Serial protocol for printer P2DS

Introduction:

Serial port setting for communication with PC is standardized with „8,n,1,n“ (data bits - 8, parity - none, stop bitas – 1, flow control - none). Supported communication speeds are: 9600, 1920, 38400, 57600, 115200, 230400 and 460800 bps. Before communication start DTR (Data Terminal Ready) line must be activated.

Command types:

Each command has standard form and variable length. Commands have following structure:

Short command:

STX	LEN	DATA	CRC

STX: 0x02
LEN: length of DATA part
CRC: checksum not including **STX and WAIT** bytes (**LEN** and **DATA** block included) placed in two bytes in order MSB,LSB.

Long command :

SOH	LEN1	LEN2	DATA	CRC

SOH: 0x03
LEN1 and LEN2 : length of DATA part (**DATA_LEN**=**LEN2**<<8|**LEN1 -** LSB,MSB) which is not greater than 512.
CRC: check sum not including **SOH** and **WAIT** bytes (**LEN1**, **LEN2** and **DATA** block included) placed in two bytes in order MSB,LSB.

STATUS: For each sent command there is **STATUS** byte returned from device, which can be:

ACK (0x06): command is correctly sent and accepted by device
NACK(0x15): There is an error in command, i.e. command form is not correctly interpreted (CRC is wrong). In this case host sends same command again, for maximum three times in a row. This is just in case to avoid any disturbance during data sending to serial port.

Device will respond with same command sent by host (PC application). After that, host should answer with status byte **ACK/NACK** depending on calculated CRC.

Communication cycle:

This is short explanation of sending one command and reading answer from printer:
PC: Sends command
Printer: Responds with **ACK/NACK***
*If **ACK**:
Printer: Sends **RESPONSE** command which may be preceded with **WAIT** byte(s) (0x08). This byte is sent when operation in printed is taking time to be executed, so host should wait until **WAIT** byte(s) disappears. Printer may send one or more **WAITs** depending on requested operation.
PC: Responds with **ACK/NACK –** if PC detects wrong CRC, by sending NACK, printer may be requested to repeat block sending for maximum three times.
*If **NACK**:
PC: In this case, host may try sending same command for maximum three times, to avoid possible disturbance in communication on serial port.

RESPONSE: Data block with same form as command sent by PC application – short or long command depending on first sent byte **STX** or **SOH.** Command byte (byte which identifies requested command) is identical to the one sent by PC application or is set to 0x7F. This byte is followed by error type byte (error code table is described on last page of this document). If error byte is 0, it means that there is no error in command execution, i.e. command is executed correctly (this response is usualy sent for commands which perform printing on device, not commands used for reading data).

WAIT: After received **ACK** (0x06), PC waits for device status of executed command. If device is busy or there is some obstacle in current command execution (for example, printer head is risen), device will respond with **WAIT** byte (0x08) for each 300ms until command execution is finished and execution status is sent. If there is an error on device display, **WAIT** byte (0x09) is returned, and if there is error on printer, device returns **WAIT** byte (0x07) with byte which desribes error type.**

** Version 11.23 has a bug where printer sends byte with wrong error byte description after **WAIT** byte 0x07. It is recommended to process this error as universal printer error.

Synthax:

In further text, command blocks will be shown by following synthax:

<N> number of bytes.

[NAME] field name

(N)* repeat number of given segment.

{} segment.

Example:

[CMD <1>] (N)*{[CODE <4>][PRICE <4>]}

Command consists of field CMD which length is 1 byte and N segments where each segment contains two fields – field CODE which length is 4 bytes and field PRICE which length is also 4 bytes.

Commands for fiscal reports:

Command for daily report 88 (0x58).

PC sends:

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

Example:

PC: 02 01 58 00 59

02 - STX
01 - LEN
58 - CMD
00 59 - CRC

Printer response: 06 08 08 08 08 08 08 08 08 08 08 08 02 02 7F 00 00 81

06 - ACK
08 - WAIT
02 - STX
02 - LEN
7F 00 – confirmation that command is executed with no errors
00 81 - CRC (sum of all sent bytes, not including STX, ACK and WAIT)

Command for current state report 89 (0x59).

PC sends:

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

Example:

PC: 02 01 59 00 5A

02 - STX
01 - LEN
59 - CMD
00 5A - CRC

Printer response: 06 08 08 08 08 08 08 08 08 08 08 08 08 02 02 7F 00 00 81

06 - ACK
08 - WAIT
02 - STX
02 - LEN
7F 00 - confirmation that command is executed with no errors
00 81 - CRC

Command for executing periodical report 90 (0x5A).

PC sends:

[STX<1>][LEN<1>][CMD<1>][START TIME <8>][END TIME <8>][CRC<2>]

Time is sent as number of miliseconds from 1.1.2000.

Example: periodical report from March the 7th 2012. to April the 5th 2012.

PC: 02 11 5A D3 1E 01 82 59 00 00 00 00 99 22 17 5A 00 00 00 03 64

02 - STX
11 - LEN
5A - CMD
D3 1E 01 82 59 00 00 00 – Start time in ms (from 1.1.2000.) (LSB, MSB)
00 99 22 17 5A 00 00 00 – End time in ms (from 1.1.2000.) (LSB, MSB)
03 64 - CRC

Printer response: 06 08 08 08 08 08 08 08 08 08 08 08 02 02 7F 00 00 81

06 - ACK
08 - WAIT
02 - STX
02 - LEN

7F 00 - confirmation that command is executed with no errors
00 81 - CRC

Communication test command:

Communication test command 101 (0x65).

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
CRC – check sum

Example 1: Communication test.

PC: 02 01 65 00 66

02 - STX , start of communication
01 - LEN command length
65 - CMD command for communication test
00 66 - CRC check sum of all sent bytes, not including STX placed in two bytes (65+1=66)

Printer: 06

06 - ACK printer reponds with ok, test is successful

Command for device settings:

Command for programming device settings 110 (0x6E).

PC sends:

[STX<1>][LEN<1>][CMD<1>][SERIAL PORT SPEED <4>][EMPTY <4>][OTHER SETTINGS <4>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
SERIAL PORT SPEED – PC communication speed. Available communication speeds: 9600, 19200, 38400, 57600, 115200, 230400 and 460800 bps. After changing this parameter, printer shoud be switched off, then turned on, so setting could be applied.
EMPTY – This field is sent becaues command has a unique form for all HCP devices. For HCP fiscal cash registerr, this field represents barcode port speed. This setting is not available in P2DS printer, but is set to 9600 and cannot be changed. You may send any value within 4 bytes.
OTHER SETTINGS – In this field, each bit is dedicated to some parameter of various device settings.
CRC – check sum

Detailed description of OTHER SETTINGS (in bits):

[PRINTING INTENSITY <3>][PRINTING SPEED <3>][FOOTER LENGTH <3>][EMPTY <1>][EXTERNAL DISPLAY <1>][CASHIERS <1>][EMPTY<4>][ADVERTISEMENT NUMBER<4>][EMPTY <4>][USE CUTTER <1>][EMPTY <7>]

By offset:

PRINTING INTENSITY - 0 (values between 0 and 6)
PRINTING SPEED - 3 (values between 0 and 6)
FOOTER LENGTH - 6 (values between 0 and 6)
EXTERNAL DISPLAY - 10 (0 if turned off, 1 if turned on)
USE CASHIERS - 11 (0 if turned off, 1 if turned on)
ADVERTISEMENT NUMBER - 16 (0 if turned off, 1 if turned on)
USE CUTTER -24 (0 if turned off, 1 if turned on)
EMPTY – Empty fields, settings are used on other devices.

Command for reading device settings 111 (0x6F).

PC sends:

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
CRC – check sum

Printer response:

[STX<1>][LEN<1>][CMD<1>][SERIAL PORT SPEED <4>][SERIAL PORT SPEED DISPLAY <4>][OTHER SETTINGS <4>][CRC<2>]

STX – start of commucation
LEN – command length
CMD - command
SERIAL PORT SPEED - PC communication speed. Available communication speeds: 9600, 19200, 38400, 57600, 115200, 230400 and 460800 bps.
SERIAL PORT SPEED DISPLAY- fixed value 9600 bps.
OTHER SETTINGS - In this field, each bit is dedicated to some parameter of various device settings..
CRC – check sum

Detailed description of OTHER SETTINGS (in bits):

[PRINTING INTENSITY <3>][PRINTING SPEED <3>][FOOTER LENGTH <3>][EMPTY <1>][EXTERNAL DISPLAY <1>][CASHIERS <1>][EMPTY<4>][ADVERTISEMENT NUMBER<4>][EMPTY <4>][USE CUTTER <1>][EMPTY <7>]

By offset:

PRINTING INTENSITY - 0 (values between 0 and 6)
PRINTING SPEED - 3 (values between 0 and 6)
FOOTER LENGTH - 6 (values between 0 and 6)
EXTERNAL DISPLAY - 10 (0 if turned off, 1 if turned on)
USE CASHIERS - 11 (0 if turned off, 1 if turned on)
ADVERTISEMENT NUMBER - 16 (0 if turned off, 1 if turned on)
USE CUTTER -24 (0 if turned off, 1 if turned on)
EMPTY – Empty fields, settings are used on other devices.

Command for printing settings 112 (0x70).

[STX<1>][LEN<1>][CMD<1>][OPTION<1>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
OPCIJA – to print settings send 0, to print GPRS settings send 1
CRC – check sum

Example: print settings.

PC: 02 02 70 00 00 72

02-STX
02-LEN
70 - CMD
00 - OPTION, value is 0 since regular settings are printed
00 72 -CRC

Štampač: 06 02 02 7f 00 00 81

06 - ACK
02 - STX
02 - LEN
7f 00 - confirmation that command is executed with no errors
00 81 - CRC

Commands for article (PLU) programming:

Command for article programming 12 (0x0C).

PC sends:

[STX<1>][LEN<1>][CMD<1>] [CODE <4>][DESCRIPTION <1-32>][M.U.+VAT<1>][PRICE<4>] [CRC<2>]

STX – start of communication
LEN – command length
CMD – command
CODE – article code (unique value) from 1 to 75000.
DESCRIPTION – article name, array with length from 1 to 32. Characters are coded by table at the end of this document.
M.U.+VAT – M.U. or measure unit, is number written in higher 4 bits, and VT is written in lower 4 bits. Measure unit may have values between 0 and 15 and VAT between 0 and 8.
PRICE – article prcice is number written in 4 bytes. This number represents real price value multiplied by 100 (price can have maximum 2 decimals, i.e. numbers after decimal dot).
CRC – check sum (written in two bytes)

Example: programming article with description TEST_ARTICLE, code value 1, measure unit kg, VAT И, price 2550,78.

