – number of decimal places for quantity displaying: (0 – 4)
 5. line – diacritic coding:
 - 0 – English
 - 1 – Slovak
 - 2 – Czech
 - 3 – Portugal
 6. line – ECR name *: max. 20 characters, can be used only characters: A – Z, a – z, 0 – 9 and 'space'
 7. line – phone number **: can be used only characters: 0 – 9

* - the values have to be identical with the values programmed in ECR.

** - if line 7 exist, entered phone number will be dialed by connected modem, transmission speed must be set to 9600Bd or to 19200Bd

Note: The modem connection can also be established using the external application **dialer.exe**; in this case the telephone number is not entered into the 'setup.dat' file.

3.2 Base data types

ID	Name	Length	Description
0	T_PRICE	8B	signed value with 4 decimal places
1	T_QUANTITY	8B	signed value with 4 decimal places
2	T_PERCENT	2B	unsigned value with 2 decimal places
3	T_FLAG1	1B	unsigned value
4	T_FLAG2	2B	unsigned value
5	T_FLAG4	4B	unsigned value
6	T_STRING	nB	string encoded by internal ECR code table
7	T_RAWSTRING	nB	array of bytes
8	T_BARCODE	8B	unsigned value

TAB 3.2 Base data types

3.3 Header files structure

- The list of transmitted data is stored in header files (default *head.in*, resp. *head.out*):
 - *head.in* – data transmission from ECR
 - *head.out* - data transmission to ECR
- the structure of each of them as following:

ID:valType;valLen

- *ID* – identification number of set attribute or parameter (according to transmitted data type)
- *valType* – attribute type
- *valLen* – the length of attribute type

3.4 Data file structure

- Data transmitted to/from ECR are stored in data files (default *data.in*, *data.out*):
 - *data.in* – data transmission from ECR
 - *data.out* - data transmission to ECR
- data files has following structure:
 - (a) *set* transmission: - **one line** of the file contains attributes of **one set item** ordered as the header file defines; particular attributes are separated with ';' (semicolon) character.
 - (b) *parameter* transmission: - **each line** of the file contains **one system parameter** ordered as the header file defines.
 - (c) *function* execution: - **one line** of the file contains **all function parameters** ordered as the header file defines.

Notes:

- Separating character ';' defines (in the file) the end of one data and the start of the next one, it means the character ';' **can not** be placed in data (e.g. in PLU name, in logo line and so on)
- All texts (PLU names, logo and so on) from the files are in **windows code page**, the program converts them to internal ECR codes by sending and reading.
- Used can be only characters from internal character set of ECR (see following table)
- The particular lines of the file are **separated with „new line“ character**
- The ECR can print the characters with double width – you need to enter the '\$' character.
- The user **responds** that each bar code is assigned to only one PLU in ECR after the bar code transmission.
- The numerical data as T_PRICE a T_QUANTITY are stored in data files (with number of decimal places according to setting in configuration file) after reading from ECR.
- Input data files can contain numerical data with decimal separator, in this case they will be changed to particular number of decimal places according to data type.
- It is necessary to observe dedicated **ranges and types of values** to provide the proper interpretation of values sent to the cash register and proper operation of cash register!!!

En	!"#\$%&'()*+,-./0123456789:;<=>?@[€]^_`{ }→← ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
----	--

3.5 Log file structure

- result of each transmitted item is stored in *data.log* file with the following structure:

ID - errVal

- ID* - identification number of set item, function or parameter (according to transmitted data type)
- errVal* - error number

3.6 Error file structure

- result of communication is stored in *err* file with the following structure:

err_num;oper_num

- err_num* - error number
- oper_num* - operation number

Note: If error isn't depend to any operation (errors from 28672), *oper_num* value is missing

Error number		Error description
Hex	Dec	
0	0	Operation done without any error
0x0001 to 0x1000	1 to 4096	HW error – occurs by function executing or parameter setting
0x3001 to 0x3FFF	12289 to 16383	communication errors
0x3001	12289	incorrect operation or data identifier
0x3002	12290	incorrect data type
0x3003	12291	incorrect data length
0x3004	12292	incorrect data value
0x3005	12293	incorrect operator value
0x3006	12294	<i>reserved</i>
0x3007	12295	incorrect command parameter
0x3008	12296	cannot change system parameter – change condition is not met
0x3009	12297	cannot change set item data ¹
0x300A	12298	cannot read set item data
0x300B	12299	cannot create object
0x300C	12300	connection is not created
0x300D	12301	connection always exists
0x300E	12302	set doesn't exist
0x300F	12303	incorrect set operation
0x3010	12304	object is not created
0x3011	12305	incorrect object size
0x3012	12306	incorrect data object number
0x3013	12307	<i>reserved</i>
0x3014	12308	primary key for object reading is not entered
0x3015	12309	cannot read system parameter
0x3016	12310	incorrect operator for set reading
0x3017	12311	function always exists
0x3018	12312	function is not created
0x3019	12313	function doesn't exist
0x301A	12314	function is executing
0x301B	12315	incorrect number of function parameters
0x301C	12316	function is not executing
0x301D	12317	incorrect function parameters
0x301E	12318	incorrect function parameters
0x301F	12319	cannot change data – change condition is not met ²

0x3020	12320	cannot remove data – remove condition is not met ³
0x3021	12321	incorrect command
0x3022	12322	incorrect packet length
0x3023	12323	incorrect „connect“ string
0x3024	12324	unknown error
0x4000 to 0x6FFF	16384 to 28671	application and fiscal core error – see ECR user manual

0x7000	28672	Missing input parameter
0x7001	28673	Incorrect port number
0x7002	28674	Incorrect data type
0x7003	28675	Incorrect operation type
0x7004	28676	Missing <i>setup.dat</i> file
0x7005	28677	Incorrect ECR link address in <i>setup.dat</i> file
0x7006	28678	Incorrect communication speed in <i>setup.dat</i> file
0x7007	28679	Incorrect country in <i>setup.dat</i> file
0x7008	28680	Incorrect number of decimal places for display price in <i>setup.dat</i> file
0x7009	28681	Incorrect number of dec. places for display quantity in <i>setup.dat</i> file
0x700A	28682	Incorrect ECR name in <i>setup.dat</i> file
0x700B	28683	Incorrect or missing data file
0x700C	28684	Incorrect or missing header file
0x700D	28685	Communication error (time out)
0x700E	28686	Open port error
0x7010	28688	Dial error

¹ occurs if attempt is made to change read-only attributes or add new record with read-only attributes

² occurs if one attempts to change the record using values that are not within the valid range or the conditions for changing the record are not observed:

for surcharges/discounts:

- given surcharge/discount was already used and one wants to change its type or its settings, regardless whether it is value/percent surcharge/discount;

for payments:

- if the payment is of customer credit type, it must be set to the master currency
- and concurrently, if it is a foreign currency, it must be of cash type
- if the payment was already used in a sale, then the old and the new record must conform to following rules: identical abbreviation, number of decimal places for rounding and displaying, the rounding method, payment type, and currency type (master, secondary, or other)

Note: In case if the payment changed to the payment in master or secondary payment, then the payment settings are read from the fiscal memory settings for master or secondary currency.

for cashiers:

- the cashier password mustn't be the same as of any other existing cashier
- and if the cashier is logged in at present, one cannot change his/her access rights for P, X, R, T, and Z modes, authorisation for performing of RA, PO, correction receipt, bill merging/splitting operations, bill transfer to another cashier, bill opening, operations upon stock and customer credit, and the article re-valuation.

for customers:

- the customer mustn't be assigned to a bill

for linked PLUs:

- index of the master PLU record mustn't be duplicate for linked PLUs and linked texts

³ occurs if one attempts to erase a record that is already out from the database or the conditions for erasing a record are not observed. In general, these conditions have to be observed:

for PLUs:

- if one wants to erase a PLU that is present in a sale, one has to perform the PLU Z report. Then it is possible to erase such PLU

for surcharges/discounts:

- if one wants to erase a used surcharge/discount or to erase a pre-defined promotional discounts, VIP discounts, or customer discounts, one has to perform the Surcharge/discount Z report. Then it is possible to erase the used record.

for departments and tables:

- if one wants to erase the records already used in a sale, one has to perform the respective Z report

for cashier and his/her payment authorisations:

- if one wants to erase the cashier that is currently logged in or a cashier that finalised some bills, one has to perform the cashier Z report and then it is possible to erase such cashier

for customers:

- if the customer in question is assigned to a bill or he/she has paid for using his/her customer credit, it is necessary to perform the customers Z report prior erasing such customer

for payments:

- if one wants to erase the cash payment in master currency or the payment in question was already used in a sale, one has to perform the payments Z report prior erasing such an entry

4 Data types

4.1 Sets

SetID	Set name	Operation		
		Receive	Send	Remove
0	PLU	X	X	X
1	DPT	X	X	X
2	SURCHARGE / DISCOUNT	X	X	X
3	PAYMENT	X	X	X
4	CASHIER	X	X	X
5	TABLE	X	-	X
6	CUSTOMER	X	X	X
9	DPT GROUP	X	X	X
10	LINK PLU	X	X	X
11	PROMOTION	X	X	X
12	UNIT	X	X	X
13	DESCRIPTIVE TEXT	X	X	X
14	LINK TEXT	X	X	X
15	DIRECT KEY	X	X	X
16	ADITIVE TEXT	X	X	X
17	CUSTOMER TEXT	X	X	X
22	HAPPY HOUR	X	X	-
24	CASHIER PAYMENT	X	X	X
25	HOUR REPORT	X	-	-
27	REPORT TIMES	X	-	-

4.1.1 Description of sets structures

4.1.1.1 PLU (0)

MemID	Data type	Description
0	T_FLAG4	record ID: (1 ÷ 999 999)
1	T_BARCODE	bar code
2	T_STRING[20]	text
3	T_PRICE	price 1
4	T_PRICE	price 2
5	T_QUANTITY	amount on stock
6	T_FLAG1	link to promo table: 0 – no, 1 ÷ 255 – ID of record from promo table
7	T_FLAG1	assignment to DPT: 1 ÷ 64 – ID of record from DPT table
8	T_FLAG1	assignment to VAT level: (0 ÷ 6) (0 means no VAT)
9	T_FLAG1	assignment to sale unit: 0 – no, 1 ÷ 15 – ID of record from unit table
10	T_FLAG1	sale type: 0 – forbidden, 1 – preset, 2 – open, 3 – preset & open
11	T_FLAG1	type of registration: 0 – positive sale, 1 – negative (refund)
12	T_FLAG1	returnable pack: 0 – no, 1 – yes
13	T_FLAG1	price restriction (HALO): (0 ÷ 9)
14	T_FLAG1	print stocks on PLU report: 0 – no, 1 – yes
15	T_FLAG1	allow fractional quantity sale: 0 – no, 1 – yes
16	T_FLAG1	allow sale to negative stock value: 0 – no, 1 – yes
17	T_FLAG1	obtain quantity from scales: 0 – no, 1 – yes
18	T_FLAG1	ask for message from cashier: 0 – no, 1 – yes
19	T_FLAG1	print PLU on external printer: 0 – do not print to external printer, 1 ÷ 7 – external printer number
20	T_FLAG1	active happy hour 1 (second price): 0 – no, 1 – yes
21	T_FLAG1	active happy hour 2 (second price): 0 – no, 1 – yes
22	T_FLAG1	active happy hour 3 (second price): 0 – no, 1 – yes
23	T_FLAG1	active happy hour 4 (second price): 0 – no, 1 – yes

24	T_FLAG1	Surcharge / discount: <i>0 – applicable,</i> <i>1 – not applicable</i>
25	T_FLAG1	PLU Link table should be checked *: <i>0 – no,</i> <i>1 – yes</i>
26	T_FLAG1	Text Link table should be checked *: <i>0 – no,</i> <i>1 – yes</i>
27	T_FLAG1	<i>reserved</i>
28	T_FLAG1	<i>reserved</i>
29	T_QUANTITY	sold quantity
30	T_PRICE	sold value

*read-only – informative data

4.1.1.2 DPT (1)

MemID	Data type	Description
0	T_FLAG1	record ID: $(1 \div 64)$
1	T_FLAG1	assignment to DPT group: $1 \div 16$ – ID of record from DPT group table
2	T_STRING[20]	text
3	T_QUANTITY	daily sold quantity
4	T_PRICE	daily sold negative value
5	T_PRICE	daily sold positive value
6	T_QUANTITY	monthly sold quantity
7	T_PRICE	monthly sold negative value
8	T_PRICE	monthly sold positive value
9	T_QUANTITY	daily sold quantity in Training mode
10	T_PRICE	daily sold negative value in Training mode
11	T_PRICE	daily sold positive value in Training mode
12	T_QUANTITY	monthly sold quantity in Training mode
13	T_PRICE	monthly sold negative value in Training mode
14	T_PRICE	monthly sold positive value in Training mode

~~4.01.00~~**4.1.1.3 Surcharge / Discount (2)**

MemID	Data type	Description
0	T_FLAG1	record ID: $(1 \div 255)$
1	T_STRING[20]	text
2	T_FLAG1	limit of surcharge / discount: $1 \div 100$ – if surcharge / discount is percentual, number of digits – if surcharge / discount is a fixed value
3	T_FLAG1	surcharge / discount: 0 – surcharge, 1 – discount
4	T_FLAG1	type: 0 – value, 1 – percentual
5	T_FLAG1	allow application to single item: 0 – no, 1 – yes
6	T_FLAG1	allow application to subtotal: 0 – no, 1 – yes
7	T_FLAG1	reserved
8	T_FLAG1	reserved
9	T_PRICE	value: 0 \div 100,00 for percentage type 0 \div 999.999.999,999 for valuable type
10	T_QUANTITY	daily sold quantity
11	T_PRICE	daily sold value
12	T_QUANTITY	monthly sold quantity
13	T_PRICE	monthly sold value

4.1.1.4 Payment (3)

MemID	Data type	Description
0	T_FLAG1	record ID: $(1 \div 99)$
1	T_STRING[20]	text of payment
2	T_STRING[3]	text of currency*
3	T_FLAG1	exchange rate coefficient ($final\ rate = rate / 10^{coefficient}$)*
4	T_PRICE	currency exchange rate (in regard to the master currency)*
5	T_FLAG1	number of decimal places for rounding: $(0 \div 3)$ *
6	T_FLAG1	number of decimal places for viewing: $(0 \div 3)$ *
7	T_FLAG1	type of rounding*: 0 – standard, 1 – up, 2 – down, 3 – special
8	T_FLAG1	entry of paid value is: 0 – optional, 1 – mandatory
9	T_FLAG1	allow drawer opening: 0 – no, 1 – yes
10	T_FLAG1	mandatory entry of descriptive number of payment: 0 – no, 1 – yes
11	T_FLAG1	payment type: 0 – cash, 1 – cheque, 2 – credit, 3 – customer credit