PC: 02 16 0C 01 00 00 00 54 45 53 54 5F 41 52 54 49 43 4C 45 16 66 E4 03 00 05 29

02 - STX
16 - LEN
0C - CMD
01 00 00 00 (bytes written in LSB first format) - CODE
54 45 53 54 5F 41 52 54 49 43 4C 45 (bytes represent characters in article description) - DESCRIPTION
16 (higher 4 bots represent measure unit and lower 4 bits VAT) – M.U.+VAT
66 E4 03 00 (bytes written in LSB first format) - PRICE
05 29 (check sum written in two bytes, MSB first – sum of all bytes not including STX) - CRC

Printer: 06 02 02 7F 00 00 81

06 - ACK
02 - STX
02 - LEN (length of DATA)
7F - CMD command. Command which represents execution status..

00 – error message byte (in this case there is no error in execution – 00; for error codes, check error code table at the end of this document).
00 81 - CRC (check sum - sum of all bytes, excluding STX i ACK)

Command for article printing 21 (0x15).

PC sends:

[STX<1>][LEN<1>][CMD<1>] [CODE <4>] [CRC<2>]

STX – start of communication
LEN – length of DATA
CMD - command
CODE: article code in range from 1 to 75000.
CRC – check sum

Printer response:

[ACK<1>][STX<1>][LEN<1>][CMD<1>] [CODE <4>] [CRC<2>]

Example: Printing article with code 1

PC: 02 05 15 01 00 00 00 00 1B

02 - STX
05 - LEN
15 - CMD
01 00 00 00 (bytes written in LSB first format) - CODE
00 1B (check sum written in two bytes, MSB first format) - CRC

Printer response: 06 02 05 15 32 00 00 00 00 1C

06 - ACK
02 - STX
05 - LEN (length of DATA)
15 - CMD (command for printing article)
32 00 00 00 - CODE (code of next article to print, LSB, MSB)
00 1C - CRC (check sum of all bytes excluding ACK i STX)
After sending ACK, printer sends command in same form, where CODE is actualy a code for nex article to print. If code value is -1 (0xFFFFFFFF) it means that there are no more articles in printer.

Command for deleting articles 14 (0x0E).

PC sends:

[STX<1>][LEN<1>][CMD<1>] [CRC<2>]

STX – start of communication
LEN – length of DATA
CMD - command
CRC – check sum

Example: Deleting all articles

PC: 02 01 0E 00 0F

02 - STX
01 - LEN
0E - CMD
00 0F (bytes written in MSB first format) - CRC

Printer response: 06 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08

06 - ACK
08 - WAIT

02 02 7F 00 00 81

02 - STX
02 - LEN (length of DATA)
7F - CMD command. Command which returns execution status.
00 – error message byte (in this case there is no error in execution – 00; for error codes, check error code table at the end of this document).
00 81 - CRC (check sum - sum of all bytes, excluding STX i ACK)

Commands for sale execution:

NOTE:

QUANTITY is sent as an integer. Value is obtained by multiplying real value by 1000. Quantity is limited to maximum 5 characters including decimal point (for example 1.234, 12.34, 123.4 or 99999).

PRICE is sent as an integer. Value is obtained by multiplying real value by 100.

NOTE:

Receipt opens automaticaly when first item on bil is sold (first article). Paying begins by sending first payment request. If paying is started, voiding of the bill will not be possible. If sold article quantity is not divisible by 1000 (quantity%1000 not equals 0), only last sold article can be voided, or whole bill can be voided. Receipt is automaticaly closed when sum of all payments exceeds bill value.

Command for selling by code 48 (0x30).

[STX<1>][LEN<1>][CMD<1>][CODE<4>][QUANTITY<4>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
CODE – article code. Integer in range from 1 to 75000 packed in 4 bytes
QUANTITY – value is sent as an integer. It is obtained from real value multiplied by 1000.
CRC – check sum

Example:

PC: 02 09 30 01 00 00 00 E8 03 00 00 01 25

02 - STX start of communication
09 - LEN (length of DATA)
30 - CMD commmand
01 00 00 00 – article CODE. In this case 1. (LSB,MSB)
E8 03 00 00 – QUANTITY, in this case 1000, real value is 1.000 (LSB,MSB)
01 25 - CRC (sum of all bytes except STX, written in two bytes MSB, LSB)

Printer response: 06 02 02 7F 00 00 81

06 - ACK response that command is read correctly - CRC correct.
02 - STX start of communication.
02 - LEN (length of DATA)
7F - CMD command. Command which returns execution status.
00 - STATUS error message byte (in this case there is no error in execution – 00; for error codes, check error code table at the end of this document).
00 81 - CRC (check sum - sum of all bytes, excluding STX i ACK)

PC: 06

06 - response that command is read correctly - CRC correct.

Command for voiding sold article (by code) 50 (0x32).

[STX<1>][LEN<1>][CMD<1>][CODE<4>][QUANTITY<4>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
CODE – article code. Integer in range from 1 to 75000 packed in 4 bytes
QUANTITY - value is sent as an integer. It is obtained from real value multiplied by 1000.
CRC – check sum

NOTE:
In case that code valu is 0, last sold article will be voided on bill (quantity value is not considered). If code value is -1 (0xFFFFFFFF) whole bill will be voided (quantity value is not considered). If quantity value is 0, all items on bill with entered code will be voided on bill. It is not possible to void items which quantity is not an integer number.

Example 1: Voiding by code and quantity.

PC: 02 09 32 01 00 00 00 E8 03 00 00 01 27

02 - STX start of communication
09 - LEN (length of DATA)
32 - CMD command
01 00 00 00 – article CODE. In this case 1. (LSB,MSB)
E8 03 00 00 - QUANTITY. in this case 1000, real value is 1.000 (LSB,MSB) (LSB,MSB)
01 27 - CRC (sum of all bytes except STX, written in two bytes MSB, LSB)

Printer response: 06 02 02 7F 00 00 81

06 - ACK response that command is read correctly - CRC correct.
02 - STX start of communication.
02 - LEN (length of DATA)
7F - CMD command. Command which returns execution status.
00 - STATUS error message byte (in this case there is no error in execution – 00; for error codes, check error code table at the end of this document).
00 81 - CRC (sum of all bytes except STX, written in two bytes MSB, LSB)

PC: 06

06 - ACK response that command is read correctly - CRC correct.

Example 2: Voiding by code and quantity 0 – all articles with given code.

PC: 02 09 32 02 00 00 00 00 00 00 00 00 3D

02 - STX start of communication
09 - LEN (length of DATA)
32 - CMD command
02 00 00 00 – article CODE. In this case 2. (LSB,MSB)
00 00 00 00 - QUANTITY. In this case 0, void all items with given code. (LSB,MSB)
00 3D - CRC (sum of all bytes except STX, written in two bytes MSB, LSB)

Printer response: 06 08 02 02 7F 00 00 81

06 - ACK response that command is read correctly - CRC correct.
08 - WAIT byte. Execution of given command takes some time, so this message is returned to PC.
02 - STX start of communication.
02 - LEN (length of DATA)
7F - CMD command. Command which returns execution status.
00 - STATUS error message byte (in this case there is no error in execution – 00; for error codes, check error code table at the end of this document).
00 81 - CRC (sum of all bytes except STX, written in two bytes MSB, LSB)

PC: 06

06 - ACK response that command is read correctly - CRC correct.

Example 3: Voiding last item.

PC: 02 09 32 00 00 00 00 00 00 00 00 00 3B

02 - STX start of communication
09 - LEN (length of DATA)
32 - CMD command
00 00 00 00 – article CODE. In this case 0 – voiding last sold item on bill. (LSB,MSB)
00 00 00 00 - QUANTITY. In this case 0, but is not considered. (LSB,MSB)
00 3B - CRC (sum of all bytes except STX, written in two bytes MSB, LSB)

Printer response: 06 02 02 7F 00 00 81

06 - ACK response that command is read correctly - CRC correct.
02 - STX start of communication.
02 - LEN (length of DATA)
7F - CMD command. Command which returns execution status.
00 - STATUS error message byte (in this case there is no error in execution – 00; for error codes, check error code table at the end of this document).
00 81 - CRC (sum of all bytes except STX, written in two bytes MSB, LSB)

PC: 06

06 - ACK response that command is read correctly - CRC correct.

Example 4: Voiding whole bill.

PC: 02 09 32 FF FF 00 00 00 00 00 00 02 39

02 - STX start of communication
09 - LEN (length of DATA)
32 - CMD command
FF FF 00 00 – article CODE. In this case 0xFFFF – void whole bill. (LSB,MSB)
00 00 00 00 - QUANTITY. In this case 0 – not considered. (LSB,MSB)
02 39 - CRC (sum of all bytes except STX, written in two bytes MSB, LSB)

Printer response: 06 08 08 02 02 7F 00 00 81

06 - ACK response that command is read correctly - CRC correct.
08 08 - WAIT byte. Execution of given command takes some time, so this message is returned to PC.
02 - STX start of communication.
02 - LEN (length of DATA)
7F - CMD command. Command which returns execution status.
00 - STATUS error message byte (in this case there is no error in execution – 00; for error codes,

check error code table at the end of this document).
00 81 - CRC (sum of all bytes except STX, written in two bytes MSB, LSB)

PC: 06

06 - ACK response that command is read correctly - CRC correct.

Command for paying 51 (0x33).

[STX<1>][LEN<1>][CMD<1>][QUANTITY<8>][TYPE<1>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
QUANTITY – Value for paying the bill/ value is sent as long number which value is obtained from real value multiplied by 100.
TYPE – whole number which defines payment type. 0 – cash, 1 – credit card, 2 - cheque
CRC – check sum

NOTE
If value for quantity is 0, bill will be closed with exact current amount of bill value for given type.

Example 1: Paying with given type and quantity.

PC: 02 0A 33 20 4E 00 00 00 00 00 00 01 00 AC

02 - STX start of communication
0A - LEN (length of DATA)
33 - CMD command
20 4E 00 00 00 00 00 00 - QUANTITY. In this case 200.00 i.e. when multiplied by 100 is 20000 (0x4E20). (LSB,MSB)
01 – paying type. In this case 1 – credit card.
00 AC - CRC (sum of all bytes except STX, written in two bytes MSB, LSB)

Printer response: 06 08 02 02 7F 00 00 81

06 - ACK response that command is read correctly - CRC correct.
08 - WAIT byte. Execution of given command takes some time, so this message is returned to PC.
02 - STX start of communication.
02 - LEN (length of DATA)
7F - CMD command. Command which returns execution status.

00 - STATUS error message byte (in this case there is no error in execution – 00; for error codes, check error code table at the end of this document).
00 81 - CRC (sum of all bytes except STX, written in two bytes MSB, LSB)

PC: 06

06 - ACK response that command is read correctly - CRC correct.

Example 2: Automatic receipt closure.

PC: 02 0A 33 00 00 00 00 00 00 00 00 00 00 3D

02 - STX start of communication
0A - LEN (length of DATA)
33 - CMD command
00 00 00 00 00 00 00 00 - QUANTITY. In this case 0, bill is closed automaticaly. (LSB,MSB)
00 – paying type. In this case 0 - cash.
00 3D - CRC (sum of all bytes except STX, written in two bytes MSB, LSB)

Printer response: 06 08 02 02 7F 00 00 81

06 - ACK response that command is read correctly - CRC correct.
08 - WAIT byte. Execution of given command takes some time, so this message is returned to PC.
02 - STX start of communication.
02 - LEN (length of DATA)
7F - CMD command. Command which returns execution status.
00 - STATUS izvršenja komande. U ovom slučaju - uspešno izvršena komanda.
00 81 - CRC (sum of all bytes except STX, written in two bytes MSB, LSB)

PC: 06

06 - ACK response that command is read correctly - CRC correct.

Command for reading bill state 56 (0x38).

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – start of communication
LEN – command length

CMD - command
CRC – check sum

Printer response:

[STX<1>][LEN<1>][CMD<1>][SUBSUM <8>][TOTAL <8>][NUM_ITEMS <4>][PAYED 0 <8>][PAYED 1 <8>][PAYED 2 <8>][BILL_NUMBER <4>][CASHIER <1>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
SUBSUM – difference between total bill value and sum of all applied payments. Value is long number obtained from real value multiplied by 100 (real value has 2 decimal spaces).
TOTAL – total bill value. Value is long number obtained from real value multiplied by 100 (real value has 2 decimal spaces).
NUM_ITEMS – total number of sold items on bill
PAYED 0 – sum of all payments made by cash. Value is long number obtained from real value multiplied by 100 (real value has 2 decimal spaces).
PAYED 1 - sum of all payments made by credit card.Value is long number obtained from real value multiplied by 100 (real value has 2 decimal spaces).
PAYED 2 - sum of all payments made by cheque. Value is long number obtained from real value multiplied by 100 (real value has 2 decimal spaces).
BILL_NUMBER – ordinal bill number.
CASHIER – cashier number. If cashier is not logged in or cashier option is not used, this value is -1(0xFF).
CRC – check sum

Example 1: Reading bill state.

PC: 02 01 38 00 39

02 - STX start of communication
01 - LEN (length of DATA)
38 - CMD command
00 39 - CRC (sum of all bytes except STX, written in two bytes MSB, LSB)

Printer response: 06 02 32 38 D0 07 00 00 00 00 00 00 88 13 00 00 00 00 00 00 02 00 00 00 D0 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 E8 03 00 00 00 00 00 00 0B 00 00 00 FF 04 AA

06 – ACK response that command is read correctly - CRC correct.
02 - STX start of communication.
32 - LEN (length of DATA). In this case 50 (0x32)

38 - CMD command.
D0 07 00 00 00 00 00 00 - SUBSUM. In this case 20.00 i.e. 2000 (0x7D0) .(LSB,MSB)
88 13 00 00 00 00 00 00 - TOTAL. In this case 50.00 i.e. 5000 (0x1388). (LSB,MSB)
02 00 00 00 – NUM_ITEMS. In this case 2.
D0 07 00 00 00 00 00 00 – payed by cash. In this case 20.00 i.e. 2000 (0x7D0) .(LSB,MSB)
00 00 00 00 00 00 00 00 – payed by credit card. In this case 0 .(LSB,MSB)
E8 03 00 00 00 00 00 00 – payed by cheque. In this case 10.00 i.e. 1000 (0x3E8) .(LSB,MSB)
0B 00 00 00- bill number. In this case 11 (0xB).
FF - CASHIER. In this case, cashier is not used.
04 AA - CRC (sum of all bytes except STX, written in two bytes MSB, LSB)

PC: 06

06 - ACK response that command is read correctly - CRC correct.

Command for reading bill item 57 (0x39).

[STX<1>][LEN<1>][CMD<1>][ITEM_NUM <4>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
ITEM_NUM – ordinal number of item sold on bill. Indexing starts from 0.
CRC – checck sum

Printer response:

[STX<1>][LEN<1>][CMD<1>][CODE<4>][QUANTITY<4>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
CODE – article code. Integer in range from 1 to 75000 packed in 4 bytes.
QUANTITY - Value of sold idem quantity. Value is sent as integer number which value is obtained from real value multiplied by 1000.
CRC – check sum

Example 1: Reading bill item.

PC: 02 05 39 01 00 00 00 00 3F

02 - STX start of communication
05 - LEN (length of DATA)
39 - CMD command
01 00 00 00 – Item ordinal number. In this case 1, second item on bill.
00 3F - CRC (sum of all bytes except STX, written in two bytes MSB, LSB)

Printer response: 06 02 09 39 02 00 00 00 D0 07 00 00 01 1B

06 - ACK response that command is read correctly - CRC correct.
02 - STX start of communication.
09 - LEN (lenfgth of DATA).
39 - CMD command.
02 00 00 00 – article CODE. In this case 2. (LSB,MSB)
D0 07 00 00- QUANTITY. In this case 20.00 i.e. 2000 (0x7D0). (LSB,MSB)
01 1B - CRC (sum of all bytes except STX and ACK, written in two bytes MSB, LSB)

PC: 06

06 - ACK response that command is read correctly - CRC correct.

Command for reading daily state 86 (0x56).

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
CRC – check sum

Printer response:

[STX<1>][LEN<1>][CMD<1>][DAILY_REPORT_NUM <4>] 9*{[VAT_TURNOVER <8>]} [PAYED 0 <8>][PAYED 1 <8>][PAYED 2 <8>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
DAILY_REPORT_NUM – ordinal .
VAT_TURNOVER – total turnover by each VAT during one fiscal day. Value is long number, obtained from real value multiplied by 100.
PAYED0 – sum of all payments by cash during a fiscal day. Value is long number obtained from real value multiplied by 100 (real value has 2 decimal spaces).
PAYED 1 - sum of all payments by credit card during a fiscal day. Value is long number

obtained from real value multiplied by 100 (real value has 2 decimal spaces).
PAYED 2 - sum of all payments by cheque during a fiscal day. Value is long number obtained from real value multiplied by 100 (real value has 2 decimal spaces).
CRC – check sum

Example 1: Reading daily state.

PC: 02 01 56 00 57

02 - STX start of communication
01 - LEN (length of DATA)
56 - CMD command
00 57 - CRC (sum of all bytes except STX, written in two bytes MSB, LSB)

Printer response: 06 02 65 56 01 00 00 00 B0 BB 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 56 22 00 00 00 00 00
00 28 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0E A1 00 00 00 00
00 00 20 4E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 F5

06 – ACK response that command is read correctly - CRC correct.
02 - STX start of communication.
65 - LEN (length of DATA). In this case 101 (0x65)
56 - CMD command.
01 00 00 00 – Daily report number. In this case 1.
B0 BB 00 00 00 00 00 00 – turnover by VAT - A (index 0) In this case 480.48 i.e. 48048 (0xBBB0).(LSB,MSB)
.......this section repeats another 8 times for each VAT (VAT have signs: А, Г, Д, Ђ, Е, Ж, И, Ј, К).
0E A1 00 00 00 00 00 00 – payed by cash. In this case 412.30 i.e. 41230 (0xA10E) .(LSB,MSB)
20 4E 00 00 00 00 00 00 – payed by card. In this case 200.00 i.e. 20000 (0x4E20).(LSB,MSB)
00 00 00 00 00 00 00 00 – payed by cheque. In this case 0 .(LSB,MSB)
03 F5 - CRC (sum of all bytes except STX and ACK, written in two bytes MSB, LSB)

PC: 06

06 - ACK response that command is read correctly - CRC correct.

Commands for VAT programming:

NOTE:

VAT value is sent as a whole number obtained from real value multiplied by 100. If you want to program undefined VAT rate, just send 0xFFFF for value. Changing VAT rates is available only after daily report has been executed.

VAT rate index starts from 0 and ends with 8 (VAT1, VAT2...VAT9).

Command for VAT programming 31 (0x1F).

[STX<1>][LEN<1>][CMD<1>][VAT1<2>][VAT 2<2>][VAT 3<2>][VAT 4<2>][VAT 5<2>][VAT 6<2>][VAT 7<2>][VAT 8<2>][VAT 9<2>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
VAT1 – VAT rate (A)
VAT 2 - VAT rate (Г)
VAT 3 - VAT rate (Д)
VAT 4 - VAT rate (Ђ)
VAT 5 - VAT rate (E)
VAT 6 - VAT rate (Ж)
VAT 7 - VAT rate (И)
VAT 8 - VAT rate (J)
VAT 9 - VAT rate (K)
CRC – check sum

Example 1: VAT rate programming. Following VAT rates will be programmed A = 0.00, Ђ =18.00 and E=8.00.

PC: 02 13 1f 00 00 ff ff ff ff 08 07 20 03 ff ff ff ff ff ff ff ff 0c 58

02 - STX , start of communication
13 - LEN , command length (13hex=19dec)
1F - CMD, command for VAT programming (1Fhex =31dec)
00 00 - PS1, VAT rate (A) with value 0.00
ff ff - PS2, VAT rate (Г) value not defined
ff ff - PS3, VAT rate (Д) value not defined

08 07 - PS4, VAT rate (Ђ); (0708hex=1800dec) with value 18.00
20 03 - PS5, VAT rate (E); (0320hex = 800dec) with value 8.00
ff ff - PS6, VAT rate (Ж) value not defined
ff ff - PS7, VAT rate (И) value not defined
ff ff - PS8, VAT rate (J) value not defined
ff ff - PS9, VAT rate (K) value not defined
0c 58 - CRC, (sum of all bytes except STX, written in two bytes MSB, LSB)
(13+1f+00+00+ff+ff+ff+ff+08+07+20+03+ff+ff+ff+ff+ff+ff+ff+ff)hex

Printer response: 06 02 02 7f 00 00 81

06 - ACK, ACK response that command is read correctly - CRC correct
02 - STX , start of communication
02 - LEN , command length
7f 00 – command executed successfuly
00 81 -CRC (7f+02)

Command for reading VAT values 32 (0x20).

PC:

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
CRC – check sum

Printer response:

[STX<1>][LEN<1>][CMD<1>][VAT1<2>][VAT 2<2>][VAT 3<2>][VAT 4<2>][VAT 5<2>][VAT 6<2>][VAT 7<2>][VAT 8<2>][VAT 9<2>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
VAT1 – VAT rate (A)
VAT 2 - VAT rate (Г)
VAT 3 - VAT rate (Д)
VAT 4 - VAT rate (Ђ)
VAT 5 - VAT rate (E)
VAT 6 - VAT rate (Ж)
VAT 7 - VAT rate (И)
VAT 8 - VAT rate (J)

VAT 9 - VAT rate (K)
CRC – check sum

Example 2: Reading VAT values.