12	T_FLAG1	<i>reserved</i>
13	T_FLAG1	print out the payment text on receipt: <i>0 – no</i> <i>1 – yes</i>
14	T_FLAG1	allow using payment terminal (only for <i>credit</i> payment type): <i>0 – no</i> <i>1 – yes</i>
15	T_FLAG1	currency type: <i>0 – master,</i> <i>1 – secondary,</i> <i>2 – other</i>
16	T_FLAG1	currency text position *: <i>0 – behind price,</i> <i>1 – in front of price</i>
17	T_FLAG1	<i>reserved</i>
18	T_FLAG1	<i>reserved</i>
19	T_QUANTITY	daily number of receive on account to drawer
20	T_PRICE	daily value of receive on account to drawer
21	T_QUANTITY	daily number of paid out from drawer
22	T_PRICE	daily value of paid out from drawer
23	T_QUANTITY	daily number of payments
24	T_PRICE	daily value of payments
25	T_PRICE	daily value in drawer1
26	T_PRICE	daily value in drawer2
27	T_QUANTITY	monthly number of receive on account to drawer
28	T_PRICE	monthly value of receive on account to drawer
29	T_QUANTITY	monthly number of paid out from drawer
30	T_PRICE	monthly value of paid out from drawer
31	T_QUANTITY	monthly number of payments
32	T_PRICE	monthly value of payments
33	T_PRICE	monthly value in drawer1
34	T_PRICE	monthly value in drawer2

* for currency type 0 (master) and 1 (secondary) fiscal–core setting is used (see. chapter 3.2.1.1. *Master currency* and 3.2.1.2. *Secondary currency*)

~~4.01.00~~**4.1.1.5 Cashier (4)**

MemID	Data type	Description
0	T_FLAG1	record ID: $(I \div 255)$
1	T_STRING[20]	name
2	T_FLAG4	password1 – entered via keyboard
3	T_FLAG4	password2 – entered via RF key
4	T_FLAG1	authorised for ECR R mode: <i>0 – no,</i> <i>1 – yes</i>
5	T_FLAG1	authorised for ECR X mode: <i>0 – no,</i> <i>1 – yes</i>
6	T_FLAG1	authorised for ECR Z mode: <i>0 – no,</i> <i>1 – yes</i>
7	T_FLAG1	authorised for ECR P mode: <i>0 – no,</i> <i>1 – yes</i>
8	T_FLAG1	authorised for ECR T mode: <i>0 – no,</i> <i>1 – yes</i>
9	T_FLAG1	<i>reserved</i>
10	T_FLAG1	allow VOID operation: <i>0 – no,</i> <i>1 – yes</i>
11	T_FLAG1	allow SUBTOTAL VOID operation: <i>0 – no,</i> <i>1 – yes</i>
12	T_FLAG1	allow REFUND operation: <i>0 – no,</i> <i>1 – yes</i>
13	T_FLAG1	allow Correction Mode accessibility: <i>0 – no,</i> <i>1 – yes</i>
14	T_FLAG1	allow DISCOUNT operation: <i>0 – no,</i> <i>1 – yes</i>
15	T_FLAG1	allow SURCHARGE operation: <i>0 – no,</i> <i>1 – yes</i>
16	T_FLAG1	allow VIP operation: <i>0 – no,</i> <i>1 – yes</i>
17	T_FLAG1	allow RA operation: <i>0 – no,</i> <i>1 – yes</i>
18	T_FLAG1	allow PO operation: <i>0 – no,</i> <i>1 – yes</i>
19	T_FLAG1	allow opening of Drawer 1: <i>0 – no,</i> <i>1 – yes</i>
20	T_FLAG1	allow opening of Drawer 2: <i>0 – no,</i> <i>1 – yes</i>
21	T_FLAG1	allow opening of a bill: <i>0 – own bills only,</i> <i>1 – all bills</i>

22	T_FLAG1	allow closing of a bill: <i>0 – no, 1 – own bills only, 2 – all bills</i>
23	T_FLAG1	allow using other price than programmed: <i>0 – no, 1 – yes</i>
24	T_FLAG1	allow using second price: <i>0 – no, 1 – yes</i>
25	T_FLAG1	allow moving bill to another cashier: <i>0 – no, 1 – yes</i>
26	T_FLAG1	allow bill merging and separation: <i>0 – no, 1 – yes</i>
27	T_FLAG1	allow stock operation (add, withdraw or replace): <i>0 – no, 1 – yes</i>
28	T_FLAG1	allow changes in customer credit (add, withdraw, positive and negative replace): <i>0 – no, 1 – yes</i>
29	T_FLAG1	allow DUPLICATE operation: <i>0 – no, 1 – yes</i>
30	T_FLAG1	allow INVOICE operation: <i>0 – no, 1 – yes</i>
31	T_FLAG1	allow changes in PLUs price: <i>0 – no, 1 – yes</i>
32	T_FLAG1	allow assigning non-existing customer to the bill and insert him/her into the database: <i>0 – no, 1 – yes</i>
33	T_FLAG1	<i>reserved</i>
34	T_FLAG1	<i>reserved</i>
35	T_FLAG1	<i>reserved</i>
36	T_FLAG1	<i>reserved</i>
37	T_FLAG1	<i>reserved</i>
38	T_FLAG1	<i>reserved</i>
39	T_FLAG1	<i>reserved</i>
40	T_FLAG1	<i>reserved</i>
41	T_QUANTITY	daily number of customer serving
42	T_PRICE	daily value of customer serving
43	T_QUANTITY	daily number of paid bills
44	T_PRICE	daily value of paid bills
45	T_QUANTITY	daily number of void operations
46	T_PRICE	daily value of void operations
47	T_QUANTITY	daily number of delete operations
48	T_PRICE	daily value of delete operations
49	T_QUANTITY	daily number of subtotal void operations
50	T_PRICE	daily value of subtotal void operations
51	T_QUANTITY	daily number of subtotal delete operations
52	T_PRICE	daily value of subtotal delete operations
53	T_QUANTITY	daily number of refund items
54	T_PRICE	daily value of refund items
55	T_QUANTITY	daily number of correction bills

56	T_PRICE	daily value of correction bills
57	T_QUANTITY	daily number of surcharge operations
58	T_PRICE	daily value of surcharge operations
59	T_QUANTITY	daily number of discount operations
60	T_PRICE	daily value of discount operations
61	T_QUANTITY	daily number of VIP operations
62	T_PRICE	daily value of VIP operations
63	T_QUANTITY	daily number of pack sold
64	T_PRICE	daily value of pack sold
65	T_QUANTITY	daily number of pack returned
66	T_PRICE	daily value of pack returned
67	T_QUANTITY	daily number of drawer1 openings
68	T_QUANTITY	daily number of drawer2 openings
69	T_QUANTITY	monthly number of customer serving
70	T_PRICE	monthly value of customer serving
71	T_QUANTITY	monthly number of paid bills
72	T_PRICE	monthly value of paid bills
73	T_QUANTITY	monthly number of void operations
74	T_PRICE	monthly value of void operations
75	T_QUANTITY	monthly number of delete operations
76	T_PRICE	monthly value of delete operations
77	T_QUANTITY	monthly number of subtotal void operations
78	T_PRICE	monthly value of subtotal void operations
79	T_QUANTITY	monthly number of subtotal delete operations
80	T_PRICE	monthly value of subtotal delete operations
81	T_QUANTITY	monthly number of refund items
82	T_PRICE	monthly value of refund items
83	T_QUANTITY	monthly number of correction bills
84	T_PRICE	monthly value of correction bills
85	T_QUANTITY	monthly number of surcharge operations
86	T_PRICE	monthly value of surcharge operations
87	T_QUANTITY	monthly number of discount operations
88	T_PRICE	monthly value of discount operations
89	T_QUANTITY	monthly number of VIP operations
90	T_PRICE	monthly value of VIP operations
91	T_QUANTITY	monthly number of pack sold
92	T_PRICE	monthly value of pack sold
93	T_QUANTITY	monthly number of pack returned
94	T_PRICE	monthly value of pack returned
95	T_QUANTITY	monthly number of drawer1 opening
96	T_QUANTITY	monthly number of drawer2 opening

~~4.01.00~~**4.1.1.6 Table (5)**

MemID	Data type	Description
0	T_FLAG1	record ID: (1 ÷ 99)
1	T_QUANTITY	daily sold quantity
2	T_PRICE	daily sold value
3	T_QUANTITY	monthly sold quantity
4	T_PRICE	monthly sold value

4.1.1.7 Customer (6)

MemID	Data type	Description
0	T_FLAG4	record ID: (1 ÷ 999 999 999)
1	T_STRING[24]	customer's name
2	T_FLAG1	programmed discount: (0 ÷ 100)
3	T_FLAG1	allow sale payment with customer credit: 0 – no, 1 – yes
4	T_FLAG1	allow other discount than customer discount: 0 – no, 1 – yes
5	T_FLAG1	reserved
6	T_FLAG1	print customer credit at the end of sale (customer must be assigned to the bill): 0 – no, 1 – only if paid by customer credit 2 – always
7	T_FLAG1	reserved
8	T_FLAG1	reserved
9	T_PRICE	limit of credit value
10	T_PRICE	value of credit value
11	T_QUANTITY	daily sold quantity
12	T_PRICE	daily sold value

4.1.1.8 DPT GROUP (9)

MemID	Data type	Description
0	T_FLAG1	record ID: (1 ÷ 16)
1	T_STRING[20]	Name

~~4.01.00~~**4.1.1.9 LINK PLU (10)**

MemID	Data type	Description
0	T_FLAG2	record ID: $(1 \div 1000)$
1	T_FLAG4	master PLU: $1 \div 999\,999$ – ID of record from PLU table
2	T_FLAG4	linked PLU 1: 0 – no, $1 \div 999\,999$ – ID of record from PLU table
3	T_FLAG4	linked PLU 2: 0 – no, $1 \div 999\,999$ – ID of record from PLU table
4	T_FLAG4	linked PLU 3: 0 – no, $1 \div 999\,999$ – ID of record from PLU table
5	T_FLAG4	linked PLU 4: 0 – no, $1 \div 999\,999$ – ID of record from PLU table
6	T_FLAG4	linked PLU 5: 0 – no, $1 \div 999\,999$ – ID of record from PLU table
7	T_FLAG4	linked PLU 6: 0 – no, $1 \div 999\,999$ – ID of record from PLU table

4.1.1.10 Promotion (11)

MemID	Data type	Description
0	T_FLAG1	record ID: $(1 \div 255)$
1	T_FLAG1	type: 0 – <i>percentual</i> , 1 – <i>value</i>
2	T_QUANTITY	qualifying quantity
3	T_QUANTITY	promotion quantity
4	T_PRICE	value of promotion (percentage or value according to type) always on 3 decimal places – 1% = 1000 or 1EUR=1000

4.1.1.11 Unit (12)

MemID	Data type	Description
0	T_FLAG1	record ID: $(1 \div 15)$
1	T_STRING[3]	Text

4.1.1.12 Descriptive text (13)

MemID	Data type	Description
0	T_FLAG2	record ID: $(1 \div 65\,535)$
1	T_STRING[24]	Text

~~4.01.00~~**4.1.1.13 Link text (14)**

MemID	Data type	Description
0	T_FLAG2	record ID: ($1 \div 65\,535$)
1	T_FLAG4	master PLU: $1 \div 999\,999$ – ID of record from PLU table
2	T_FLAG2	linked text 1: 0 – no, $1 \div 65\,535$ – ID of record from descriptive text table
3	T_FLAG2	linked text 2: 0 – no, $1 \div 65\,535$ – ID of record from descriptive text table
4	T_FLAG2	linked text 3: 0 – no, $1 \div 65\,535$ – ID of record from descriptive text table
5	T_FLAG2	linked text 4: 0 – no, $1 \div 65\,535$ – ID of record from descriptive text table
6	T_FLAG2	linked text 5: 0 – no, $1 \div 65\,535$ – ID of record from descriptive text table
7	T_FLAG2	linked text 6: 0 – no, $1 \div 65\,535$ – ID of record from descriptive text table

4.1.1.14 Direct key (15)

MemID	Data type	Description
0	T_FLAG2	record ID: ($1 \div 560$)
1	T_FLAG1	set type: 0 – PLU, 1 – Payment, 2 – Surcharge / Discount, 3 – Additive text
2	T_FLAG4	ID of record according to type (depending of set type)

4.1.1.15 Additive text (16)

MemID	Data type	Description
0	T_FLAG1	record ID: ($1 \div 255$)
1	T_STRING[20]	Text

4.1.1.16 Customer text (17)

MemID	Data type	Description
0	T_FLAG4	record ID: ($1 \div 999\,999\,999$)
1	T_STRING[24]	text 1
2	T_STRING[24]	text 2
3	T_STRING[24]	text 3
4	T_STRING[24]	text 4
5	T_STRING[24]	text 5

~~4.01.00~~**4.1.1.17 Happy hours (22)**

MemID	Data type	Description
0	T_FLAG1	record ID: $(1 \div 4)$
1	T_FLAG1	start time in tens of minutes: $(0 \div 235)$
2	T_FLAG1	end time in tens of minutes: $(0 \div 235)$
3	T_FLAG1	Monday: <i>0 - inactive,</i> <i>1 - active</i>
4	T_FLAG1	Tuesday: <i>0 - inactive,</i> <i>1 - active</i>
5	T_FLAG1	Wednesday: <i>0 - inactive,</i> <i>1 - active</i>
6	T_FLAG1	Thursday: <i>0 - inactive,</i> <i>1 - active</i>
7	T_FLAG1	Friday: <i>0 - inactive,</i> <i>1 - active</i>
8	T_FLAG1	Saturday: <i>0 - inactive,</i> <i>1 - active</i>
9	T_FLAG1	Sunday: <i>0 - inactive,</i> <i>1 - active</i>
10	T_FLAG1	Happy hour actual activity: <i>0 - no,</i> <i>1 - yes</i>

4.1.1.18 Cashier payment (24)

MemID	Data type	Description
0	T_FLAG2	record ID: $(\text{Cashier ID} * 100) + \text{Payment ID}$
1	T_FLAG1	<i>0 - drawer1 is affected,</i> <i>1 - drawer2 is affected</i>
2	T_QUANTITY	daily number of receive on account to drawer
3	T_PRICE	daily value of receive on account to drawer
4	T_QUANTITY	daily number of paid out from drawer
5	T_PRICE	daily value of paid out from drawer
6	T_QUANTITY	daily number of payments
7	T_PRICE	daily value of payments
8	T_QUANTITY	monthly number of receive on account to drawer
9	T_PRICE	monthly value of receive on account to drawer
10	T_QUANTITY	monthly number of paid out from drawer
11	T_PRICE	monthly value of paid out from drawer
12	T_QUANTITY	monthly number of payments
13	T_PRICE	monthly value of payments