PC: 02 01 20 00 21

02 - STX , start of communication
01- LEN , command length
20 - CMD, command for reading VAT values
00 21 - CRC , check sum (20+1=21)

Printer response: 06 02 13 20 00 00 ff ff ff ff 08 07 20 03 ff ff ff ff ff ff ff ff 0c 59

06- ACK response that command is read correctly
02- STX
13- LEN
20 -CMD
00 00 - VAT1, value for VAT A= 0,00
ff ff - VAT2, value for VAT (not definied) Г= *,**
ff ff - VAT3, value for VAT (not definied) Д= *,**
08 07 - VAT4, value for VAT (0708hex =1800) Ђ= 18,00
20 03 - VAT5, value for VAT (0320hex = 800) E = 8,00
ff ff - VAT6, value for VAT (not definied) Ж= *,**
ff ff - VAT7, value for VAT (not definied) И= *,**
ff ff - VAT8, value for VAT (not definied) J= *,**
ff ff - VAT9, value for VAT (not definied) K= *,**
0c 59 - CRC, check sum (13+20+00+00+ff+ff+ff+ff+08+07+20+03+ff+ff+ff+ff+ff+ff+ff+ff)
hex

Commands for measure unit programming:

NOTE:

Measure unit name is sent as an array of 2 bytes, represented by character table described at the end of this document. There are 15 measure units supported in printer. Indexing starts from 0 (first measure unit is pcs). First 10 measure units cannot be changed (they have fixed names). Measure units from 10 to 15 can be changed and programmed in any name you need. Programming new measure units can be done after daily report is executed.

Command for programming measure units 62 (0x3E).

[STX<1>][LEN<1>][CMD<1>][MU10<2>][MU 11<2>][MU 12<2>][MU 13<2>][MU 14<2>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
MU10 – measure unit name
MU11 - measure unit name
MU12 - measure unit name
MU13 - measure unit name
MU14 - measure unit name
CRC – check sum

Example 1: Measure unit programming. Measure unit MU10 will be defined as "kw".

PC: 02 0b 3e 6b 77 20 20 20 20 20 20 20 20 02 2b

02 - STX , start communication
ob- LEN , command length (ob)hex = 11dec
3e – CMDcommand for measure unit programming
6b 77 - MU10, measure unit description : (6b)hex = 107dec in character table at the end of this document, where you can see is defined as"k"; (77)hex = 119dec in table defined as "w"
20 20 - MU11 , not defined value
20 20 - MU12 , not defined value
20 20 - MU13 , not defined value
20 20 - MU14 , not defined value
02 2b - CRC ,check sum (0b+3e+6b+77+20+20+20+20+20+20+20+20) hex

Printer response: 06

06 - ACK response that command is read correctly - CRC correct

Command for reading measure units 61 (0x3D).

PC:
[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
CRC - checksum

Printer response:

[STX<1>][LEN<1>][CMD<1>][MU0 <2>][MU 1 <2>][MU 2 <2>][MU 3 <2>][MU4 <2>][MU5 <2>][MU6 <2>][MU7 <2>][MU8 <2>][MU9 <2>][MU10 <2>][MU11 <2>][MU12 <2>][MU13 <2>][MU14 <2>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
MU0 –first measure unit in device, cannot be changed
MU1 - second measure unit in device, cannot be changed
MU2 - third measure unit in device, cannot be changed
MU3 - fourth measure unit in device, cannot be changed
MU4 - fifth measure unit in device, cannot be changed
MU5 - sixth measure unit in device, cannot be changed
MU6 - seventh measure unit in device, cannot be changed
MU7 - eighth measure unit in device, cannot be changed
MU8 - ninth measure unit in device, cannot be changed
MU9 - tenth measure unit in device, cannot be changed
MU10 - eleventh measure unit in device, can be changed
MU11 - twelveth measure unit in device, can be changed
MU12 - thirteenth measure unit in device, can be changed
MU13 - fourteenth measure unit in device, can be changed
MU14 - fifteenth measure unit in device, can be changed

ff ff -*
CRC – check sum

Example 2: Reading measure units.

PC: 06 02 01 3d 00 3e

06 - ACK, response that command is read correctly
02 - STX , start of communication
01- LEN , command length
3d - CMD, command for measure unit reading
00 3e - CRC , check sum (01+3d)hex

Printer response: 06 02 21 3d 20 20 6b 67 67 20 74 20 6c 20 64 6c 6d 20 6d 32 6d 33 68 20 6b 77 20 20 20 20 20 20 20 20 ff ff 0a 1b

06 - ACK, response that command is read correctly
02 - STX , start of communication
21 - LEN, command length
3d - CMD, command for measure unit reading
20 20 - MU0, first measure unit in device - pcs
6b 67 - MU1 , second measure unit in device - kg
67 20 - MU2 , - third measure unit in device - g
74 20 - MU3 , fourth measure unit in device - t
6c 20 - MU4 , fifth measure unit in device - l
64 6c - MU5, sixth measure unit in device - dl
6d 20 - MU6, seventh measure unit in device - m
6d 32 - MU7, eighth measure unit in device - m2
6d 33 - MU8, ninth measure unit in device - m3
68 20 - MU9, tenth measure unit in device - h
6b 77 - MU10, eleventh measure unit in device (previously programmed) - kw
20 20 - MU11, twelveth measure unit in device –not defined
20 20 - MU12, thirteenth measure unit in device–not defined
20 20 - MU13, fourteenth measure unit in –not defined

20 20 - MU14, fifteenth measure unit in –not defined
ff ff -*
0a 1b - CRC, check sum

* Firmware version 11.23 has a bug when reading measure units. Last two bytes FF FF can be ignored.

Commands related to programming the cashier:

NOTE:
The printer may have information about the 16 cashier who has a code and three levels of access. The lowest code that can be defined is 100 (three digits) and the maximum can be 9 digits. The lowest level is only the possibility of sale, price change and enter a new article. Medium has the ability to print reports while the highest level of access opens all the functions at the device. The option of using cashier and memory sold at Kasiri must be defined in the settings. The printer can not be enforced lock cashier. The device must have at least one cashier with the highest level of access.
Cashiers are defined from 0 - 15.

Command for programming cashiers is 40 (0x28).

[STX<1>][LEN<1>][CMD<1>][CASHIER NUMBER<1>][LEVEL<1>][CASHIER NAME <22>][CODE<8>][CRC<2>]

STX –start of communication
LEN – command length
CMD - command
CASHIER NUMBER - unique cashier number, which is defined from 0 to 15
LEVEL – access level (low -00, medium-01 or high-02)
CASHIER NAME – defined cashier name , the character in the text are coded by the table at the end of the instructions.
CODE – unique cashier code
CRC – control check sum

Example 1: Programming the cashier.

Define the cashiers with the name „Cashier“(on Serbian: „Kasir“), code „1111“, the level od access „hi“ and another cashier with the name „Cashier 1“, the code „100“ and the level of access „medium“. When defining more cashiers each cashier must be separately programmed.

Computer: 02 21 28 00 02 4b 61 73 69 72 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 57 04 00 00 00 00 00 00 04 c0

02 - STX

21-LEN

28-CMD

00 – CASHIER NUMBER, defines the first cashier

02- LEVEL , access level cashier 02- hi ie. the highest access level

4b 61 73 69 72 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 –CASHIER NAME , ima kasira (4b)hex=75dec in table of caracters it is "K"; 61hex = 97dec ie caracter"a"; 73hex=115dec "s"; 69hex=105dec "i";72hex =114dec "r", others are 20 to indicates empty places, ie undefined field. Cashier name can be maximum of 22 caracters.

57 04 00 00 00 00 00 00 - CODE, unique cashier code. 00 00 00 00 00 00 04 57 ie 457hex = 1111dec

04 c0 - CRC, control check sum

Printer: 06 02 02 7f 00 00 81

06 - ACK

02 - STX

02 - LEN

7f 00 – command successfully executed

00 81 - CRC

Computer: 06 02 21 28 01 01 4b 61 73 69 72 20 32 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 64 00 00 00 00 00 00 00 04 db

06-ACK

02 - STX

21 - LEN

28 - CMD

01 – CASHIER NUMBER, defines a second cashier

01- LEVEL , access level cashier 01- medium

4b 61 73 69 72 20 32 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 –CASHIER NAME , (4b)hex=75dec in table of caracters it is "K"; 61hex = 97dec ie caracter "a"; 73hex=115dec "s"; 69hex=105dec "i";72hex =114dec "r", 20 empty place; 32hex = 50 ie "2"others are 20 to indicates empty places ie undefined empty field. Cashier name can be maximum of 22 caracters.

64 00 00 00 00 00 00 00 – CODE, unique cashier code. 00 00 00 00 00 00 00 64 ie 457hex = 100dec

04 db - CRC, control check sum

Printer: 06 02 02 7f 00 00 81

06 - ACK

02 - STX

02 - LEN

7f 00 – command successfully executed

00 81 - CRC

Command for reading the cashier 38 (0x26).

The computer sends a block in form:

[STX<1>][LEN<1>][CMD<1>][CASHIER NUMBER<1>][CRC<2>]

STX – start of communication

LEN – command length

CMD - command

CASHIER NUMBER – unique cashier number, which is defined from 0 to 15

CRC – control check sum

The printer return the block in form:

[STX<1>][LEN<1>][CMD<1>][CASHIER NUMBER<1>][LEVEL<1>][CASHIER NAME<1>][CODE<8>][SALES BY CASHIER<8>][CRC<2>]

STX – start of communication

LEN – command length

CMD - command

CASHIER NUMBER- unique cashier number, which is defined from 0 to 15

LEVEL – access level (low -00, medium-01 or high-02)

CASHIER NAME – defined cashier name , the character in the text are coded by the table at the end of the instructions.

CODE – unique cashier code

SALE BY CASHIER - overall turnover cashier realized during the period when active, multiplied by 100

CRC – control check sum

Example 2: Reading cashier.

Computer: 02 02 26 00 00 28

02 - STX

02 - LEN

26 - CMD, command reserver for reading cashier from the printer

00 – CASHIER NUMBER, cashier number, zero ie first

00 28 - CRC, control check sum

Printer: 06 02 29 26 00 02 4b 61 73 69 72 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 57 04 00 00 00 00 00 00 86 83 a1 c2 00 00 00 00 07 32

06 - ACK

02 - STX

29 - LEN

26 - CMD

00- CASHIER NUMBER , in this case the first cashier

02 - LEVEL , access level cashier is 02-hi, highest access

4b 61 73 69 72 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 – CASHIER NAME, see explanation when cashier defining.