4.1.1.19 Hour report (25)

MemID	Data type	Description
0	T_FLAG1	record ID: $(I \div 24)$
1	T_QUANTITY	daily sold quantity
2	T_PRICE	daily sold value
3	T_QUANTITY	monthly sold quantity
4	T_PRICE	monthly sold value

4.1.1.20 Report times (27)

MemID	Data type	Description
0	T_FLAG1	record ID: $(1 \div 9)$
1	T_FLAG1	set ID
2	T_STRING[6]	last daily Z report date (DDMMYY)
3	T_STRING[6]	last daily Z report time (HHMMSS)
4	T_STRING[6]	last monthly Z report date (DDMMYY)
5	T_STRING[6]	last monthly Z report time (HHMMSS)

4.1.2 Set items reading

txt2500.exe <portNum> /s[SetID] 1 <firstItemNum> <lastItemNum>

or

txt2500.exe <portNum> /s[SetID] 1 <firstItemNum> <lastItemNum> <dataFileName> <headerFileName>

Example:

- The command reads the items of set No. 0 (PLU) from 1 to 100 from ECR connected to COM1 port; only attributes listed in *head.in* file will be read:

txt2500.exe 1 /s0 1 1 100

- Example of *head.in* file for reading 6 attributes of PLU:

```
0:4;4 //PLU ID
1:8;8 //bar code
2:6;20 //name
3:0;8 //price1
4:0;8 //price2
5:1;8 //stock
```

- Example of *data.in* file (price and quantity are displaying at 4 decimal places):

```
1;1234567890123456;Milk;20,0000;15,0000;10,0000
2;9876543210987654;Bread;15,0000;10,0000;10,0000
:
```

- for reading PLUs with non zero sale command has a following format:

txt2500.exe 1 /s0n 1

Examples:

- all items will be read

txt2500.exe 1 /s0 1

or

txt2500.exe 1 /s0 1 0 0

- the items from the range 100 to last item will be read

txt2500.exe 1 /s0 1 100

or

txt2500.exe 1 /s0 1 100 0

- the items from the range from first to 100. item will be read

txt2500.exe 1 /s0 1 1 100

or

txt2500.exe 1 /s0 1 0 100

- all items will be read along with the defined filenames for data file

txt2500.exe 1 /s0 1 0 0 <dataFileName>

- all items will be read along with the defined filenames for data and header file

txt2500.exe 1 /s0 1 0 0 <dataFileName> <headerFileName>

- the items from the range 100 – 1000 will be read along with the defined filenames for data and header file

txt2500.exe 1 /s0 1 1 100 <dataFileName> <headerFileName>

4.1.3 Programming of set attributes

txt2500.exe <portNum> /s[SetID] 0 <firstItemNum> <lastItemNum>

or

txt2500.exe <portNum> /s[SetID] 0 <firstItemNum> <lastItemNum> <dataFileName> <headerFileName>

Example:

- The command programs the set items No.0 (PLU) from 1 to 100 from the input data file to ECR connected to COM1 port (in case the range will be not defined the all items will be transmitted):

txt2500.exe 1 /s0 0 1 100

- Example of *data.out* file with the structure equal to *head.out* file:

```
1;1234567890123456;Milk;20,0000;15,0000;10,0000
2;9876543210987654;Bread;15,0000;10,0000;10,0000
:
```

- Example of *head.out* file for 6 PLU attributes programming:

```
0:4;4 //PLU ID
1:8;8 //bar code
2:6;20 //name
3:0;8 //price1
4:0;8 //price2
5:1;8 //stock
```


4.1.4 Set items removing

txt2500.exe <portNum> /s[SetID] 2 <firstItemNum> <lastItemNum>

or

txt2500.exe <portNum> /s[SetID] 2 <firstItemNum> <lastItemNum> <dataFileName> <headerFileName>

Example:

- the command erases the set items No.0 (PLU) defined in *data.out* file from ECR connected to COM1 port. The type of ID attribute has to be listed in *head.out* file:

txt2500.exe 1 /s0 2

- example of *data.out* file:

```
1
4
10
:
```

- only PLUs listed in *data.out* file will be erased in case the range is defined, e.g.:

txt2500.exe 1 /s0 2 1 100

- the all items of defined range will be erased if *data.out* file does not exist. The **all** items will be erased in case a range is not defined.

4.1.5 Set structure reading

txt2500.exe <portNum> /s[SetID] 3

or

txt2500.exe <portNum> /s[SetID] 3 0 0 <dataFileName> <headerFileName>

Example:

- The command reads the sets structure No.0 (PLU) from the ECR connected to port No. 1 and save the structure to *head.in* file:

txt2500.exe 1 /s0 3

4.2 Parameters

Legend:

^Y – flag can be changed after yearly report

^M – flag can be changed after monthly report

^D – flag can be changed after daily report

^G – flag can be changed when grand totals are cleared

Note: yearly (^Y) a monthly (^M) report are available only for fiscal only for non fiscal countries. In fiscal countries is only daily (^D) report therefore all requirements for yearly (^Y) a monthly (^M) reports are transferred to daily (^D).

column marked **En** contains default values for this national ECR version

bold character indicate fixed values (read-only)

4.2.1 Fiscal core parameters

4.2.1.1 Master currency

No.	ParID	Data type	Description	En
1	1	T_FLAG1	number of decimal places for prices rounding ^D : (0 – 3)	2
2	2	T_FLAG1	number of decimal places for taxes rounding ^D : (0 – 3)	2
3	3	T_FLAG1	number of decimal places for total sale price rounding ^D : (0 – 3)	2
4	4	T_FLAG1	number of decimal places for displaying taxes, prices and total sale price ^M : (0 – 3)	2
5	5	T_FLAG1	price rounding method ^Y : 0 – standard – 0.00 – 0.49 = 0.00, 0.50 – 0.99 = 1.00,, 1 – up, 2 – down, 3 – 0.00 – 0.25 = 0.00, 0.25 – 0.75 = 0.5, 0.75 – 1.00 = 1.00	0
6	6	T_FLAG1	taxes rounding method ^Y : (0 – 3)	0
7	7	T_FLAG1	total sale price rounding method ^Y : (0 – 3)	0
8	8	T_FLAG1	allow negative sum in drawer for master currency: 0 – no, 1 – yes	1
9	9	T_FLAG1	tax system ^G : 0 – European, 1 – American	0
10	10	T_FLAG1	currency text position ^M : 0 – behind price, 1 – in front of price, 2 – none	1
11	11	T_STRING[3]	master currency text ^M	\$

Note:

- flag 4 must be >= than flags 1 – 3
- before changing of 11th flag it is recommended to perform an annual report because in other case there will be mixed financial values for different currencies.
- master currency text can be an empty string and can't contain the numbers
- for fiscal countries are these parameters read only. To set master currency are used functions (page: 53)

~~4.01.00~~**4.2.1.2 Secondary currency**

No.	ParID	Data type	Description	En
1	51	T_PRICE	exchange rate of secondary currency to master currency: (1e1 – 1e12)	0,9
2	52	T_FLAG1	secondary currency is active ^M : 0 – no, 1 – yes	1
3	53	T_FLAG1	number of decimal places for rounding secondary currency total sale price ^D : (0 – 3)	2
4	54	T_FLAG1	number of decimal places for displaying secondary currency sale price ^M : (0 – 3)	2
5	55	T_FLAG1	secondary currency sale price rounding method: (0 – 3)	0
6	56	T_FLAG1	allow negative sum in drawer for secondary currency: 0 – no, 1 – yes	1
7	57	T_FLAG1	currency text position ^M : 0 – behind price, 1 – in front of price,	0
8	58	T_STRING[3]	secondary currency text ^M	EUR
9	59	T_FLAG1	power of 10 for exchange rate denominator* (0 – 9)	0

Note:

- flag 4 must be >= then flag 3
- before changing of 8th flag it is recommended to perform an annual report because in other case there will be mixed financial values for different currencies.
- secondary currency text can be an empty string and can't contain the numbers

*Example: If power is number 2 and exchange rate = 123,234 then resulting exchange rate = $123,234 / (10^2) = 1,23234$ is the actual exchange rate of secondary currency to master currency. It means 100 units of secondary currency represent 123,234 units of master currency

~~4.01.00~~**4.2.1.3 Various**

No.	ParID	Data type	Description	En
1	101	T_FLAG1	number of decimal places for quantity rounding: (0 – 3)	3
2	102	T_FLAG1	quantity rounding method: (0 – 3)	0
3	103	T_FLAG1	the way of sale finish if the pay sum overloads subtotal: 0 – finish immediately, 1 – finish immediately if pay sum is in master currency 2 – finish immediately only if difference between pay sum and subtotal is zero	0
4	104	T_FLAG1	not in use	0
5	105	T_FLAG1	automatic daily report: (0 – 240) 0 – not active, XX – after XX hours from the first sale	0
6	106	T_FLAG1	receipt finishing when subtotal is zero: 0 – normal, 1 – receipt voiding	0
7	107	T_FLAG1	printing in training mode: 0 – on the both sides, 1 – just to receipt, 2 – to journal, 3 – printer off, 4 – printing is controlled by application flags setting for printing and printing off in the training mode (ParID 1458 a 1460)	1
8	108	T_FLAG1	printing in registration mode: 0 – on the both sides, 1 – just to receipt, 2 – to journal, 3 – printer off, 4 – printing is controlled by application flags setting for printing and printing off in the registration mode (ParID 1458 a 1460)	0
9	109	T_FLAG1	other currency payment available (not secondary or master currency): 0 – no, 1 – yes	1
10	110	T_FLAG1	not in use	0
11	111	T_FLAG1	update stock information for current sale should be done after finishing sale: 0 – no, 1 – yes	1
12	112	T_FLAG1	allow negative value of total daily sales ^D : 0 – yes, 1 – daily sales in several tax levels can be negative, but total sum must be positive, 2 – daily sales in several tax levels must be positive	0
13	113	T_FLAG1	ban of the sale with zero PLU price : 0 – no, 1 – yes	0

~~4.01.00~~**4.2.1.4 Tax levels**

No.	ParID	Data type	Description	En
1	151	T_PERCENT	tax rate 1: 0 – 10000 (0.00% – 100.00%)	0
2	152	T_FLAG1	tax flag 1 ^M : 0 – VAT, 1 – TAX	0
3	153	T_FLAG1	tax counting method 1 ^M : 0 – at the end of day (for each tax 1 – 6), 1 – at the end of receipt	0
4	154	T_FLAG1	type of tax level: 0 – normal, 1 – non-taxable tax level	1
5	155	T_PERCENT	tax rate 2 ^M : (0 – 10000)	1000
6	156	T_FLAG1	tax flag 2 ^M : 0 – VAT, 1 – TAX	0
7	157	T_FLAG1	tax counting method 2 ^M : 0 – at the end of day (for each tax 1 – 6), 1 – at the end of receipt	0
8	158	T_FLAG1	type of tax level: 0 – normal, 1 – non-taxable tax level	0
9	159	T_PERCENT	tax rate 3 ^M : (0 – 10000)	2300
10	160	T_FLAG1	tax flag 3 ^M : 0 – VAT, 1 – TAX	0
11	161	T_FLAG1	tax counting method 3 ^M : 0 – at the end of day (for each tax 1 – 6), 1 – at the end of receipt	0
12	162	T_FLAG1	type of tax level: 0 – normal, 1 – non-taxable tax level	0
13	163	T_PERCENT	tax rate 4 ^M : 0 – 10000 – default 0	0
14	164	T_FLAG1	tax flag 4 ^M : 0 – VAT, 1 – TAX	0
15	165	T_FLAG1	tax counting method 4 ^M : 0 – at the end of day (for each tax 1 – 6), 1 – at the end of receipt	0
16	166	T_FLAG1	type of tax level: 0 – normal, 1 – non-taxable tax level	0
17	167	T_PERCENT	tax rate 5 ^M : (0 – 10000)	10000
18	168	T_FLAG1	tax flag 5 ^M : 0 – VAT, 1 – TAX	0
19	169	T_FLAG1	tax counting method 5 ^M : 0 – at the end of day (for each tax 1 – 6), 1 – at the end of receipt	0
20	170	T_FLAG1	type of tax level: 0 – normal, 1 – non-taxable tax level	0
21	171	T_PERCENT	tax rate 6 ^M : (0 – 10000)	10000
22	172	T_FLAG1	tax flag 6 ^M : 0 – VAT, 1 – TAX	0
23	173	T_FLAG1	tax counting method 2 ^M : 0 – at the end of day (for each tax 1 – 6), 1 – at the end of receipt	0

~~4.01.00~~

24	174	T_FLAG1	type of tax level: <i>0 – normal,</i> <i>1 – non-taxable tax level</i>	0
25	175	T_PERCENT	tax rate 7 ^M : <i>(0 – 10000)</i>	10000
26	176	T_FLAG1	tax flag 7 ^M : <i>0 – VAT,</i> <i>1 – TAX</i>	0
27	177	T_FLAG1	tax counting method 2 ^M : <i>0 – at the end of day (for each tax 1 – 6),</i> <i>1 – at the end of receipt</i>	0
28	178	T_FLAG1	type of tax level: <i>0 – normal,</i> <i>1 – non-taxable tax level</i>	0

- for fiscal countries are these parameters read only. To set master currency are used functions (page: 53)

4.2.1.5 Identification numbers

No.	ParID	Data type	Description	En
1	201	T_STRING[33]	identification number 1 *	
2	202	T_STRING[33]	identification number 2 **	
3	203	T_STRING[33]	identification number 3	
4	204	T_STRING[33]	identification number 4	
5	205	T_STRING[33]	identification number 5	
6	206	T_STRING[33]	identification number 6	
7	207	T_STRING[33]	identification number 7	
8	208	T_STRING[33]	identification number 8	
9	209	T_STRING[33]	identification number 9	
10	210	T_FLAG1	print ID 1: 0 – do not print 1 – at the top of the receipt, 2 – at the bottom of the receipt	1
11	211	T_FLAG1	print ID 2: 0 – do not print 1 – at the top of the receipt, 2 – at the bottom of the receipt	1
12	212	T_FLAG1	print ID 3: 0 – do not print 1 – at the top of the receipt, 2 – at the bottom of the receipt	1
13	213	T_FLAG1	print ID 4: 0 – do not print 1 – at the top of the receipt, 2 – at the bottom of the receipt	0
14	214	T_FLAG1	print ID 5: 0 – do not print 1 – at the top of the receipt, 2 – at the bottom of the receipt	0
15	215	T_FLAG1	print ID 6: 0 – do not print 1 – at the top of the receipt, 2 – at the bottom of the receipt	0
16	216	T_FLAG1	print ID 7: 0 – do not print 1 – at the top of the receipt, 2 – at the bottom of the receipt	0
17	217	T_FLAG1	print ID 8: 0 – do not print 1 – at the top of the receipt, 2 – at the bottom of the receipt	0
18	218	T_FLAG1	print ID 9: 0 – do not print 1 – at the top of the receipt, 2 – at the bottom of the receipt	0

*: for fiscal countries is this parameter read-only. Change is not possible.

**: for fiscal countries is this parameter read-only. Change is possible with function „Reregister ECR“ (page: 53)

4.2.1.6 Text logos

No.	ParID	Data type	Description	En
1	251	T_STRING[33]	1 st top text logo line	
2	252	T_STRING[33]	2 nd top text logo line	
3	253	T_STRING[33]	3 rd top text logo line	
4	254	T_STRING[33]	4 th top text logo line	
5	255	T_STRING[33]	5 th top text logo line	
6	256	T_STRING[33]	6 th top text logo line	
7	257	T_FLAG1	print top text logo lines: (0 – 6)	6
8	258	T_STRING[33]	1 st bottom text logo line	
9	259	T_STRING[33]	2 nd bottom text logo line	
10	260	T_STRING[33]	3 rd bottom text logo line	
11	261	T_STRING[33]	4 th bottom text logo line	
12	262	T_STRING[33]	5 th bottom text logo line	
13	263	T_STRING[33]	6 th bottom text logo line	
14	264	T_FLAG1	print bottom text logo lines: (0 – 6)	1
15	265	T_FLAG1	print top text logo on journal: 0 – no, 1 – yes	0
16	266	T_FLAG1	print bottom text logo on journal: 0 – no, 1 – yes	0

~~4-01-00~~**4.2.1.7 Functional texts**

No.	ParID	Data type	Description	En
1	301	T_STRING[20]	name of cashier – read-only	
2	302	T_STRING[20]	cash	CASH
3	303	T_STRING[20]	cheque	CHECK
4	304	T_STRING[20]	credit card	CREDIT CARD
5	305	T_STRING[20]	change	CHANGE
6	306	T_STRING[20]	pay away	PAY AWAY
7	307	T_STRING[20]	due	DUE
8	308	T_STRING[20]	total	TOTAL
9	309	T_STRING[20]	subtotal	SUBTOTAL
10	310	T_STRING[20]	subtotal void	SUBTOTAL VOID
11	311	T_STRING[20]	receipt voidance	RECEIPT VOIDANCE
12	312	T_STRING[20]	void	VOID
13	313	T_STRING[20]	refund	REFUND
14	314	T_STRING[20]	received on account	RA
15	315	T_STRING[20]	paid out	PO
16	316	T_STRING[20]	correction mode	CORRECTION MODE
17	317	T_STRING[20]	sold returnable pack	SOLD RETURNABLE PACK
18	318	T_STRING[20]	buy-out returnable pack	BUY OUT RETURN. PACK
19	319	T_STRING[20]	invoice	INVOICE
20	320	T_STRING[20]	duplicate receipt	DUPLICATE RECEIPT
21	321	T_STRING[20]	VIP-discount	VIP-DISCOUNT
22	322	T_STRING[20]	customer credit	CUSTOMER CREDIT
23	323	T_STRING[7]	returnable pack text abbreviation	RtnPack
24	324	T_STRING[24]	invalid receipt	INVALID RECEIPT
25	325	T_STRING[24]	printing interrupted	PRINTING INTERRUPTED
26	326	T_STRING[24]	sale interrupted	SALE INTERRUPTED
27	327	T_STRING[24]	report interrupted	REPORT INTERRUPTED

4.2.1.8 ECR number

No.	ParID	Data type	Description	En
1	351	T_FLAG1	ECR number: (1 – 99)	1
2	352	T_FLAG1	zeroing number of fiscal receipts in registration mode (N.o.F.R. in RM) and zeroing total number of receipts (T.N.o.R.) ^M 0 – N.o.F.R. in RM after daily report, T.N.o.R. after daily report (fixed), 1 – N.o.F.R. in RM after daily report, T.N.o.R. after monthly report, 2 – N.o.F.R. in RM after monthly report, T.N.o.R. after monthly report, 3 – N.o.F.R. in RM after daily report, T.N.o.R. never, 4 – N.o.F.R. in RM after monthly report, T.N.o.R. never, 5 – N.o.F.R. in RM never, T.N.o.R. never	0
3	353	T_FLAG1	zeroing number of fiscal receipts in training mode ^M : 0 – after daily report, 1 – after monthly report, 2 – never	0
4	354	T_FLAG1	Increasing of the counter „fiscal receipt number“ and „total receipt number“ although the receipts are not printed on the internal printer: 0 – counters are not increasing, 1 – only fiscal counters are increasing, 2 – all counters are increasing	0
5	355	T_FLAG1	Clearing the number of printed invoices in registration mode ^M : 0 – after daily report, 1 – after monthly report, 2 – never	0
6	356	T_FLAG1	Clearing the number of printed invoices in training mode ^M : 0 – after daily report, 1 – after monthly report, 2 – never	0

4.2.1.9 Date and time

No.	ParID	Data type	Description
1	401	T_STRING[6]	Date (DD-MM-YY)
2	402	T_STRING[6]	Time (HH-MM-SS)

~~4.01.00~~**4.2.1.10 HW**

No.	ParID	Data type	Description	En
1	451	T_FLAG1	non fiscal character: 0 - @, 1 - », 2 - #, 3 - \$, 4 - no character	0
2	452	T_FLAG1	font of the receipt: 0 - <i>always standard</i> , 1 - <i>always sedimentary (in existing HW settings)</i> ,	0
3	453	T_FLAG1	font of the journal: 0 - <i>always standard</i> , 1 - <i>always sedimentary (in existing HW settings)</i> , (flag is ignored if flag 9 = 1, 2 a 3 – just standard font)	0
4	454	T_FLAG1	the height of the printed line on receipt: 0 - <i>always standard</i> , 1 - <i>default</i>	1
5	455	T_FLAG1	the height of the printed line on journal: 0 - <i>always standard</i> , 1 - <i>default</i>	0
6	456	T_FLAG1	not in use	0
7	457	T_FLAG1	not in use	0
8	458	T_FLAG1	not in use	0
9	459	T_FLAG1	number of characters per line: – read-only 0 - 29 (54 * 54), 1 - 33 (70 * 38), 2 - 29 (54 * 38), 3 - 24 (38 * 38)	0
10	460	T_FLAG1	customer display (internal) type: – read-only 1 - <i>alphanumeric</i> , 2 - <i>segment</i>	1
11	461	T_FLAG1	customer display type: – read-only 0 - <i>none</i> 1 - <i>alphanumeric</i> , 2 - <i>segment</i>	0
12	462	T_FLAG1	cutter: – read-only 0 - <i>no</i> , 1 - <i>yes</i>	0

4.2.1.11 Reports

No.	ParID	Data type	Description	En
1	501	T_FLAG1	printing of interrupted report: 0 - <i>from the beginning</i> , 1 - <i>continuous</i>	1
2	502	T_FLAG1	report printing: 0 - <i>all Z reports are printing on both sides</i> , 1 - <i>only registration Z reports are printing on both sides</i> , 2 - <i>according to user control</i>	1
3	503	T_FLAG1	Number of decimal places for financial values printed in financial reports for the master currency: (0 – 4)	4

~~4.01.00~~**4.2.1.12 Limits**

No.	ParID	Data type	Description	En
1	551	T_QUANTITY	the limit for total quantity for one PLU: (0 – 1e8)	1e8
2	552	T_PRICE	the limit for unit price of PLU: (0 – 1e12)	1e12
3	553	T_PRICE	the limit for the total price of PLU: (0 – 1e12)	1e12
4	554	T_PRICE	the limit for the total sale: (0 – 1e13)	1e13
5	555	T_PRICE	the limit for the daily total ^D : (0 – 1e13)	1e13
6	556	T_PRICE	the limit for the GT ^D : (0 – 4.5e15)	4.5e15

Note: – each value is fixed to 3 decimal places

– condition: flag #2 <= flag #3 <= flag #4 <= flag #5 <= flag #6

4.2.1.13 Format

No.	ParID	Data type	Description	En
1	601	T_FLAG1	print tax info on receipt: 0 – <i>never</i> , 1 – <i>always</i> , 2 – <i>tax info printing is controlled by pressing of the dedicated key</i>	1
2	602	T_FLAG1	not in use	0
3	603	T_FLAG1	decimal separator type ^D : 0 – “.”, 1 – “,”	0
4	604	T_FLAG1	print thousands separator ^D : 0 – <i>no</i> , 1 – <i>yes</i> , if 1 – <i>then is different from decimal separator</i>	0
5	605	T_FLAG1	display thousands separator: 0 – <i>no</i> , 1 – <i>yes</i> , if 1 – <i>then is different from decimal separator</i>	1
6	606	T_FLAG1	print time and date on receipt: 0 – <i>on the top</i> , 1 – <i>at the bottom</i> , 2 – <i>both</i>	0
7	607	T_FLAG1	print time on receipt: 0 – <i>never</i> , 1 – <i>always</i>	1
8	608	T_FLAG1	date format ^D : 0 – DDMMYY, 1 – MMDDYY, 2 – YYMMDD, 3 – YYDDMM	1
9	609	T_FLAG1	tax assignment format ^M : 0 – XX%, 1 – <i>Roman numeral</i> , 2 – <i>letters A, B ...</i> , 3 – <i>nothing</i>	1
10	610	T_FLAG1	print unit price for unit value: 0 – <i>no</i> , 1 – <i>yes</i>	0
11	611	T_FLAG1	receipt printing after interruption: <i>feature is not implemented</i>	0
12	612	T_FLAG1	print DUPLICATE title, when case 1 st receipt printing is off: 0 – <i>no</i> , 1 – <i>yes</i>	1
13	613	T_FLAG1	PLU printing (to change a flag it is allowed in retail version only): 0 – <i>at the end of receipt</i> , 1 – <i>immediately</i>	0
14	614	T_FLAG1	print PLU number on receipt: 0 – <i>no</i> , 1 – <i>yes</i>	0

4-01.00

15	615	T_FLAG1	print sign “=” before values: <i>0 – no,</i> <i>1 – yes</i>	1
16	616	T_FLAG1	format of tax info at the end of receipt: <i>0 – by lines,</i> <i>1 – in columns</i>	0
17	617	T_FLAG1	print taxable values with tax levels as tax report on receipt: <i>0 – no,</i> <i>1 – yes</i>	1
18	618	T_FLAG1	print net values with tax levels as tax report on receipt: <i>0 – no,</i> <i>1 – yes</i>	1
19	619	T_FLAG1	print tax values with tax levels as tax report on receipt: <i>0 – no,</i> <i>1 – yes</i>	1
20	620	T_FLAG1	print course on receipt: <i>0 – no,</i> <i>1 – yes</i>	1

4.2.1.14 Top graphical logo

No.	ParID	Data type	Description	En
1	651	T_FLAG1	number of micro lines: <i>(0 – 240)</i>	240
2	652	T_RAWSTRING[1350]	1 st – 30 th micro line	
3	653	T_RAWSTRING[1350]	31 st – 60 th micro line	
4	654	T_RAWSTRING[1350]	61 st – 90 th micro line	
5	655	T_RAWSTRING[1350]	91 st – 120 th micro line	
6	656	T_RAWSTRING[1350]	121 st – 150 th micro line	
7	657	T_RAWSTRING[1350]	151 st – 180 th micro line	
8	658	T_RAWSTRING[1350]	181 st – 210 th micro line	
9	659	T_RAWSTRING[1350]	211 th – 240 th micro line	

Note: – one micro line contains 360 points, there are 8 points stored in 1 B => micro line is 360 / 8 = 45 B long
– 1 – 30 micro lines can be programmed at the same time => T_RAWSTRING value size is 45 – 1350 B

4.2.1.15 Bottom graphical logo

No.	ParID	Data type	Description	En
1	701	T_FLAG1	number of micro lines: <i>(0 – 240)</i>	100
2	702	T_RAWSTRING[1350]	1 st – 30 th micro line	
3	703	T_RAWSTRING[1350]	31 st – 60 th micro line	
4	704	T_RAWSTRING[1350]	61 st – 90 th micro line	
5	705	T_RAWSTRING[1350]	91 st – 120 th micro line	
6	706	T_RAWSTRING[1350]	121 st – 150 th micro line	
7	707	T_RAWSTRING[1350]	151 st – 180 th micro line	
8	708	T_RAWSTRING[1350]	181 st – 210 th micro line	
9	709	T_RAWSTRING[1350]	211 th – 240 th micro line	

4.2.2 Service and program mode parameters

4.2.2.1 PC

No.	ParID	Data type	Description	En
1	850	T_FLAG1	device state: <i>0 – inactive,</i> <i>1 – active</i>	1
2	851	T_FLAG1	device type	0
3	852	T_FLAG2	communication speed (Bd): <i>9,600,</i> <i>19,200,</i> <i>38,400</i>	38400
4	853	T_FLAG1	parity: <i>0 – none,</i> <i>1 – even,</i> <i>2 – odd</i>	0
5	854	T_FLAG1	stop bits: <i>(1 – 2)</i>	1
6	855	T_FLAG1	number of data bit: <i>(7 – 8)</i>	8

4.2.2.2 Scanner

No.	ParID	Data type	Description	En
1	857	T_FLAG1	device state: <i>0 – inactive,</i> <i>1 – active</i>	1
2	858	T_FLAG1	device type	0
3	859	T_FLAG2	communication speed (Bd): <i>2,400,</i> <i>4,800,</i> <i>9,600,</i> <i>19,200,</i> <i>38,400</i>	9600
4	860	T_FLAG1	parity: <i>0 – none,</i> <i>1 – even,</i> <i>2 – odd</i>	0
5	861	T_FLAG1	stop bits: <i>(1 – 2)</i>	1
6	862	T_FLAG1	number of data bit: <i>(7 – 8)</i>	8

4.2.2.3 Payment terminal

No.	ParID	Data type	Description	En
1	864	T_FLAG1	device state: 0 – inactive, 1 – active	0
2	865	T_FLAG1	device type (<i>Bull protocol</i>)	0
3	866	T_FLAG2	communication speed (Bd): 2,400, 4,800, 9,600, 19,200, 38,400	9600
4	867	T_FLAG1	parity: 0 – none, 1 – even, 2 – odd	0
5	868	T_FLAG1	stop bits: (1 – 2)	1
6	869	T_FLAG1	number of data bit: (7 – 8)	8