57 04 00 00 00 00 00 00 - CODE, unique cashier code. (04 57)hex = 1111dec

86 83 a1 c2 00 00 00 00 – SALE BY CASHIER ,(c2 a1 83 86)hex = 3265364870 /100 = 32653648,70

04 c6 - CRC

Computer: 06 02 02 26 01 00 29

06 - ACK

02 - STX

02 - LEN

26 - CMD, command reserved for reading cashier from the printer

01 – CASHIER NUMBER, cashier number, second cashier

00 29 - CRC, control check sum

Printer: 06 02 29 26 01 01 4b 61 73 69 72 20 32 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 64 00 00 00 00 00 00 00 e8 13 14 6f 00 00 00 00 06 5f

06 - ACK

02 - STX

29 - LEN

26 - CMD

01- CASHIER NUMBER , in this case, is the second cashier

01 - LEVEL , access level cashier is 01- medium

4b 61 73 69 72 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 – CASHIER NAME, see explanation when cashier defining.

64 00 00 00 00 00 00 00 - CODE, unique cashier code. (64)hex = 100dec

e8 13 14 6f 00 00 00 00 – SALE BY CASHIER ,(6f 14 13 e8)hex = 1863586792 /100 = 18635867,92

06 5f - CRC, control check sum

Computer: 06 02 02 26 02 00 2a

06 - ACK

02 - STX

02 - LEN

26 - CMD, command reserved for reading cashier from the printer

02 – CASHIER NUMBER, cashier number, third cashier

00 2a - CRC, control check sum

Printer: 06 02 29 26 02 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 3b

06 - ACK

02 - STX

29 - LEN

26 - CMD

02- CASHIER NUMBER, in this case is the third cashier

00 - LEVEL , cashier access level is je 00-low, lowest level

ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff – CASHIER NAME, undefined cashier

00 00 00 00 00 00 00 00 - CODE, unique cashier code -undefined

00 00 00 00 00 00 00 00 – SALE BY CASHIER - undefined

16 3b - CRC, control check sum

Computer: 06 02 02 26 03 00 2b

06 - ACK

02 - STX

02 - LEN

26 - CMD, command reserved for reading cashier from the printer

03 – CASHIER NUMBER, ordinal cashier number , fourth cashier

00 2b - CRC, control check sum

Printer: 06 02 29 26 03 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 3c

Computer: 06 02 02 26 04 00 2c

Printer: 06 02 29 26 04 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 3d

Computer: 06 02 02 26 05 00 2d

Printer: 06 02 29 26 05 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 3e

Computer: 06 02 02 26 06 00 2e

Printer: 06 02 29 26 06 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 3f

Computer: 06 02 02 26 07 00 2f

Printer: 06 02 29 26 07 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 40

Computer: 06 02 02 26 08 00 30

Printer: 06 02 29 26 08 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 41

Computer: 06 02 02 26 09 00 31

Printer: 06 02 29 26 09 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 42

Computer: 06 02 02 26 0a 00 32

Printer: 06 02 29 26 0a 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 43

Computer: 06 02 02 26 0b 00 33

Printer: 06 02 29 26 0b 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 44

Computer: 06 02 02 26 0c 00 34

Printer: 06 02 29 26 0c 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 45

Computer: 06 02 02 26 0d 00 35

Printer: 06 02 29 26 0d 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 46

Computer: 06 02 02 26 0e 00 36

Printer: 06 02 29 26 0e 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 47

Computer: 06 02 02 26 0f 00 37

Printer: 06 02 29 26 0f 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 16 48

Komanda za prijavu kasira je 42 (0x2A).

[STX<1>][LEN<1>][CMD<1>][CODE <4>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
CODE – defined cashier code, code for the login
CRC – control check sum

Example 3: Login cashier.

Computer: 02 05 2a 57 04 00 00 00 8a

02 - STX

05 - LEN

2a - CMD

57 04 00 00 - CODE , (04 57) hex = 1111dec

00 8a - CRC

Printer: 06 02 02 7f 00 00 81

06 - ACK, confirmation of the successful execution of the command
02 - STX , start of communication
02 - LEN , command length
7f 00 – successfully executed command
00 81 -CRC (7f+02)

Commmands related to fiscalization:

NOTE:
Reset, time setting and technical inspection require the presence of a jumper. When it is reset, jumper must be present at printer startup, while for adjusting time and technical inspection it is enough to set jumper just before sending commands. If the printer is started again with a jumper a reset must be made. Time changes from winter to daylight saving, the printer performs automatically.

The command for time setting 1 (0x01).

The computer sends a block in form:

[STX<1>][LEN<1>][CMD<1>][TIME IN MILLISECONDS<8>][CRC<2>]

STX – start of communication
LEN – length of DATA sequence
CMD - command
TIME IN MILLISECONDS – Time is sent as a number of milliseconds from 1.1.2000 GMT.
CRC – control check sum (the sum of all bytes except STX)

Example 1: Time setting

Computer: 02 09 01 C8 CF 3C 7D 59 00 00 00 02 B3

02 - STX
09 - LEN (length of DATA sequence)
01 - CMD (komanda)
C8 CF 3C 7D 59 00 00 00- TIME IN MILLISECONDS (LSB,MSB)
02 B3 - CRC (the sum of all bytes except STX, located in 2 bytes MSB,LSB)

Printer: 06 02 02 7F 00 00 81

06 - ACK
02 - STX
02 - LEN (length of DATA sequence)
7F 00 – Command successfully executed
00 81 - CRC (the sum of all bytes except STX i ACK, located in 2 bytes MSB,LSB)

Command for reading time 2 (0x02).

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – start of communication
LEN – length of DATA sequence
CMD - command
CRC – control check sum (sum of all bytes except STX)

After sending ACK printer returns the block in form:

[ACK<1>][STX<1>][LEN<1>][CMD<1>][TIME IN MILLISECONDS<8>][CRC<2>]

TIME IN MILLISECONDS - Time is sent as a number of milliseconds from 1.1.2000 GMT.

Example 2: Reding time

Computer: 02 01 02 00 03

02 - STX
01 - LEN (length of DATA sequence)
02 - CMD
00 03 - CRC (sum of all bytes except STX, located in 2 bytes MSB, LSB)

Pritner: 06 02 09 02 8E 09 42 7D 59 00 00 00 01 BA

06 - ACK
02- STX
09 - LEN
02 - CMD
8E 09 42 7D 59 00 00 00 – TIME IN MILLISECONDS (LSB, MSB)
01 BA - CRC (sum of all bytes except STX i ACK, located in 2 bytes MSB, LSB)

Command for sending PIB 7 (0x07)

The computer sends a block in form:

[STX<1>][LEN<1>][CMD<1>][PIB<9>][CRC<2>]

STX – start of communication
LEN – length of DATA sequence
CMD - command
PIB – tax identification number of device
CRC – control check sum (sum of all bytes except STX)

PIB is sent as a sequence of bytes that indicate figures from the table of caracters (ASCII digits). This is followed by sending command for fiscalization: CMD=4 (0x04). During fiscalization, the printer is interrupted serial communication.

Example 3: Send PIB

Computer: 02 0A 07 31 32 33 34 35 36 37 38 39 01 EE

02 - STX
0A - LEN
07 - CMD (first sends a command for PIB)
31 32 33 34 35 36 37 38 39 - PIB (see the character table at the end of the instructions)
01 EE - CRC (sum of all bytes except STX, located in 2 bytes MSB, LSB)

Printer: 06 08 08 08 08 08 08 08 08 08 08 08 08 02 02 7F 00 00 81

06 - ACK
08 - WAIT
02 - STX
02 - LEN
7F 00 – Command successfully executed
00 81 - CRC (sum of all bytes except STX i ACK, located in 2 bytes MSB, LSB)

Command for fiscalization 4 (0x04).

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

Example 4: sending command for fiscalization.

Computer: 06 02 01 04 00 05

06 - ACK
02 - STX
01 - LEN
04 - CMD (command for fiscalization)
00 05 - CRC (sum of all bytes except STX i ACK, located in 2 bytes MSB, LSB)

Printer: 06 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08

06 - ACK
08 - WAIT

Command for reset 5 (0x05).

The computer sends a block in form:

[STX<1>][LEN<1>][CMD<1>][TYPE<1>][CRC<2>]

STX – start of communication
LEN – length of DATA sequence
CMD - command
TYPE - type reset
CRC – control check sum (sum of all bytes except STX)

TYPE reset is:

- 1 for type P
- 3 for type C

Reset P deletes only the RAM memory, while C clears the RAM i FLASH memory with the items.
Because the printer some items currently stored in RAM, it is possible to lose some items by doing reset P.

Example 5: Reset P

Computer: 02 02 05 01 00 08

02 - STX
02 - LEN
05 - CMD
01 – Reset TYPE (for reset P, TYPE = 1)
00 08 - CRC control check sum (sum of all bytes except STX)

Printer: 06 08 08 08 08 08 08 08 08 08 08 02 02 7F 00 00 81

06 - ACK
08 - WAIT
02 - STX
02 - LEN

7F 00 – Command successfully executed

00 81 - CRC (sum of all bytes STX i ACK, located in 2 bytes MSB, LSB)

Example 6: Reset C

Computer: 02 02 05 03 00 0A

02 - STX
02 - LEN
05 - CMD
03 – Reset TYPE (for reset C, TYPE = 3)
00 0A- CRC control check sum (sum of all bytes except STX)

Pritner: 06 08 08 08 08 08 08 08 08 08 08 08 02 02 7F 00 00 81

06 - ACK
08 - WAIT
02 - STX
02 - LEN
7F 00 – command successfully executed
00 81 - CRC (sum of all bytes except STX i ACK, located in 2 bytes MSB, LSB)

Command for reading data about fiscalization 3 (0x03).

The computer sends a block in form:

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – start of fiscalization
LEN – length of DATA sequence
CMD - command
CRC – control check sum (sum of all bytes except STX)

The printer returns the block in form:

[ACK<1>][STX<1>][LEN<1>][CMD<1>][TIME FISCALIZATION <8>][IBFM<8>][PIB<9>][NUMBER OF DAILY REPORTS <4>]

[NUMBER OF RESETS <4>][NUMBER OF CHANGE OF TAX RATES <4>][NUMBER OF TECHNICAL INSPECTIONS <4>][CRC<2>]

TIME FISCALIZATION: Time fiscalization as the number of milliseconds 1.1.2000 GMT.
IBFM: IBFM the number of register as a string of bytes that indicate the characters from the character table.
PIB: PIB user as a sequence of bytes that indicate the characters from the character table (ASCII).
NUMBER OF DAILY REPORTS: the total number of daily reports (LSB, MSB)
NUMBER OF RESETS: the total number of resets (LSB, MSB)
NUMBER OF CHANGE OF TAX RATES: the total number of changes in tax rates (LSB, MSB)
NUMBER OF TECHNICAL INSPECTIONS: the total number of technical inspections (LSB, MSB)

Example 7: Reading data on fiscalization

Fiscalization device with the following fiscal data - IBFM: XX123456, PIB: 123456789

Computer: 02 01 03 00 04

02 - STX
01 - LEN
03 - CMD
00 04 - CRC (sum of all bytes except STX, located in 2 bytes MSB, LSB)

Printer: 06 02 2A 03 10 52 51 E8 35 01 00 00 58 58 31 32
33 34 35 36 31 32 33 34 35 36 37 38 39 00 00 00
00 02 00 00 00 00 00 00 00 01 00 00 00 05 C3

06 - ACK
02 - STX
2A - LEN
03 - CMD
10 52 51 E8 35 01 00 00 – TIME FISCALIZATION (LSB, MSB)
58 58 31 32 33 34 35 36 - IBFM (one after the other character by character IBFM based on the table at the end of the instructions)
31 32 33 34 35 36 37 38 39 - PIB (one after the other character by character IBFM based on the table at the end of the instructions)
00 00 00 00 - the number of daily reports (LSB, MSB)
02 00 00 00 - number of resets (LSB, MSB)

00 00 00 00 - number of changes in tax rates (LSB, MSB)
01 00 00 00 - number of technical inspections (LSB, MSB)
05 C3 - CRC (sum of all bytes except STX i ACK, located in 2 bytes MSB, LSB)

Command for executing technical inspection 67 (0x43).

The computer sends a block in form:

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – start of communication
LEN – length of DATA sequence
CMD - command
CRC – control check sum (sum of all bytes except STX, located in 2 bytes MSB, LSB)

Example 8: Execution of technical inspection

Computer: 06 02 01 43 00 44

06 - ACK
02 - STX
01 - LEN
43 - CMD
00 44 - CRC (sum of all bytes except STX i ACK, located in 2 bytes MSB, LSB)

Printer: 06 02 02 7F 00 00 81

06 - ACK
02 - STX
02 - LEN
7F 00 – Command successfully executed
00 81 - CRC (sum of all bytes except STX i ATX, located in 2 bytes MSB, LSB)

The command to print all technical inspections 66 (0x42).

The computer sends a block in form:

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – start of communication
LEN – length of DATA sequence
CMD - command

CRC – control chek sum (sum of all bytes except STX i ATX, located in 2 bytes MSB, LSB)

Example 9: Printing all technical inspections

Computer: 02 01 42 00 43

02 - STX
01 – LEN

42- CMD
00 43 - CRC (sum of all bytes except STX i ATX, located in 2 bytes MSB, LSB)

Printer: 06 02 02 7F 00 00 81

06 - ACK
02 - STX
02 - LEN
7F 00 - command successfully executed
00 81 - CRC (sum of all bytes except STX i ATX, located in 2 bytes MSB, LSB)

Command for control device:

The command for printing on external display is 33 (0x21).

The computer sends a block in form:

[SOH<1>][LEN1<1>][LEN2<1>][CMD<1>][UPPER ROW<20>][LOWER ROW <20>][CRC<2>]

SOH- start of communication (long command)
LEN1 - lower byte LEN
LEN2 - high byte LEN
CMD - command
UPPER ROW - a string of characters that are displayed in the top row
LOWER ROW - a string of characters that are displayed in the lower row
CRC – control check sum (sum of all bytes except SOH)

Example: Displaying text DISPLAY TEST

Computer: 03 29 00 21 54 45 53 54 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 44 49 53 50 4C 45 4A 41 20 20 20 20 20 20 20 20 20 20 20 20 07 56

03 - SOH
29 - LEN1
00 - LEN2
21 - CMD
54 45 53 54 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 - UPPER ROW ("TEST")
44 49 53 50 4C 45 4A 41 20 20 20 20 20 20 20 20 20 20 20 20 – LOWER ROW ("DISPLAY")
07 56 - CRC control check sum (sum of all bytes except SOH)

Printer: 06 02 02 7F 00 00 81

06 - ACK
02 - STX
02 - LEN
7F 00 - confirmation of the successful completion of command
00 81 - CRC

The command for opening the drawer is 36 (0x24).

The computer sends a block in form:

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – start of communication
LEN – length of DATA sequence
CMD - command
CRC – control check sum (sum of all bytes except STX)

Example:

Computer: 02 01 24 00 25

02 - STX
01 - LEN
24 - CMD
00 25 - CRC control check sum (sum of all bytes except STX)

Printer: 06 08 02 02 7F 00 00 81

06 - ACK
08 - WAIT
02 - STX
02 - LEN
7F 00 - confirmation of the successful completion of command
00 81 - CRC

Command for sound signal is 34 (0x22).

The computer sends a block in form:

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – start of communication
LEN – length of DATA sequence
CMD - command
CRC – control check sum (sum of all bytes except STX)

Example:

Computer: 02 01 22 00 23

02 - STX
01 - LEN
22 - CMD
00 23 - CRC

Printer: 06 02 02 7F 00 00 81

06 - ACK
02 - STX
02 - LEN
7F 00 - confirmation of the successful completion of command
00 81 - CRC

The command for cutting the paper is 27 (0x1B).

The computer sends a block in form:

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – start of communication
LEN – length of DATA sequence
CMD - command
CRC – control check sum (sum of all bytes except STX)

Example:

Computer: 02 01 1B 00 1C

02 - STX
01 - LEN
1B - CMD
00 1C - CRC

Printer: 06 08 08 02 02 7F 00 00 81

06 - ACK
08 - WAIT
02 - STX
02 - LEN

7F 00 - confirmation of the successful completion of command
00 81 – CRC

The command for turning printhead is 28 (0x1C).

The computer sends a block in form:

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – start of communication
LEN – length of DATA sequence
CMD - command
CRC – control check sum (sum of all bytes except STX)

Example:

Computer: 02 01 1C 00 1D

02 - STX
01 - LEN
1B - CMD
00 1C - CRC

Printer: 06 02 02 7F 00 00 81

06 - ACK
02 - STX
02 - LEN
7F 00 - potvrda o uspešno obavljenoj komandi
00 81 - CRC

The command to display the subtotal is 29 (0x1D).

The computer sends a block in form:

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – start of communication
LEN – length of DATA sequence
CMD - command

CRC – control check sum (sum of all bytes except STX)

Example:

Computer: 02 01 1D 00 1E

02 - STX
01 - LEN
1D - CMD
00 1E - CRC

Printer: 06 02 01 1D 00 1E

06 - ACK
02 - STX
02 - LEN
1D - vraća kod naredbe koju je izvršio*
00 1E- CRC

*Note: In software version 11.23, only with command for display the subtotal, the printer returns code of command for display the subtotal 1D instead 7F 00 to confirm the successful completion of command.

Long commands:

For certain operations with the device, mainly when programming large amount of data, it is necessary to send a larger block of data at once in order to gain time during communication. For this purpose, we use the long command. Most often they are used in programming items.

Each command has a standardized format and variable length. Controls are the following forms:

SOH	LEN1	LEN2	DATA	CRC

SOH: 0x03
LEN1 i LEN2 : length DATA sequence (**DATA_LEN**=**LEN2**<<8|**LEN1** - LSB,MSB) no larger than 512.
CRC: sum of all bytes except **SOH** and **WAIT** bytes (**LEN1**, **LEN2** i **DATA** block) located in 2 bytes. MSB,LSB

STATUS: For each command gets the **STATUS** byte transferred commands which can be:

ACK (0x06): command is correctly transferred to the device, and accepted by the device.
NACK(0x15): there is an error in the command ie. command has no valid format (CRC is not good) In this case, again sends the same command, up to three times in order to eliminate the possibility of interference in communication.

Answer by printer is command identical in shape to the command sent by the PC application, you also need to answer with a status byte ACK / NACK depending on the evaluated CRC.

Communication cycle:

This is short explanation of sending one command and reading answer from printer:
PC: Sends command
Printer: Responds with **ACK/NACK***
*If **ACK**:
Printer: Sends **RESPONSE** command which may be preceded with **WAIT** byte(s) (0x08). This byte is sent when operation in printed is taking time to be executed, so host should wait until **WAIT** byte(s) disappears. Printer may send one or more **WAITs** depending on requested operation.
PC: Responds with **ACK/NACK –** if PC detects wrong CRC, by sending NACK, printer may be requested to repeat block sending for maximum three times.
*If **NACK**:
PC: In this case, host may try sending same command for maximum three times, to avoid possible disturbance in communication on serial port.

RESPONSE: Data block with same form as command sent by PC application – short or long command depending on first sent byte **STX** or **SOH.** Command byte (byte which identifies requested command) is identical to the one sent by PC application or is set to 0x7F. This byte is followed by error type byte (error code table is described on last page of this document). If error byte is 0, it means that there is no error in command execution, i.e. command is executed correctly (this response is usualy sent for commands which perform printing on device, not commands used for reading data).

WAIT: After received **ACK** (0x06), PC waits for device status of executed command. If device is busy or there is some obstacle in current command execution (for example, printer head is risen), device will respond with **WAIT** byte (0x08) for each 300ms until command execution is finished and execution status is sent. If there is an error on device display, **WAIT** byte (0x09) is returned, and if there is error on printer, device returns **WAIT** byte (0x07) with byte which desribes error type.**

** Version 11.23 has a bug where printer sends byte with wrong error byte description after **WAIT** byte 0x07. It is recommended to process this error as universal printer error.

Commands for header and advertisement programming:

NOTE

Header and advertisement messages consist of 11 rows with 32 characters per row. Long command is use for programming. Changing header is allowed only after daily report is executed..

NOTE:

DATA block consists of 11 blocks where each has control byte and an 42 byte array which represent data (characters) to be printed in each row. For advertisement programming there is an additional byte which is used for advertisement number. In printer, this number is 0 (only one advertisement can be programmed). Control byte TYPE can have following values: 0xFF – empty row, 0x7F – row program defined, 0x3F – row program and bold text.

NOTE:

If command is sent without DATA block, printer returns current header – advertisement.

Command for header programming 20 (0x14).

[SOX<1>][LEN<2>][CMD<1>] (11)*{ [TIP <1>][TEXT <42>]} [CRC<2>]

SOX – start of communication
LEN – command length
CMD - command
TIP – control byte for row type definition
TEXT – text for current row. Charaters are coded using code table described at the end of this document.
CRC – check sum

Command for reading header 20 (0x14).

[SOX<1>][LEN<2>][CMD<1>][CRC<2>]

SOX - start of communication
LEN - command length
CMD - command
CRC – check sum

Printer response:

[SOX<1>][LEN<2>][CMD<1>] (11)*{ [TYPE <1>][TEXT <42>]} [CRC<2>]

SOX - start of communication
LEN – command length
CMD - command
TYPE - control byte for row type definition
TEXT - text for current row. Charaters are coded using code table described at the end of this document
CRC – check sum

Command for advertisement programming 22 (0x16).