4.2.2.4 Scales 1

No.	ParID	Data type	Description	En
1	871	T_FLAG1	device state: 0 – inactive, 1 – active	0
2	872	T_FLAG1	device type: 1 – MORCAN, 2 – MARTES, 3 – MACCA_K 4 – none	4
3	873	T_FLAG2	communication speed (Bd): 2,400, 4,800, 9,600, 19,200, 38,400	9600
4	874	T_FLAG1	parity: 0 – none, 1 – even, 2 – odd	0
5	875	T_FLAG1	stop bits: (1 – 2)	1
6	876	T_FLAG1	number of data bit: (7 – 8)	8

~~4.01.00~~**4.2.2.5 Scales 2**

No.	ParID	Data type	Description	En
1	878	T_FLAG1	device state: 0 – <i>inactive</i> , 1 – <i>active</i>	0
2	879	T_FLAG1	device type: 1 – <i>MORCAN</i> , 2 – <i>MARTES</i> , 3 – <i>MACCA_K</i> 4 – <i>none</i>	4
3	880	T_FLAG2	communication speed (Bd): 2,400, 4,800, 9,600, 19,200, 38,400	9600
4	881	T_FLAG1	parity: 0 – <i>none</i> , 1 – <i>even</i> , 2 – <i>odd</i>	0
5	882	T_FLAG1	stop bits: (1 – 2)	1
6	883	T_FLAG1	number of data bit: (7 – 8)	8

4.2.2.6 ECM

No.	ParID	Data type	Description	En
1	885	T_FLAG1	device state: 0 – <i>inactive</i> , 1 – <i>active</i>	0
2	886	T_FLAG1	device type:	0
3	887	T_FLAG2	communication speed (Bd): 9,600, 38,400	38400
4	888	T_FLAG1	parity: 0 – <i>none</i> , 1 – <i>even</i> , 2 – <i>odd</i>	0
5	889	T_FLAG1	stop bits: (1 – 2)	1
6	890	T_FLAG1	number of data bit: (7 – 8)	8

4.2.2.7 External printers 1 – 7

Printer no.1:

No.	ParID	Data type	Description	En
1	950	T_STRING[10]	printer name	
2	951	T_FLAG1	device state: 0 – inactive, 1 – active	0
3	952	T_FLAG1	communication port: 1 – local 1, 2 – local 2, 3 – external 1 (ECM), 4 – external 2 (ECM), 5 – external 3 (ECM), 6 – external 4 (ECM), 7 – external 5 (ECM), 8 – external 6 (ECM), 9 – external 7 (ECM)	1
4	953	T_FLAG1	device type: 1 – iDP_3421RF – CITIZEN, 2 – TM_U210B_021 – EPSON, 3 – TM_T88II_011 – EPSON, 4 – E_3202 – DATAMAX, 5 – TM_U295_011 – EPSON, 6 – TM_U675_011 – EPSON, 7 – TM_H5000II_011 – EPSON	1
5	954	T_FLAG1	SLIP: 0 – no, 1 – yes	0
6	955	T_FLAG1	feeding between bills: (0 – 255)	4
7	956	T_FLAG1	feeding between rows: (0 – 255)	3
8	957	T_FLAG2	communication speed (Bd): 2,400, 4,800, 9,600, 19,200, 38,400	9600
9	958	T_FLAG1	parity: 0 – none, 1 – even, 2 – odd	0
10	959	T_FLAG1	stop bits: (1 – 2)	1
11	960	T_FLAG1	number of data bit: (7 – 8)	8
12	961	T_RAWSTRING[10]	terminative sequence	

	963 – 974	external printer #2
	976 – 987	external printer #3
	989 – 1000	external printer #4
	1002 – 1013	external printer #5
	1015 – 1026	external printer #6
	1028 – 1039	external printer #7

4.2.2.8 IN-LINE

No.	ParID	Data type	Description	En
1	1100	T_FLAG1	device state: <i>0 – inactive,</i> <i>1 – active</i>	1
2	1101	T_STRING[20]	slave device name	ECR0
3	1102	T_FLAG1	link address: (2 – 250)	2

4.2.2.9 Memory ranges

No.	ParID	Data type	Description	En
1	1250	T_FLAG1	PLU set ID	0
2	1251	T_FLAG2	PLU set range	1000
3	1252	T_FLAG1	DPT set ID	1
4	1253	T_FLAG2	DPT set range	16
5	1254	T_FLAG1	Surcharge/discount set ID	2
6	1255	T_FLAG2	Surcharge/discount set range	7
7	1256	T_FLAG1	Payment set ID	3
8	1257	T_FLAG2	Payment set range	12
9	1258	T_FLAG1	Cashier set ID	4
10	1259	T_FLAG2	Cashier set range	6
1	1260	T_FLAG1	Table set ID	5
12	1261	T_FLAG2	Table set range	99
13	1262	T_FLAG1	Customer set ID	6
14	1263	T_FLAG2	Customer set range	20
15	1264	T_FLAG1	Bill items set ID	7
16	1265	T_FLAG2	Bill items set range	500
17	1266	T_FLAG1	Bill set ID	8
18	1267	T_FLAG2	Bill set range	100
19	1268	T_FLAG1	DPT group set ID	9
20	1269	T_FLAG2	DPT group set range	8
21	1270	T_FLAG1	Link PLU set ID	10
22	1271	T_FLAG2	Link PLU set range	50
23	1272	T_FLAG1	Promotion set ID	11
24	1273	T_FLAG2	Promotion set range	0
25	1274	T_FLAG1	Unit set ID	12
26	1275	T_FLAG2	Unit set range	8
27	1276	T_FLAG1	Descriptive text set ID	13
28	1277	T_FLAG2	Descriptive text set range	20
29	1278	T_FLAG1	Link text set ID	14
30	1279	T_FLAG2	Link text set range	50
31	1280	T_FLAG1	Direct key set ID	15
32	1281	T_FLAG2	Direct key set range	160
33	1282	T_FLAG1	Additive text set ID	16
34	1283	T_FLAG2	Additive text set range	10
35	1284	T_FLAG1	Customer text set ID	17
36	1285	T_FLAG2	Customer text set range	10
37	1286	T_FLAG1	Promo set ID	18
38	1287	T_FLAG2	Promo set range	0

4.2.2.10 Program mode parameters

No.	ParID	Data type	Description	En
-----	-------	-----------	-------------	----

4.01.00

1	1400	T_FLAG1	entering document number at the beginning of a bill is mandatory: <i>0 – no,</i> <i>1 – yes</i>	0
2	1401	T_FLAG1	mandatory entering of customer number at the beginning of a bill: <i>0 – no,</i> <i>1 – yes</i>	0
3	1402	T_FLAG1	realise RA (in master currency) after daily report: <i>0 – no,</i> <i>1 – yes</i>	0
4	1403	T_FLAG1	realise PO (in master currency) before daily report: <i>0 – no,</i> <i>1 – yes</i>	0
5	1404	T_FLAG1	mandatory pressing of <i>SUBTOTAL</i> key before sale finish <i>0 – no,</i> <i>1 – yes</i>	0
6	1405	T_FLAG1	print subtotal after press <i>SUBTOTAL</i> key <i>0 – no,</i> <i>1 – yes</i>	0
7	1406	T_FLAG1	allow PLU multiple quantity sale: <i>0 – no,</i> <i>1 – yes</i>	1
8	1407	T_FLAG1	allow PLU fragmentary quantity sale: <i>0 – no,</i> <i>1 – yes</i>	1
9	1408	T_FLAG1	print unit price to external printer: <i>0 – no,</i> <i>1 – yes</i>	0
10	1409	T_STRING[20]	ECR name printed to external printers	
11	1410	FLAG1	display text or time on display if inactivity: <i>0 – text</i> <i>1 – time</i>	0
12	1411	FLAG1	idle time: <i>0 – inactive,</i> <i>1 – 255 – time in seconds</i>	20

4.2.2.11 Service mode parameters

No.	ParID	Data type	Description	En
1	1450	T_FLAG1	application mode: – read-only 0 – Basic Shopping 1 – Hold System 2 – Overlapping 3 – Restaurant	0
2	1451	T_FLAG1	allow date setting in program mode: 0 – no, 1 – yes	1
3	1452	T_FLAG1	allow time setting in program mode: 0 – no, 1 – yes	1
4	1453	T_FLAG1	allow changing the upper text logo parameters in program mode: 0 – no, 1 – yes	1
5	1454	T_FLAG1	allow changing the bottom text logo parameters in program mode: 0 – no, 1 – yes	1
6	1455	T_FLAG1	open drawer mode: 0 – cannot open 1 – open after press OPEN_DRAWER key 2 – open after press OPEN_DRAWER key with info printing	1
7	1456	T_FLAG1	enter price mode: 0 – is not necessary to enter decimal point (entered number includes decimal places) 1 – is necessary to enter decimal point	0
8	1457	T_FLAG1	register mode printing: 0 – both sides, 1 – receipt side only, 2 – journal side only, 3 – none	0
9	1458	T_FLAG1	training mode printing: 0 – both sides, 1 – receipt side only, 2 – journal side only, 3 – none	1
10	1459	T_FLAG1	allow switch off printing in register mode: 0 – both sides, 1 – receipt side only, 2 – journal side only, 3 – none	0
11	1460	T_FLAG1	allow switch off printing in training mode: 0 – both sides, 1 – receipt side only, 2 – journal side only, 3 – none	0
12	1461	T_FLAG1	allow print bill duplicate: 0 – no, 1 – both sides, 2 – receipt side only, 3 – user control	2
13	1462	T_FLAG1	service and program mode printing: 0 – both sides, 1 – receipt side only	1
14	1463	T_FLAG1	X and Z mode printing: 0 – both sides, 1 – receipt side only	1

4-01-00

15	1464	T_FLAG1	PO / RA realisation info: <i>0 – no description of operation, 1 – necessary to enter one descriptive row, 2 – necessary to enter two descriptive rows, 3 – necessary to enter three descriptive rows, 4 – necessary to enter four descriptive rows</i>	1
16	1465	T_FLAG1	price levels switching: <i>0 – without return, 1 – immediate return, 2 – return if sale is finished by payment, cancelled or bill is suspended, 3 – return if other cashier logged in</i>	1
17	1466	T_FLAG1	refund type of credit payment: (pay. terminal): <i>0 – if negative sale and payment through credit, will be financial operation realised (subtract) from cash 1 – if negative sale and payment through credit, will be financial operation realised (subtract) from credit (also through pay. terminal)</i>	0
18	1467	T_FLAG1	PLU Surcharge / Discount application: <i>0 – only to Master PLU 1 – to Master PLU and linked PLUs</i>	0
19	1468	T_FLAG1	bill items printing after bill ejection: <i>0 – no, 1 – summary only – new subtotal, 2 – new bill items, 3 – new bill items and summary</i>	3
20	1469	T_FLAG2	direct key step: (0 – 560)	52
21	1470	T_FLAG1	key press beep: <i>0 – off, 1 – on</i>	1
22	1471	T_FLAG2	bill number, used by accounting without opening a bill: (10 – 1000), <i>10 means bill no. 1.0, 1000 – accounting without opening a bill is not active</i>	990
23	1472	T_FLAG1	allow cashier report if there are some bills open: <i>0 – no, 1 – yes</i>	0
24	1473	T_FLAG1	allow print invoice: <i>0 – no, 1 – both sides, 2 – receipt side only, 3 – user control</i>	2
25	1474	T_FLAF1	Auto log off after suspending bill, finishing receipt with payment, invocation of invoice or of duplicate: <i>0 – no, 1 – yes</i>	0
26	1475	T_FLAF1	font height settings for the receipt printed on the external printers <i>0 – normal font, 1 – double height</i>	1

4.2.2.12 Electronic journal parameters

No.	ParID	Data type	Description	En
1	1500	T_FLAG1	text journal line length	29
2	1501	T_PERCENT	journal memory occupancy state: (1 – 10 000) (1% = 100)	8000
3	1502	T_FLAG1	store binary journal: 0 – no, 1 – yes	0
4	1503	T_FLAG1	store text journal: 0 – no, 1 – yes	0
5	1504	T_FLAG1	store training mode in EJ: 0 – no, 1 – yes	0
6	1505	T_FLAG1	store receipt in EJ: 0 – no, 1 – yes	0
7	1506	T_FLAG1	store RA in EJ: 0 – no, 1 – yes	0
8	1507	T_FLAG1	store PO in EJ: 0 – no, 1 – yes	0
9	1508	T_FLAG1	store daily Z report in EJ: 0 – no, 1 – yes	0
10	1509	T_FLAG1	store monthly Z report in EJ: 0 – no, 1 – yes	0
11	1510	T_FLAG1	store annual Z report in EJ: 0 – no, 1 – yes	0
12	1511	T_FLAG1	store bill items in EJ: 0 – no, 1 – yes	0
13	1512	T_FLAG1	store payments info in EJ: 0 – no, 1 – yes	0
14	1513	T_FLAG1	store taxes info in EJ: 0 – no, 1 – yes	0
15	1514	T_FLAG1	store customers info in EJ: 0 – no, 1 – yes	0
16	1515	T_FLAG1	store financial records in EJ: 0 – no, 1 – yes	0
17	1516	T_FLAG1	store bill item type in EJ: 0 – no, 1 – yes	0
18	1517	T_FLAG1	store tax ID in EJ: 0 – no, 1 – yes	0
19	1518	T_FLAG1	store department ID in EJ: 0 – no, 1 – yes	0
20	1519	T_FLAG1	store PLU ID in EJ: 0 – no, 1 – yes	0
21	1520	T_FLAG1	store programmed PLU price in EJ: 0 – no, 1 – yes	0

4-01-00

22	1521	T_FLAG1	store total quantity in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
23	1522	T_FLAG1	store price in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
24	1523	T_FLAG1	store total price in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
25	1524	T_FLAG1	store surcharge / discount in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
26	1525	T_FLAG1	store subtotal surcharge / discount in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
27	1526	T_FLAG1	store promo discount in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
28	1527	T_FLAG1	store tax rate in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
29	1528	T_FLAG1	store payment ID in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
30	1529	T_FLAG1	store currency type in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
31	1530	T_FLAG1	store payment value in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
32	1531	T_FLAG1	store descriptive text 1 in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
33	1532	T_FLAG1	store descriptive text 2 in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
34	1533	T_FLAG1	store tax type in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
35	1534	T_FLAG1	store taxable in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
36	1535	T_FLAG1	store net in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
37	1536	T_FLAG1	store tax in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
38	1537	T_FLAG1	store tax rate in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
39	1538	T_FLAG1	store customer credit value in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
40	1539	T_FLAG1	store customer credit limit in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
41	1540	T_FLAG1	store customer discount in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
42	1541	T_FLAG1	store customer discount type in EJ: <i>0 – no,</i> <i>1 – yes</i>	0