[SOX<1>][LEN<2>][CMD<1>](11)*{[TYPE <1>][TEXT <42>]}][NO <1>] [CRC<2>]

SOX - start of communication
LEN - command length
CMD - command
TYPE - control byte for row type definition
TEXT - text for current row. Charaters are coded using code table described at the end of this document.
NO – advertisement ordinal number. In this case 0, printer has only 1 advertisement.
CRC – check sum

Example: programming advertisement message with following data:

Reklamna poruka
TESTIRANJE

PC:

```
03 DB 01 16 7F 52 65 6B 6C 61 6D 6E 61 20 70 6F 72 75 6B 61 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 7F 54 45 53 54 49 52 41 4E 4A 45 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 FF 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 FF 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
```

```
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 FF 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 FF 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 FF 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 FF 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 FF 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 FF 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 2020 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 FF 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 00 4A 5D
```

03 - SOH
DB 01 - LEN data length. In this case 475 (0x1DB)
16 - CMD
7F – control byte which defines which row is printed
52 65 6B 6C 61 6D 6E 61 20 70 6F 72 75 6B 61 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 – First row text. Block length is 42 although only 32 characters are printed.
7F - control byte which defines which row is printed
54 45 53 54 49 52 41 4E 4A 45 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20- Second row text. Block length is 42 although only 32 characters are printed.
FF - control byte which defines which row is printed
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20- Empty block. Block length is 42 although only 32 characters are printed.

........additional 8 blocks which are not printed, but have same structure as previous one.
4A 5D - CRC

Printer response: 06 02 02 7F 00 00 81

06 - ACK response that command is read correctly - CRC correct.
02 - STX start of communication.
02 - LEN (length of DATA)
7F - CMD command. Command which returns execution status message.
00 - STATUS error message byte (in this case there is no error in execution – 00; for error codes, check error code table at the end of this document).
00 81 - CRC (check sum - sum of all bytes, excluding STX i ACK)

PC: 06

06 - ACK response that command is read correctly - CRC correct.

Command for advertisement reading 22 (0x16).

[SOX<1>][LEN<2>][CMD<1>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
CRC – check sum

Printer response:

[SOX<1>][LEN<2>][CMD<1>] (11)*{ [TYPE <1>][TEXT <42>]} [CRC<2>]

SOX - start of communication
LEN – command length
CMD - command
TYPE - control byte for row type definition
TEXT - text for current row. Charaters are coded using code table described at the end of this document.
CRC – check sum

Command for header printing 24 (0x18).

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
CRC – check sum

Command for advertisement printing 25 (0x19).

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
CRC – check sum

GPRS commands:

NOTE:
For GPRS setting we use long command which can hold maximum 512 bytes. For GPRS parameters change JUMPER must be present.

Command for setting data for FTP server of tax office 103 (0x67). Jumper must be present on board.

[SOH<1>][LEN<2>][CMD<1>][TYPE<1>][IP<4>][USERNAME LENGTH<2>][USERNAME<22>][PASSWORD LENGTH<2>][PASSWORD<22>][UPLOAD ADDRESS LENGTH<2>][UPLOAD ADDRESS <46>][DOWNLOAD ADDRESS LENGTH<2>][DOWNLOAD ADDRESS <46>][CRC<2>]

SOH – communication start
LEN - command lenght
CMD - command
TYPE - FTP settings: TIP=1 for printer
IP – adress of FTP server
USERNAME LENGTH – lenght of user name
USERNAME – user name
PASSWORD LENGTH – password lenght
PASSWORD –password for server
UPLOAD ADDRESS LENGTH – command lenght
UPLOAD ADDRESS – adress on FTP server of tax office where data are sent
DOWNLOAD ADDRESS LENGTH - command lenght
DOWNLOAD ADDRESS – adress on FTP server of tax office from where commands are downloaded.
CRC - control check sum

Example for setting data for FTP server.

From PC to pritner: 03 96 00 67 01 a2 24 d8 59 05 00 67 75 65 73 74 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 05 00 67 75 65 73 74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 06 00 72 65 73 75 6c 74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 07 00 63 6f 6d 6d 61 6e 64 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 0c da

03- SOH
96 00 -LEN
67 - CMD , 67hex= 103dec
01 -TYPE
a2 24 d8 59 -IP , IP adress , in this example we set IP 89.216.36.162, i.e.
 59hex= 89 ;d8hex= 216; 24hex = 36; a2hex = 162
05 00 -USERNAME LENGTH
67 75 65 73 74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - USERNAME, user name in this example is "guest" tj 67hex = 103dec (g);75hex 117dec(u);65hex=101dec(e);73hex=115dec(s);74hex=116dec(t)
05 00 - PASSWORD LENGTH
67 75 65 73 74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 -PASSWORD ,in this example password is "guest" tj 67hex = 103dec (g);75hex 117dec(u);65hex=101dec(e);73hex=115dec(s);74hex=116dec(t)
06 00 - UPLOAD ADDRESS LENGTH
72 65 73 75 6c 74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - UPLOAD ADDRESS, adress on tax office server where we send data, in this example we set adress with "result"72hex =114dec(r)...
07 00 - DOWNLOAD ADDRESS LENGTH
63 6f 6d 6d 61 6e 64 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - DOWNLOAD ADDRESS, adress on tax office server from which we download commands, in this example we defined adress with
 "command" , 63hex=99dec(c)...
0c da -CRC,control check sum

From printer: 06 02 02 7f 00 00 81

06 - ACK
02 - STX
02 - LEN
7f 00 – successfully done command
00 81 - CRC

Example of setting data for FTP server.

Command for setting data for APN of GPRS provider 103 (0x67). Jumper must be present on board.

[SOH<1>][LEN<2>][CMD<1>][TYPE<1>][APN LENGTH<2>][APN <22>][USERNAME LENGTH<2>][USERNAME <22>][PASSWORD LENGTH<2>][PASSWORD <22>][CRC<2>]

SOH – communication start
LEN – command lenght
CMD - command

TYPE -APN settings: TIP=4 for printer;
APN LENGTH –command lenght
APN – adress of APN
USERNAME LENGTH **-** command lenght
USERNAME – user name
PASSWORD LENGTH –password lenght
PASSWORD -password
CRC – control check sum

From PC to printer : 06 03 4a 00 67 04 08 00 69 6e 74 65 72 6e 65 74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 07 00 74 65 6c 65 6e 6f 72 00 00 00 00 00 00 00 00 00 00 00 00 00 04 00 67 70 72 73 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 e6

06 - ACK
03- SOH
4a 00 -LEN
67 - CMD
04-TIP
08 00 - APN LENGTH
69 6e 74 65 72 6e 65 74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - APN , in this example APN - "internet"
07 00 - USERNAME LENGTH
74 65 6c 65 6e 6f 72 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - USERNAME , in this example is "telenor"
04 00 - PASSWORD LENGTH
67 70 72 73 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 - PASSWORD, in this example is"gprs"
08 e6 -CRC, control check sum

From printer: 06 02 02 7f 00 00 81

06 - ACK
02 - STX
02 - LEN
7f 00 - successfully done command
00 81 - CRC

Command for reading GPRS data from printer 104 (0x68).

PC sends data block :
[SOH<1>][LEN<2>][CMD<1>][TYPE<1>][CRC<2>]

SOH - communication start
LEN – command lenght
CMD – command
TYPE - FTP setting: TIP=1 for printer;
CRC – control check sum

Printer response:
[SOH<1>][LEN<2>][CMD<1>][TYPE<1>][IP<4>][USERNAME LENGTH<2>][USERNAME<22>][PASSWORD LENGTH<2>][PASSWORD<22>][UPLOAD ADDRESS LENGTH<2>][UPLOAD ADDRESS <46>][DOWNLOAD ADDRESS LENGTH<2>][DOWNLOAD ADDRESS <46>][CRC<2>]

SOH - communication start
LEN – command lenght
CMD – command
TYPE - FTP setting: TIP=1 for printer;
IP - Adress of FTP
USERNAME LENGTH – lenght of user name
USERNAME – user name for server
PASSWORD LENGTH – lenght of password
PASSWORD – server password
UPLOAD ADDRESS LENGTH – command lenght
UPLOAD ADDRESS – adress on FTP server of tax office where we send data from printer.
DOWNLOAD ADDRESS LENGTH – command lenght
DOWNLOAD ADDRESS – adress on FTP server of tax office from where we download commands.
CRC – control check sum

Example of GPRS data reading from printer.

From PC to printer: 03 02 00 68 01 00 6b
From printer: 06 03 96 00 68 01 a2 24 d8 59 05 00 67 75 65 73 74 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 05 00 67 75 65 73 74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 06 00 72 65 73 75 6c 74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 07 00 63 6f 6d 6d 61 6e 64 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 0c db

Command for reading

From PC to printer :
[SOH<1>][LEN<2>][CMD<1>][TYPE<1>][CRC<2>]

SOH – communication start
LEN – command lenght
CMD - command
TYPE - APN settings: TIP=4 for printer
CRC – control check sum

Printer response:
[SOH<1>][LEN<2>][CMD<1>][TYPE<1>][APN LENGTH<2>][APN <22>][USERNAME LENGTH<2>][USERNAME <22>][PASSWORD LENGTH<2>][PASSWORD <22>][CRC<2>]

SOH - communication start
LEN - command lenght
CMD - command
TYPE -APN settings: TIP=4 for printer;
APN LENGTH –command lenght
APN – adress of APN
USERNAME LENGTH –command lenght
USERNAME – user name
PASSWORD LENGTH – password lenght
PASSWORD -password
CRC – control check sum

Example of reading data for APN of GPRS provider from printer.

From PC to printer: 06 03 02 00 68 04 00 6e
From printer: 06 03 4a 00 68 04 08 00 69 6e 74 65 72 6e 65 74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 07 00 74 65 6c 65 6e 6f 72 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 00 67 70 72 73 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 e7

Command for resetting GPRS command 64 (0x40). Jumper must be on the board.

[SOH<1>][LEN<2>][CMD<1>][TYPE<1>][CRC<2>]

SOH – communication start
LEN – command lenght
CMD - command
TYPE - TIP=0 for printer
CRC – control check sum

Command for reading data from GPRS 96 (0x60).