4-01-00

43	1542	T_FLAG1	store difference in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
44	1543	T_FLAG1	store void quantity in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
45	1544	T_FLAG1	store void value in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
46	1545	T_FLAG1	store refund quantity in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
47	1546	T_FLAG1	store refund value in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
48	1547	T_FLAG1	store surcharge quantity in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
49	1548	T_FLAG1	store surcharge value in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
50	1549	T_FLAG1	store discount quantity in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
51	1550	T_FLAG1	store discount value in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
52	1551	T_FLAG1	store pack sold quantity in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
53	1552	T_FLAG1	store pack sold value in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
54	1553	T_FLAG1	store pack returned quantity in EJ: <i>0 – no,</i> <i>1 – yes</i>	0
55	1554	T_FLAG1	store pack returned value in EJ: <i>0 – no,</i> <i>1 – yes</i>	0

Note: Journals are in fiscal countries not implemented

4.2.2.13 LOG parameters

No.	ParID	Data type	Description	En
1	1600	T_FLAG1	store LOG info: <i>0 – no, 1 – print to journal receipt, 2 – store in text journal</i>	0
2	1601	T_FLAG1	store cashier operations: <i>0 – no, 1 – yes</i>	0
3	1602	T_FLAG1	store bill operations: <i>0 – no, 1 – yes</i>	0
4	1603	T_FLAG1	store other operations: <i>0 – no, 1 – yes</i>	0
5	1604	T_FLAG1	store mode change operations: <i>0 – no, 1 – yes</i>	0
6	1605	T_FLAG1	store bill assign operations: <i>0 – no, 1 – yes</i>	0
7	1606	T_FLAG1	store bill open operations: <i>0 – no, 1 – yes</i>	0
8	1607	T_FLAG1	store bill close operations: <i>0 – no, 1 – yes</i>	0
9	1608	T_FLAG1	store RA/PO operations: <i>0 – no, 1 – yes</i>	0
10	1609	T_FLAG1	store normal bill operations: <i>0 – no, 1 – yes</i>	0
11	1610	T_FLAG1	store inverse bill operations: <i>0 – no, 1 – yes</i>	0
12	1611	T_FLAG1	store cross-bill operations: <i>0 – no, 1 – yes</i>	0
13	1612	T_FLAG1	store void and VIP operations:: <i>0 – no, 1 – yes</i>	0
14	1613	T_FLAG1	store payment operations: <i>0 – no, 1 – yes</i>	0
15	1614	T_FLAG1	store P mode access operations: <i>0 – no, 1 – yes</i>	0
16	1615	T_FLAG1	store service mode access operations: <i>0 – no, 1 – yes</i>	0
17	1616	T_FLAG1	store record change operations: <i>0 – no, 1 – yes</i>	0
18	1617	T_FLAG1	store change service settings operations: <i>0 – no, 1 – yes</i>	0
19	1618	T_FLAG1	store Z report executing operations: <i>0 – no, 1 – yes</i>	0
20	1619	T_FLAG1	store communication operations: <i>0 – no, 1 – yes</i>	0

Note: Logs are in fiscal countries not implemented

4.2.2.14 Application texts

No.	ParID	Data type	Description	En
1	1650	T_STRING[11]	Bill	Bill
2	1651	T_STRING[16]	New bill	New bill
3	1652	T_STRING[20]	Enter bill number	Enter bill number
4	1653	T_STRING[16]	Suspended bills	Suspended bills
5	1654	T_STRING[16]	Put off bill	Put off bill
6	1655	T_STRING[16]	Cancel bill	Cancel bill
7	1656	T_STRING[16]	Give over bill	Give over bill
8	1657	T_STRING[16]	Document number	Document number
9	1658	T_STRING[20]	Enter document num.	Enter document num.
10	1659	T_STRING[16]	Doc.Number	Doc.Number
11	1660	T_STRING[24]	New subtotal	New subtotal
12	1661	T_STRING[24]	Old subtotal	Old subtotal
13	1662	T_STRING[20]	Welcome	WELCOME
14	1663	T_STRING[100]	I am euro 2500tx–the best cash register. have a nice day.	I AM EURO 2500TX–THE BEST CASH REGISTER. HAVE A NICE DAY.
15	1664	T_STRING[15]	Table number	Table number
16	1665	T_STRING[14]	Tables	Tables

4.2.2.15 Customer texts description

No.	ParID	Data type	Description	En
1	1700	T_STRING[20]	customer's name	Name
2	1701	T_STRING[20]	customer's name – line 2	
3	1702	T_STRING[20]	Vat number	Vat number
4	1703	T_STRING[20]	Address	Address
5	1704	T_STRING[20]	Address – line 2	
6	1705	T_STRING[20]	Address – line 3	
7	1706	T_STRING[20]	Demand for entering customer's name description on invoice	Enter name
8	1707	T_STRING[20]	Demand for entering customer's 2 nd line description on invoice	Enter name-line2
9	1708	T_STRING[20]	Demand for entering customer's 3 rd line description on invoice	Enter VAT number
10	1709	T_STRING[20]	Demand for entering customer's 4 th line description on invoice	Enter address
11	1710	T_STRING[20]	Demand for entering customer's 5 th line description on invoice	Enter address-line2
12	1711	T_STRING[20]	Demand for entering customer's 6 th line description on invoice	Enter address-line3

Note: – first 6 texts are printed on the receipt with customer data or on the invoice as the description to the customer's text, second 6 texts are demands to enter customer's texts for the invoice if these texts are not programmed in the database and they cannot be empty.

4.2.2.16 Memory dump

No.	ParID	Data type	Description
1	2048	T_RAWSTRING [1024]	1 st 1024 bytes-long memory block
2	2049	T_RAWSTRING [1024]	2 nd 1024 bytes-long memory block
3	2050	T_RAWSTRING [1024]	3 rd 1024 bytes-long memory block
...
2046	4093	T_RAWSTRING [1024]	2046 th 1024 bytes-long memory block
2047	4094	T_RAWSTRING [1024]	2047 th 1024 bytes-long memory block
2048	4095	T_RAWSTRING [1024]	2048 th 1024 bytes-long memory block

4.2.3 Reading of ECR parameters

txt2500.exe <portNum> /p 1

or

txt2500.exe <portNum> /p 1 0 0 <dataFileName> <headerFileName>

Example:

- The command reads the system parameters listed in *head.in* file from ECR connected to COM1 port:

txt2500.exe 1 /p 1

- Example of *head.in* file for reading upper text logo:

```
251:6;33      //1. logo line
252:6;33      //2. logo line
253:6;33      //3. logo line
254:6;33      //4. logo line
255:6;33      //5. logo line
256:6;33      //6. logo line
```

- Example of *data.in* file:

```
1. logo line
2. logo line
3. logo line
4. logo line
5. logo line
6. logo line
```

- Example of *head.in* file for reading the upper graphical logo:

```
652:7;1350     //1. – 30. microline
653:7;1350     //31. – 60. microline
654:7;1350     //61. – 90. microline
655:7;1350     //91. – 120. microline
656:7;1350     //121. – 150. microline
657:7;1350     //151. – 180. microline
658:7;1350     //181. – 210. microline
659:7;1350     //211. – 240. microline
```

- The read graphical logo will be stored to *glogo1.bmp* file (or *glogo2.bmp* file for bottom logo)

4.2.4 Programming of ECR parameters

txt2500.exe <portNum> /p 0

or

txt2500.exe <portNum> /p 0 0 0 <dataFileName> <headerFileName>

Example:

- The command programs the system parameters (listed in *head.out* file) to ECR connected at the COM1 port:

txt2500.exe 1 /p 0

- Example of *data.out* file with structure equal to *head.out* file structure:

1. *logo line*
2. *logo line*
3. *logo line*
4. *logo line*
5. *logo line*
6. *logo line*

- Example of *head.out* file for the upper text logo programming:

251:6;33	//1. <i>logo line</i>
252:6;33	//2. <i>logo line</i>
253:6;33	//3. <i>logo line</i>
254:6;33	//4. <i>logo line</i>
255:6;33	//5. <i>logo line</i>
256:6;33	//6. <i>logo line</i>

4.3 Functions

Function types:

FuncID	Description
0	modify PLU stock
1	execute PLU report
2	execute DPT report
3	execute SURCHARGE / DISCOUNT report
4	execute PAYMENT report
5	execute CASHIER report
6	execute TABLE report
7	execute CUSTOMER report
8	execute HOUR report
9	execute PLU stock report
10	<i>reserved</i>
11	execute FINANCIAL report
12	read FINANCIAL report
20	JOURNAL – read
23	JOURNAL – delete
30	Block ECR
31	Unblock ECR
40	Set master currency *
41	Set TAX levels *
42	Reregister ECR *
50	Read settings for Get FM *
51	Get next FM *

*implemented only in fiscal countries

Function executing:

txt2500.exe <portNum> If[funcID]

or

txt2500.exe <portNum> If[funcID] 0 0 0 <dataFileName> <headerFileName>

4.3.1 Stock modification

4.3.1.1 PLU stock modification

Input parameters

MemID	Data type	Description
0	T_FLAG4	PLU ID: (1 – 999 999)
1	T_FLAG1	stock operation: 0 – <i>overwrite</i> , 1 – <i>add</i> , 2 – <i>subtract</i>
2	T_QUANTITY	stock value

Output parameters

MemID	Data type	Description
0	T_FLAG2	return code
1	T_FLAG4	PLU ID
2	T_QUANTITY	new stock value

Example:

- The command executes the function No.0 (PLU stock modification) at the ECR connected to COM1 port:

```
txt2500.exe 1 /f0
```

- Example of *head.out* file:

```
0:5;4 //PLU ID
1:3;1 //operation
2:1;8 //stock value
```

- Example of *data.out* file:

```
1;0;20,0000 //change stock
2;1;10,0000 //add stock
3;2;10,0000 //sub stock
:
```

- file *data.in* contains the return code of the operation, ID of PLU and the value of the new stock for the each item.

- Example of *data.in* file:

```
0;1;20,0000 //change stock
0;2;110,0000 //add stock
0;3;2;90,0000 //sub stock
:
```

4.3.2 Report execution

4.3.2.1 Execute PLU report

Input parameters

MemID	Data type	Description
0	T_FLAG1	report type: 0 – <i>daily</i> , 1 – <i>monthly</i> , 2 – <i>yearly</i>
1	T_FLAG1	print report: 0 – <i>no</i> , 1 – <i>yes</i>
2	T_FLAG1	report: 0 – <i>Z</i> , 1 – <i>X</i>
3	T_FLAG1	<i>reserved</i>
4	T_FLAG4	first record ID
5	T_FLAG4	last record ID

Output parameters

MemID	Data type	Description
0	T_FLAG2	return code

Note: All report executing functions have the same structure of parameters

Example:

- The command executes the function No.1 (PLU report) at the ECR connected to COM1 port:

```
txt2500.exe 1 /f1
```

- Example of *head.out* file for financial report execution:

```
0:3;1 //report type
1:3;1 //print report
2:3;1 //erase report
3:3;1 //reserved
4:5;4 //low index
5:5;4 //high index
```

- Example of input data file:

```
0;1;1;0;1;5
```

- The file *data.in* contains the return code of the operation
- Example of *data.in* file:

```
0;
```

4.3.2.2 Execute FINANCIAL report

Input parameters

MemID	Data type	Description
0	T_FLAG1	report type: 0 – daily, 1 – monthly, 2 – yearly
1	T_FLAG1	report printing*: 0 – to both paper tapes 1 – to receipt tape only 2 – do not print the report
2	T_FLAG1	report: 0 – Z, 1 – X
3	T_FLAG1	mode: 0 – R, 1 – T
4	T_FLAG4	<i>reserved</i>
5	T_FLAG4	<i>reserved</i>

Output parameters

MemID	Data type	Description
0	T_FLAG2	return code

* In case of a Z report, this parameter can be overridden by the system flag 11.2 (see chapter 3.2.1.11) so, that the report will be printed on both paper tapes despite the value of this flag (see relevant flag values for both flags)

Example:

- The command executes the function No.11 (financial report) at the ECR connected to COM1 port:

```
txt2500.exe 1 /f11
```

- Example of *head.out* file for financial report execution:

```
0:3;1 //report type
1:3;1 //print report
2:3;1 //erase report
3:3;1 //mode
4:5;4 //reserved
5:5;4 //reserved
```

- Example of input data file:

```
0;0;0;0;0;0
```

- The file *data.in* contains the return code of the operation
- Example of *data.in* file:

```
0;
```


~~4.01.00~~

4.3.3 Report reading

4.3.3.1 Financial report reading

Input parameters

MemID	Data type	Description
0	T_FLAG1	report type: 0 – daily, 1 – monthly, 2 – yearly
1	T_FLAG1	mode: 0 – registration, 1 – training

Output parameters

MemID	Data type	Description
0	T_FLAG2	return code
1	T_PRICE	GT1
2	T_PRICE	GT2
3	T_PRICE	GT3
4	T_FLAG2	number of daily reports
5	T_FLAG2	number of monthly reports
6	T_FLAG2	number of MR (cannot be erased)
7	T_FLAG4	number of fiscal receipts
8	T_FLAG4	number of all receipts
9	T_PRICE	TAXABLE 1
10	T_PRICE	TAXABLE 2
11	T_PRICE	TAXABLE 3
12	T_PRICE	TAXABLE 4
13	T_PRICE	TAXABLE 5
14	T_PRICE	TAXABLE 6
15	T_PRICE	TAXABLE 7
16	T_PRICE	NETTO 1
17	T_PRICE	NETTO 2
18	T_PRICE	NETTO 3
19	T_PRICE	NETTO 4
20	T_PRICE	NETTO 5
21	T_PRICE	NETTO 6
22	T_PRICE	NETTO 7
23	T_PRICE	VAT 1
24	T_PRICE	VAT 2
25	T_PRICE	VAT 3
26	T_PRICE	VAT 4
27	T_PRICE	VAT 5
28	T_PRICE	VAT 6
29	T_PRICE	VAT 7
30	T_PRICE	Total SUM – accepted value
31	T_PRICE	Difference (Total SUM – Total TAXABLE 1 to 7)
32	T_QUANTITY	number of done or finished sales in individual ECR modes
33	T_QUANTITY	number of corrections – absolute value
34	T_PRICE	value of corrections – absolute value
35	T_QUANTITY	number of refunds
36	T_PRICE	value of refunds
37	T_QUANTITY	number of surcharges

~~4-01-00~~

38	T_PRICE	value of surcharges
39	T_QUANTITY	number of discounts
40	T_PRICE	value of discounts
41	T_QUANTITY	number of sold returnable packs
42	T_PRICE	value of sold returnable packs
43	T_QUANTITY	number of bought-out returnable packs
44	T_PRICE	value of bought-out returnable packs
45	T_PRICE	value of refund receipts – absolute value
46	T_QUANTITY	number of refund receipts
47	T_PRICE	value of refund receipts
48	T_QUANTITY	number of cancelled receipts – absolute value
49	T_PRICE	value of cancelled receipts – absolute value
50	T_QUANTITY	number of cash payment in master currency
51	T_PRICE	value of cash payment in master currency
52	T_QUANTITY	number of cash payment in secondary currency
53	T_PRICE	value of cash payment in secondary currency
54	T_QUANTITY	number of check payment in master currency
55	T_PRICE	value of check payment in master currency
56	T_QUANTITY	number of check payment in secondary currency
57	T_PRICE	value of check payment in secondary currency
58	T_QUANTITY	number of credit payment in master currency
59	T_PRICE	value of credit payment in master currency
60	T_QUANTITY	number of credit payment in secondary currency
61	T_PRICE	value of credit payment in secondary currency
62	T_QUANTITY	number of customer credit payment in master currency
63	T_PRICE	value of customer credit payment in master currency
64	T_QUANTITY	number of customer credit payment in secondary currency
65	T_PRICE	value of customer credit payment in secondary currency
66	T_QUANTITY	number of value in other currency transformed to master currency
67	T_PRICE	value of value in other currency transformed to master currency
68	T_PRICE	value of cash in drawer of master currency
69	T_PRICE	value of cash in drawer of secondary currency
70	T_PRICE	value of other currency transferred to master currency in cash drawer
71	T_PRICE	value of check in drawer of master currency
72	T_PRICE	value of check in drawer of secondary currency
73	T_PRICE	value of credit in drawer of master currency
74	T_PRICE	value of credit in drawer of secondary currency
75	T_PRICE	<i>reserved</i>
76	T_PRICE	<i>reserved</i>
77	T_QUANTITY	number of cash received on account of master currency
78	T_PRICE	value of cash received on account of master currency
79	T_QUANTITY	number of cash received on account of secondary currency
80	T_PRICE	value of cash received on account of secondary currency
81	T_QUANTITY	number of cheques received on account of master currency
82	T_PRICE	value of cheques received on account of master currency
83	T_QUANTITY	number of cheques received on account of secondary currency
84	T_PRICE	value of cheques received on account of secondary currency
85	T_QUANTITY	number of credit received on account of master currency
86	T_PRICE	value of credit received on account of master currency
87	T_QUANTITY	number of credit received on account of secondary currency
88	T_PRICE	value of credit received on account of secondary currency
89	T_QUANTITY	number of cash paid out in master currency
90	T_PRICE	value of cash paid out in master currency

91	T_QUANTITY	number of cash paid out in secondary currency
92	T_PRICE	value of cash paid out in secondary currency
93	T_QUANTITY	number of cheques paid out in master currency
94	T_PRICE	value of cheques paid out in master currency
95	T_QUANTITY	number of cheques paid out in secondary currency
96	T_PRICE	value of cheques paid out in secondary currency
97	T_QUANTITY	number of credit paid out in master currency
98	T_PRICE	value of credit paid out in master currency
99	T_QUANTITY	number of credit paid out in secondary currency
100	T_PRICE	value of credit paid out in secondary currency
101	T_STRING[6]	date of last Z report
102	T_STRING[6]	time of last Z report
103	T_QUANTITY	<i>number of invoices</i>
104	T_QUANTITY	<i>reserved</i>
105	T_PRICE	<i>reserved</i>

Example:

- The command executes the function No.12 (financial report reading) at the ECR connected to port No.1:

```
txt2500.exe 1 /f12
```

- Example of *head.out* file for financial report execution:

```
0:3;1 //report type
1:3;1 //mode
```

- Example of input data file:

```
0;0
```

- File *data.in* contains the report
- Example of *data.in* file:

```
0;596,8000;596,8000;0,0000;0;0;0;4;6;0,0000;317,8000;279,0000;0,0000;0,0000;0,0000;0,0000;0,0000;278,800
0;232,5000;0,0000;0,0000;0,0000;0,0000;0,0000;39,0000;46,5000;0,0000;0,0000;0,0000;0,0000;596,8000;0,0000;
4,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,
0000;0,0000;0,0000;4,0000;596,8000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,
0000;0,0000;0,0000;0,0000;0,0000;0,0000;596,8000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,
0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,00
00;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;0,0000;
```

4.3.4 Electronic journal

4.3.4.1 Read first journal item

Input parameters

MemID	Data type	Description
0	T_FLAG1	journal type: 0 – binary, 1 – text
1	T_FLAG1	delete after reading: 0 – no, 1 – yes
2	T_FLAG1	reserved

Output parameters

MemID	Data type	Description
0	T_FLAG2	return code
1	T_FLAG1	next journal item: 0 – no, 1 – yes
2	T_FLAG2	number of fully read bills/rows
3	T_RAWSTRING	binary journal data

Example:

- The command executes the function No.20 (electronic journal reading) at the ECR connected to port No.1:

```
txt2500.exe 1 /f20
```

- Example of *head.out* file for electronic journal reading:

```
0:3;1 //journal type
1:3;1 //delete after reading
2:3;1 //reserved
```

- Example of input data file:

```
0;0;0
```

- The file *data.in* contains the output parameters (return code of the operation, number of bills)
- Example of *data.in* file:

```
0;0;6
```

- the *head.in* file contains the description of the output parameters:

```
0:4;2 //return code
1:3;1 //next journal item
2:4;2 //number of fully read bills/rows
```

- the *journal.cfg* file contains the journal configuration in the binary format
- the *journal.bin* file contains the journal itself in the binary format
- the *journal.txt* file contains the journal itself converted into the text format
- command for converting journal has a following format:

```
txt2500.exe 1 /f20c
```

4.3.4.2 Structure of "journal.txt" file

- for textual journal, file contains copy of journal receipt

Example:

```

                DKP
                DIC
                ICO
26-03-2003 10:49:00 Wednesday
01*00002                SMITH

LEMONADE B                =10,00
ICE TEA B                 =15,00
PRESSO B                  =12,50
VIENNA COFFEE B          =14,50
BLACK TEA B               =11,00
FRUIT TEA B               =11,00
PLZEN 0,5l B              =24,00
PLZEN 0,3l B              =17,00
GAMBRINUS 0,5l B          =26,00
GAMBRINUS 0,3l B          =19,00
PLZEN BOTTLE 0,3l B       =22,00
GAMBRINUS BTL. 0,5l B     =23,00
SARIS BTL. 0,5l B         =24,00
SARIS CAN 0,5l B          =27,00
~~~~~
TAXABLE(B)                256,00
NET(B)                    213,33
VAT(B)                     42,67
TOTAL                      256,00
~~~~~
CASH                       $256,00
Number of Receipt          *2

```

~~4.01.00~~

- for binary journal, file structure is following:

descID:atr1;atr2;...

- **descID** - identifier number of record:
 - 10 - header
 - 20 - bill item
 - 30 - payment
 - 40 - tax record
 - 50 - customer record
 - 60 - financial record
 - 70 - PO
 - 80 - RA
 - 90 - daily Z report
 - 100 - monthly Z report
 - 110 - yearly Z report
- **atrX** - record attributes
 - field items marked '●' are always stored in field
 - field items marked '○' are configurable by system flags settings (depend to journal flags settings)

❑ **header (10) - for bill item, payment, taxes, customer and financial record**

- record type:
 - 1 - sale
 - 3 - sale in correction mode
 - 5 - sale in training mode
- sale items count
- promo items count
- payments count
- cash paid: 0 – no, 1 – yes
- subtotal surcharge / discount count
- used taxes count
- receipt number
- fiscal receipt number
- start sale date: DDMMYY
- end sale date: DDMMYY
- start sale time: HHMMSS
- end sale time: HHMMSS
- document number
- ECR number
- cashier ID
- cashier name
- customer ID: 0 if no customer assigned
- bill number_

Example:

10:1;14;0;1;0;0;1;7;5;250303;123405;250303;123418;;1;1;NOVÁK ml.;0;990;

4.01.00

☐ **bill item (20)**

- record type:

0	0	0	0	0	x	x	x
---	---	---	---	---	---	---	---

void: 0 – no, 1 – yes

returnable container: 0 – no, 1 – yes

quantity dividing/multiplying: 0 – $Q = Q1 / Q2$, 1 – $Q = Q1 \times Q2$

- tax ID
- department ID
- PLU ID
- programmed PLU price
- total quantity
- price
- total price
- discount
- subtotal discount
- promo discount
- tax rate

Example:

20:0;2;2;27;1000;1000;10.000;10.000;0.000;0.000;0.000;2000;

☐ **payment (30)**

- payment type: 10 – cash, 11 – cheque, 12 – credit, 13 – , 14 –
- payment ID
- currency: 0 – master, 1 – secondary, 2 – other
- value
- descriptive text 1
- descriptive text 2

Note: if no payment is included in electronic journal, it means VIP payment

Example:

30:10;1;0;256.000;;;

☐ **tax record (40)**

- tax ID
- tax type
- taxable value
- net value
- tax value
- tax rate

Example:

40:2;0;256.000;213.300;42.700;2000;

☐ **customer record (50)**

- credit
- limit
- discount
- discount type

Example:

50:10.000;10.000;1.000;0;

☐ **financial record (60)**

- total sale value in master currency
- difference
- void quantity
- void value
- refund quantity
- refund value
- surcharge quantity
- surcharge value
- discount quantity
- discount value
- sold pack quantity
- sold pack value
- returned pack quantity
- returned pack value

Example:

60:256.000;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;

☐ **header (10) - for RA and PO**

- record type:
 - 8 - RA
 - 10 - RA in correction mode
 - 12 - RA in training mode
 - 16 - PO
 - 18 - PO in correction mode
 - 20 - PO in training mode
- receipt number
- start sale date: DDMMYY
- start sale time: HHMMSS
- ECR number
- cashier ID
- cashier name

Example:

10:8;4;250303;152001;1;1;NOVÁK ml.;

☐ **RA (70), PO (80)**

- value
- currency type: 0 – master, 1 – secondary, 2 – other
- payment type: 10 – cash, 11 – cheque, 12 – credit, 13 – , 14 –

Example:

70:100000.000;0;12;

80:1000.000;0;10;

~~4.01.00~~

❑ **header (10) - for Z report (daily, monthly, yearly)**

- record type:
 - 32 - *daily*
 - 34 - *daily in correction mode*
 - 36 - *daily in training mode*
 - 64 - *monthly*
 - 66 - *monthly in correction mode*
 - 68 - *monthly in training mode*
 - 128 - *yearly*
 - 130 - *yearly in correction mode*
 - 132 - *yearly in training mode*
- receipt number
- start sale date: DDMMYY
- start sale time: HHMMSS
- ECR number
- cashier ID
- cashier name

Example:

10:32;12;250303;135557;1;1;NOVÁK ml.;

❑ **daily (90), monthly (100) and yearly (110) Z report**

- grand total 1
- grand total 2
- grand total 3
- number of daily reports
- number of monthly reports
- number of MR
- number of fiscal receipts
- number of all receipts
- taxable 1
- taxable 2
- taxable 3
- taxable 4
- taxable 5
- taxable 6
- taxable 7
- net 1
- net 2
- net 3
- net 4
- net 5
- net 6
- net 7
- tax 1
- tax 2
- tax 3
- tax 4
- tax 5
- tax 6
- tax 7
- total sale
- difference
- number of done or finished sales
- number of corrections – absolute value
- value of corrections – absolute value
- number of refunds
- value of refunds

- number of surcharges
- value of surcharges
- number of discounts
- value of discounts
- number of sold returnable packs
- value of sold returnable packs
- number of bought-out returnable packs
- value of bought-out returnable packs
- value of refund receipts – absolute value
- number of refund receipts
- value of refund receipts
- number of cancelled receipts – absolute value
- value of cancelled receipts – absolute value
- number of cash payment in master currency
- value of cash payment in master currency
- number of cash payment in secondary currency
- value of cash payment in secondary currency
- number of cheque payment in master currency
- value of cheque payment in master currency
- number of cheque payment in secondary currency
- value of cheque payment in secondary currency
- number of credit payment in master currency
- value of credit payment in master currency
- number of credit payment in secondary currency
- value of credit payment in secondary currency
- number of customer credit payment in master currency
- value of customer credit payment in master currency
- number of customer credit payment in secondary currency
- value of customer credit payment in secondary currency
- number of value in other currency transformed to master currency
- value of value in other currency transformed to master currency
- value of cash in drawer of master currency
- value of cash in drawer of secondary currency
- value of other currency transferred to master currency in cash drawer
- value of cheques in drawer of master currency
- value of cheques in drawer of secondary currency
- value of credit in drawer of master currency
- value of credit in drawer of secondary currency
- *reserved*
- *reserved*
- number of cash received on account of master currency
- value of cash received on account of master currency
- number of cash received on account of secondary currency
- value of cash received on account of secondary currency
- number of cheques received on account of master currency
- value of cheques received on account of master currency
- number of cheques received on account of secondary currency
- value of cheques received on account of secondary currency
- number of credit received on account of master currency
- value of credit received on account of master currency
- number of credit received on account of secondary currency
- value of credit received on account of secondary currency
- number of cash paid out in master currency
- value of cash paid out in master currency
- number of cash paid out in secondary currency
- value of cash paid out in secondary currency
- number of cheques paid out in master currency
- value of cheques paid out in master currency

~~4.01.00~~

- number of cheques paid out in secondary currency
- value of cheques paid out in secondary currency
- number of credit paid out in master currency
- value of credit paid out in master currency
- number of credit paid out in secondary currency
- value of credit paid out in secondary currency
- date of last Z report
- time of last Z report
- *number of invoices*
- *reserved*
- *reserved*

Example:

90:1361.000;1361.000;0.000;0;0;0;8;12;0.000;719.000;642.000;0.000;0.000;0.000;0.000;0.000;630.700;53
5.000;0.000;0.000;0.000;0.000;0.000;88.300;107.000;0.000;0.000;0.000;0.000;1361.000;0.000;8;0;0.00
0;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;8;1361.000;0;0.000;0;0.000;0;0.000;0;
0.000;0;0.000;0;0.000;0;0.000;0;0.000;1361.000;0.000;0.000;0.000;0.000;0.000;0.000;0.000;0;0.0
00;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;
100:1361.000;1361.000;0.000;1;0;0;8;13;0.000;719.000;642.000;0.000;0.000;0.000;0.000;630.700;5
35.000;0.000;0.000;0.000;0.000;0.000;88.300;107.000;0.000;0.000;0.000;0.000;1361.000;0.000;8;0;0.0
00;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;8;1361.000;0;0.000;0;0.000;0;0.000;0
;0.000;0;0.000;0;0.000;0;0.000;0;0.000;1361.000;0.000;0.000;0.000;0.000;0.000;0.000;0.000;0;0.
000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;
0;
110:1361.000;1361.000;0.000;1;0;0;8;13;0.000;719.000;642.000;0.000;0.000;0.000;0.000;630.700;5
35.000;0.000;0.000;0.000;0.000;0.000;88.300;107.000;0.000;0.000;0.000;0.000;1361.000;0.000;8;0;0.0
00;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;8;1361.000;0;0.000;0;0.000;0;0.000;0
;0.000;0;0.000;0;0.000;0;0.000;0;0.000;1361.000;0.000;0.000;0.000;0.000;0.000;0.000;0.000;0;0.
000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;0;0.000;
0;

4.3.4.3 Delete electronic journal

Input parameters

MemID	Data type	Description
0	T_FLAG1	journal type: <i>0 – binary,</i> <i>1 – text</i>
1	T_FLAG4	number of deleted bills/rows (0xFFFF for all)
2	T_FLAG1	<i>reserved</i>

Output parameters

MemID	Data type	Description
0	T_FLAG2	return code
1	T_FLAG4	<i>reserved</i>

Example:

- The command executes the function No.23 (delete electronic journal) at the ECR connected to port No.1:

txt2500.exe 1 /f23

- Example of *head.out* file for delete electronic journal:

```
0:3;1 //journal type:0 – binary, 1 – text
1:5;4 //number of deleted bills/rows (0xFFFF for all)
2:3;1 //reserved
```

- Example of input data file:

0;65535;0

- File *data.in* contains the return code
- Example of *data.in* file:

0;0;

4.3.5 ECR blocking

4.3.5.1 Block ECR

Input parameters

MemID	Data type	Description
0	T_FLAG1	<i>reserved</i>

Output parameters

MemID	Data type	Description
0	T_FLAG2	return code

Note: – if ECR is blocked, no keyboard input is available

Example:

- The command executes the function No.30 (block ECR) at the ECR connected to port No.1:

```
txt2500.exe 1 /f30
```

- Example of *head.out* file for block ECR:

```
0:3;1 //reserved
```

- Example of input data file:

```
0
```

- File *data.in* contains the return code
- Example of *data.in* file:

```
0;0;
```

4.3.5.2 Unblock ECR

Input parameters

MemID	Data type	Description
0	T_FLAG1	<i>reserved</i>

Output parameters

MemID	Data type	Description
0	T_FLAG2	return code

Note: – if ECR stay blocked, turn off and on to unblock it

Example:

- The command executes the function No.31 (unblock ECR) at the ECR connected to port No.1:

```
txt2500.exe 1 /f31
```

- Example of *head.out* file for unblock ECR:

```
0:3;1 // reserved
```

- Example of input data file:

```
0
```

- File *data.in* contains the return code
- Example of *data.in* file:

```
0;0;
```

4.3.6 Set master currency, TAX levels, Reregister ECR

4.3.6.1 Set master currency

Input parameters

MemID	Data type	Description
0	T_STRING[33]	password printed on
1	T_FLAG1	<ul style="list-style-type: none"> ▪ 0 – both sides ▪ 1 – left side only
2	T_FLAG1	number of decimal places for prices rounding ^D : (0 – 3)
3	T_FLAG1	number of decimal places for taxes rounding ^D : (0 – 3)
4	T_FLAG1	number of decimal places for total sale price rounding ^D : (0 – 3)
5	T_FLAG1	number of decimal places for displaying taxes, prices and total sale price ^D : (0 – 3)
6	T_FLAG1	price rounding method ^D : 0 - standard - $<0.00 \div 0.50 = 0.00$, $<0.50 \div 1.00 = 1.00$, 1 - up, 2 - down, 3 - quarter $<0.00 \div 0.25 = 0.00$, $<0.25 \div 0.75 = 0.5$, $<0.75 \div 1.00 = 1.00$
7	T_FLAG1	taxes rounding method ^D : (0 – 3) (same meaning as price)
8	T_FLAG1	total sale price rounding method ^D : (0 – 3) (same meaning as price)
9	T_FLAG1	allow negative sum in drawer for master currency: 0 - no, 1 - yes
10	T_FLAG1	tax system: 0 - European, 1 - American
11	T_FLAG1	currency text position ^D : 0 - behind price, 1 - in front of price, 2 - none
12	T_STRING[3]	master currency text ^D

Note:

- flag 4 must be \geq than flags 1 – 3
- master currency text can be an empty string and can't contain the numbers

Output parameters

MemID	Data type	Description
0	T_FLAG2	return code

Example:

- The command executes the function No.40 (Set master currency) at the ECR connected to port No.1:

```
txt2500.exe 1 /f40
```

- Example of head.out file for Set master currency:

```
0:6;33 //password
1:3;1 //printed on: 0 – both sides / 1 – left side only
2:3;1 //number of decimal places for prices rounding D: (0 – 3)
3:3;1 //number of decimal places for taxes rounding D: (0 – 3)
4:3;1 //number of decimal places for total sale price rounding D: (0 – 3)
5:3;1 //number of decimal places for displaying taxes, prices and total sale price D: (0 – 3)
6:3;1 //price rounding method D: 0 - standard / 1 - up / 2 - down / 3 - quarter
7:3;1 //taxes rounding method D:
8:3;1 //total sale price rounding method
9:3;1 //allow negative sum in drawer for master currency: 0 - no / 1 - yes
```

~~4.01.00~~

10:3;1 //tax system: 0 - European / 1 - American

11:3;1 //currency text position D: 0 - behind price / 1 - in front of price / 2 - none

12:6;3 //master currency text

- Example of input data file:

(password);1;1;1;2;0;0;0;0;0;0;\$

- File *data.in* contains the return code
- Example of *data.in* file:

0;0;

4.3.6.2 Set TAX levels

Input parameters

MemID	Data type	Description
0	T_STRING[33]	password printed on
1	T_FLAG1	<ul style="list-style-type: none"> 0 – both sides 1 – left side only
2	T_PERCENT	tax rate 1 ^D : 0 – 10000 (0.00% – 100.00%)
3	T_FLAG1	tax flag 1 ^D : 0 - VAT, 1 - TAX
4	T_FLAG1	tax counting method 1 ^D : 0 – at the end of day (for each tax 1 – 6), 1 – at the end of receipt
5	T_FLAG1	type of tax level: 0 – normal, 1 – non-taxable tax level
6	T_PERCENT	tax rate 2 ^D : 0 – 10000 (0.00% – 100.00%)
7	T_FLAG1	tax flag 2 ^D : 0 - VAT, 1 - TAX
8	T_FLAG1	tax counting method 2 ^D : 0 – at the end of day (for each tax 1 – 6), 1 – at the end of receipt
9	T_FLAG1	type of tax level: 0 – normal, 1 – non-taxable tax level
10	T_PERCENT	tax rate 3 ^D : 0 – 10000 (0.00% – 100.00%)
11	T_FLAG1	tax flag 3 ^D : 0 - VAT, 1 - TAX
12	T_FLAG1	tax counting method 3 ^D : 0 – at the end of day (for each tax 1 – 6), 1 – at the end of receipt
13	T_FLAG1	type of tax level: 0 – normal, 1 – non-taxable tax level
14	T_PERCENT	tax rate 4 ^D : 0 – 10000 (0.00% – 100.00%)
15	T_FLAG1	tax flag 4 ^D : 0 - VAT, 1 - TAX
16	T_FLAG1	tax counting method 4 ^D : 0 – at the end of day (for each tax 1 – 6), 1 – at the end of receipt
17	T_FLAG1	type of tax level: 0 – normal, 1 – non-taxable tax level

18	T_PERCENT	tax rate 5 ^D : 0 – 10000 (0.00% – 100.00%)
19	T_FLAG1	tax flag 5 ^D : 0 - VAT, 1 - TAX
20	T_FLAG1	tax counting method 5 ^D : 0 – at the end of day (for each tax 1 – 6), 1 – at the end of receipt
21	T_FLAG1	type of tax level: 0 – normal, 1 – non-taxable tax level
22	T_PERCENT	tax rate 6 ^D : 0 – 10000 (0.00% – 100.00%)
23	T_FLAG1	tax flag 6 ^D : 0 - VAT, 1 - TAX
24	T_FLAG1	tax counting method 6 ^D : 0 – at the end of day (for each tax 1 – 6), 1 – at the end of receipt
25	T_FLAG1	type of tax level: 0 – normal, 1 – non-taxable tax level
26	T_PERCENT	tax rate 7 ^D : 0 – 10000 (0.00% – 100.00%)
27	T_FLAG1	tax flag 7 ^D : 0 - VAT, 1 - TAX
28	T_FLAG1	tax counting method 7 ^D : 0 – at the end of day (for each tax 1 – 6), 1 – at the end of receipt
29	T_FLAG1	type of tax level: 0 – normal, 1 – non-taxable tax level

Output parameters

MemID	Data type	Description
0	T_FLAG2	return code

Example:

- The command executes the function No.41 (Set TAX levels) at the ECR connected to port No.1:

txt2500.exe 1 /f41

- Example of *head.out* file for Set TAX levels:

```
0:6;33 //password
1:3;1 //printed on: 0 – both sides / 1 – left side only
2:2;2 //tax rate 1
3:3;1 //tax flag 1D: 0 - VAT / 1 – TAX
4:3;1 //tax counting method 1D: 0 – at the end of day (for each tax 1 – 6) / 1 – at the end of receipt
5:3;1 //type of tax level: 0 – normal / 1 – non-taxable tax level
6:2;2 //tax rate 2
7:3;1 //tax flag 2D: 0 - VAT / 1 – TAX
8:3;1 //tax counting method 2D: 0 – at the end of day (for each tax 1 – 6) / 1 – at the end of receipt
9:3;1 //type of tax level: 0 – normal / 1 – non-taxable tax level
10:2;2 //tax rate 3
11:3;1 //tax flag 3D: 0 - VAT / 1 – TAX
12:3;1 //tax counting method 3D: 0 – at the end of day (for each tax 1 – 6) / 1 – at the end of receipt
13:3;1 //type of tax level: 0 – normal / 1 – non-taxable tax level
14:2;2 //tax rate 4
15:3;1 //tax flag 4D: 0 - VAT / 1 – TAX
16:3;1 //tax counting method 4D: 0 – at the end of day (for each tax 1 – 6) / 1 – at the end of receipt
```

17:3;1 //type of tax level: 0 – normal / 1 – non-taxable tax level
 18:2;2 //tax rate 5
 19:3;1 //tax flag 5D: 0 - VAT / 1 – TAX
 20:3;1 //tax counting method 5D: 0 – at the end of day (for each tax 1 – 6) / 1 – at the end of receipt
 21:3;1 //type of tax level: 0 – normal / 1 – non-taxable tax level
 22:2;2 //tax rate 6
 23:3;1 //tax flag 6D: 0 - VAT / 1 – TAX
 24:3;1 //tax counting method 6D: 0 – at the end of day (for each tax 1 – 6) / 1 – at the end of receipt
 25:3;1 //type of tax level: 0 – normal / 1 – non-taxable tax level
 26:2;2 //tax rate 7
 27:3;1 //tax flag 7D: 0 - VAT / 1 – TAX
 28:3;1 //tax counting method 7D: 0 – at the end of day (for each tax 1 – 6) / 1 – at the end of receipt
 29:3;1 //type of tax level: 0 – normal / 1 – non-taxable tax level

- Example of input data file:

(password);1;0,00;0;0;1;19,00;0;0;0;100,00;0;0;0;0,00;0;0;0;100,00;0;0;0;100,00;0;0;0;100,00;0;0;0

- File *data.in* contains the return code
- Example of *data.in* file:

0;0;

4.3.6.3 Reregister ECR

Input parameters

MemID	Data type	Description
0	T_STRING[33]	password
1	T_FLAG1	printed on <ul style="list-style-type: none"> 0 – both sides 1 – left side only
2	T_STRING[33]	TAX payer number

Output parameters

MemID	Data type	Description
0	T_FLAG2	return code

Example:

- The command executes the function No.42 (Reregister ECR) at the ECR connected to port No.1:

txt2500.exe 1 /f42

- Example of *head.out* file for Reregister ECR:

0:6;33 //password
 1:3;1 //printed on: 0 – both sides / 1 – left side only
 2:6;33 //TAX payer number

- Example of input data file:

(password);1;(TAX payer)

- File *data.in* contains the return code
- Example of *data.in* file:

0;0;

4.3.7 Reading and Decoding of FM**4.3.7.1 Read settings for Get FM****Input parameters**

MemID	Data type	Description
0	T_FLAG1	type of call: 0 – read by index 1 – read by date 2 – read full FM (all others parameters, except password, are ignored)
1	T_FLAG1	Record type 0 – all records 1 – FM header 2 – Reregister ECR 3 – Master currency changes 4 – TAX levels changes 5 – daily reports 6 – MASTER resets
2	T_STRING[6]	bottom index (depends from type of call) type of call = 0: index (number as text) 1: date (format is DDMMYY)
3	T_STRING[6]	upper index (depends from type of call) type of call = 0: index (number as text) 1: date (format is DDMMYY)
4	T_STRING[33]	password

Output parameters

MemID	Data type	Description
0	T_FLAG2	return code
1	T_FLAG2	remaining iterations

4.3.7.2 Get next FM**Input parameters**

MemID	Data type	Description
0	T_FLAG1	number of iterations to get: 0 – end of Get FM function <i>n</i> – number of records (98 B) or packets (256 B) FM for read <i>n</i> muss be „0 < <i>n</i> < 8“ for records or 0 < <i>n</i> < 4 for Get full FM

Output parameters

MemID	Data type	Description
0	T_FLAG2	return code
1	T_FLAG2	remaining iterations
3	T_RAWSTRING	dat in first two bytes is length of actual data (length = first byte * 256 + second byte)

Example:

- The command executes the function No.50 (Get FM) at the ECR connected to port No.1:
 read ~~and decode~~ FM
 txt2500.exe 1 /f50
 ~~txt2500.exe 1 /f50d~~
 read and save binary FM without decoding
 txt2500.exe 1 /f50b
 read and save FM in HEX format
 txt2500.exe 1 /f50a
 ~~decode FM only~~
 ~~txt2500.exe 1 /f50e~~
- Example of *head.out* file for GetFM:: (when execute param is „/f50c“ is file ignored)
 - 0:3;1 //type of call: 0 – read by index / 1 – read by date / 2 – read full FM (all others parameters, except password, are ignored)
 - 1:3;1 //Record type 0 – all records / 1 – FM header / 2 – Reregister ECR / 3 – Master currency changes / 4 – TAX levels changes / 5 – daily reports / 6 – MASTER resets
 - 2:6;6 //bottom index (depends from type of call) type of call : 0: index (number as text) / 1: date (format is DDMMYY)
 - 3:6;6 //upper index (depends from type of call) type of call : 0: index (number as text) / 1: date (format is DDMMYY)
 - 4:6;33 //password
- Example of input data file: ~~(when execute param is „/f50c“ is file ignored)~~
 0;0;1;65535;(password);