PC sends dat block:

[STX<1>][LEN<1>][CMD<1>][CRC<2>]

STX – communication start
LEN – command lenght
CMD - command
CRC – control check sum

Printer response:

[STX<1>][LEN<1>][CMD<1>][SIGNAL LEVEL <1>][SEND TIME <4>][NEXT SEND <4>][SIM NUMBER <24>][CRC<2>]

STX - communication start
LEN - command lenght
CMD - command
SIGNAL LEVEL- GSM signal level
SEND TIME – time of sending data
NEXT SEND – scheduled time for next data sending
SIM NUMBER - ICCID number from SIM card
CRC – control check sum

From PC: 02 01 60 00 61

STX – communication start
LEN – command lenght
CMD -60hex=96dec
CRC – control check sum

From printer : 06 02 22 60 0f de 50 eb 16 5e a2 ec 16 06 38 39 33 38 31 30 31 31 31 30 34 30 38 35 32 35 34 34 38 30 06 06 06 06 08 b0

06 - ACK
02 - STX , communication start
22 - LEN, command lenght
60 - CMD , command 60hex=96dec
0f - SIGNAL LEVEL, GSM signal in moment of data reading, signal strenght should be in range 0-31, in this example is 0f hex=15dec
de 50 eb 16 - SEND TIME, time of data sending in seconds, this time represents number of seconds from 01.01.2000. (LSB-MSB)
5e a2 ec 16 - NEXT SEND,scheduled time when data sending will be performed, this time represents number of seconds from 01.01.2000. (LSB-MSB)

06 38 39 33 38 31 30 31 31 31 30 34 30 38 35 32 35 34 34 30 06 06 06 06 - SIM NUMBER, valid SIM number must start with 0x06 which represents control byte
08 b0 - CRC ,control check sum

Fast PLU programming:

NOTE: For PLU programming long command is used which can hold maximum 512 bytes.

Fast PLU programming command is 12 (0x0C).

[SOH <1>][LEN1<1>][LEN2<1>][CMD <1>] (N)*{[LEN <1>][CODE<4>][NAME<1-32>][M.UNIT+T.RATE<1>][PRICE<4>]}[CRC <2>]

(N)* number for next segment repeating. As many times as how many PLUs can contain so that command don't exceed maximum size of 512 bytes.

SOH – communication start

LEN1 –low byte with lenght of DATA block

LEN2 - high byte with lenght of DATA block

CMD - command

LEN – DATA block lenght

CODE: PLU code from 1 to 75000.

NAME: textual description on PLU which represents byte string from 1 to 32. Charachters from PLU description are coded by code table at the end of the document.

M.UNIT+T.RATE: Measure unit is number written in high four bits, tax rate is written in low four bits. Measure unit has value from 0 to 14 and tax rate from 0 to 8.

PRICE: Price which is written in 4 bytes is multiply by 100 (price is written with two decimal places).

CRC – control check sum

Example: Three articles programming

NAME: Article 1; CODE: 1; PRICE: 1000,00; T.RATE: A; M.UNIT: kom

NAME: Article 2; CODE:2; PRICE:2000,00; T.RATE: Г; M.UNIT: kg

NAME Article 3; CODE: 3; PRICE: 3000,00; T.RATE: Д; M.UNIT: g

From PC: 03 3A 00 0C 12 01 00 00 00 41 72 74 69 63 6C 65 20 31 00 A0 86 01 00 12 02 00 00 00 41 72 74 69 63 6C 65 20 32 11 40 0D 03 00 12 03 00 00 00 41 72 74 69 63 6C 65 20 33 22 E0 93 04 00 0C E5

03 - SOH
3A - LEN1
00 - LEN2
0C - CMD

12 - LEN (block lenght – for first article)
01 00 00 00 – Article 1 code number (LSB, MSB)
41 72 74 69 63 6C 65 20 31 - NAME (characters are written in order with values from code table at the end of this document) ARTICLE 1
00 – M.UNIT+T.RATE (tax rate **A** and **kom** in this example)
A0 86 01 00 - PRICE (LSB, MSB)

12 - LEN (block lenght – for second article)
02 00 00 00 - CODE drugog artikla
41 72 74 69 63 6C 65 20 32 - NAME (characters are written in order with values from code table at the end of this document) ARTICLE 2

11 – M.UNIT+T.RATE (Г and kg)
40 0D 03 00- PRICE (LSB, MSB)

12 - LEN (block lenght – for third article)
03 00 00 00 – third article code (LSB, MSB)
41 72 74 69 63 6C 65 20 33 - NAME(characters are written in order with values from code table at the end of this document) ARTICLE 3
22 - M.UNIT+T.RATE (Д and g)
E0 93 04 00 - PRICE (LSB, MSB)

0C E5 - CRC

From Printer: 06 02 02 7F 00 00 81

06 - ACK
02 - STX
02 - LEN
7F 00 – confirmation about successfully executed comand
00 81 - CRC

Fast PLU reading:

NOTE:
For PLU reading long command is used which can hold maximum 512 bytes. Request is sent for next article (PLU). Printer gives response with articles,one after the other by requested code in requested range.

Command for fast PLU reading is 19 (0x13).

PC sends data block:

[SOH<1>][LEN<2>][CMD<1>][CODE<4>][CRC<2>]

SOH – communication start
LEN – command lenght
CMD - command
CODE – article code which is number from 1 to 75000.
CRC – control check sum

Printer response:

[SOH<1>][LEN<2>][CMD<1>][N*{[LEN<1>][CODE<4>][NAME<32>][M.U+T.RATE <1>][PRICE <4>]}[CRC<2>]

(N)* number for next segment repeating. As many times as how many PLUs can contain so that command don't exceed maximum size of 512 bytes.

SOH - communication start
LEN - command lenght
CMD - command
LEN – command lenght
CODE- code
NAME- article name
M.U+T.RATE - Measure unit is number written in low four bites, tax rate is written in high four bits. Measure unit has value from 0 to 15 and tax rate from 0 to 8.
PRICE - Price which is written in 4 bytes is multiply by 100 (price is written with two decimal places).
CRC – control check sum

Example 1: Article reading:

From PC:: 06 03 05 00 13 01 00 00 00 00 19

06-ACK
03 - SOH
05 00 - LEN
13 - CMD
01 00 00 00 - CODE , CODE from which reading starts
00 19 - CRC, control check sum (05 + 13+ 01)hex

Printer response: 06 08 08 08 08 08 08 03 60 00 13 12 01 00 00 00 41 72 74 69 63 6c 65 20 31 b0 54 2c 00 00 12 02 00 00 00 41 72 74 69 63 6c 65 20 32 03 04 ae 00 00 12 03 00 00 00 41 72 74 69 63 6c 65 20 33 63 f2 bd 0000 12 04 00 00 00 41 72 74 69 63 6c 65 20 34 85 a5 fe 00 00 12 05 00 00 00 41 72 74 69 63 6c 65 20 35 07 e9 65 01 00 17 c4

06 - ACK
08 - WAIT
03 -SOH
60 00 - LEN
13 - CMD
12 - LEN, lenght of next code 12hex=18dec
01 00 00 00 - CODE
41 72 74 69 63 6c 65 20 31- NAME, 41hex = 65dec(A); 72hex = 114dec(r), 74hex = 116dec(t); 69hex = 105dec(i); 63hex = 99dec(c); 6chex = 108dec(l); 65hex = 101 dec(e); 20hex = 32dec(); 31hex = 49dec(1) – look at the code table at the end of this document.
b0 – M.U+T.RATE, bhex =11dec this is measure unit, 0 is tax rate (A)
54 2C 00 00 -PRICE, 2C54hex = 11348/100 is 113.48 which is article price
12 - LEN, lenght of next code 12hex=18dec
02 00 00 00 - CODE
41 72 74 69 63 6c 65 20 32- NAME
03 – M.U+T.RATE, measure unit is 0, and 3 is tax rate (Ђ)
04 ae 00 00 -PRICE, ae04hex = 44548/100 is 445.48 which is article price
12 - LEN, lenght of next code 12hex=18dec
03 00 00 00 – CODE
41 72 74 69 63 6c 65 20 33- NAME
63 - JM+PORESKA STOPA, 6 jedinica mere definisana JM6, 3 je poreska stopa PS4 (Ђ)
f2 bd 00 00 -CENA, (bd f2)hex = 48626 /100=486,26
12 - LEN, lenght of next code 12hex=18dec
04 00 00 00 - KOD
41 72 74 69 63 6c 65 20 34- IME
85 - M.U+T.RATE, 8 is measure unit, 5 is tax rate (Ж)
a5 fe 00 00 -PRICE,(fea5)hex = 65189/100=651,89
12 - LEN, lenght of next code 12hex=18dec
05 00 00 00 - CODE

41 72 74 69 63 6c 65 20 35- NAME
07 – M.U+T.RATE, 0 is measure unit, 7 is tax ratet (J)
e9 65 01 00 -PRICE,(0165e9)hex = 91625/100=916,25
17 c4 -CRC

After this, PC sends command

From PC: 06 03 05 00 13 06 00 00 00 00 1e

06-ACK
03-SOH
05 00 -LEN
13-CMD
06 00 00 00- CODE , sends code of next code whose place is free for defining (this code is not programmed in printer and because of that printer sends error 12, which means given price is not valid because this article does not exists in printer).

Printer: 06 08 08 08 08 08 02 02 7f 12 00 93

06-ACK
08-WAIT
02 -STX
02 -LEN
7f 12 - 7f - FLAG which points to error which is written in next byte ; 12 - given price is not valid

00 93 -CRC

Fast PLU deleting:

NOTE:
For PLU delete long command is used which can hold maximum 512 bytes. In printer there must be programmed at least one article. Command for fast article delete is 13 (0x0D).

PC send command block:

[SOX<1>][LEN<2>][CMD<1>][(N)*[CODE <4>]][CRC<2>]

(N)* number for next segment repeating. As many times as how many PLUs can contain so that command don't exceed maximum size of 512 bytes.

SOH – communication start
LEN – command lenght
CMD – command
CODE - article code which is number from 1 to 75000.
CRC - control check sum

Example 1: command for fast article delete.

From PC: 03 0d 00 0d 01 00 00 00 02 00 00 00 03 00 00 00 00 20

03 - SOH, communication start
0d 00- LEN
0d - CMD, command 0dhex =13dec
01 00 00 00 - CODE, first article code
02 00 00 00 – CODE, second article code
03 00 00 00 - CODE, third article code
00 20 - CRC, control check sum (0d+0d+01+02+03)hex

Printer : 06 02 02 7f 00 00 81

06 - ACK
02 - STX
02 - LEN
7f 00 - successfully executed comand
00 81 - CRC

PLU price change:

NOTE:
For PLU price change long command is used which can hold maximum 512 bytes.
Price can not be 0 (zero). It's possible to change price for more than one article at same time.

Command for price change is 11 (0x0B).

[SOH<1>][LEN<2>][CMD<1>][(N)*{[CODE <4>][PRICE<4>]}][CRC<2>]

(N)* number for next segment repeating. As many times as how many PLUs can contain so that command don't exceed maximum size of 512 bytes.

SOH – communication start
LEN - command lenght
CMD - command
KOD - article code which is number from 1 to 75000
PRICE Price which is written in 4 bytes is multiply by 100 (price is written with two decimal places).
CRC – control check sum

Example of command for article price change.

From PC: 03 09 00 0b 04 00 00 00 88 13 00 00 00 b3

03 - SOH
09 00 - LEN
0b - CMD, 0bhex = 11dec
04 00 00 00 - CODE , article code for price change
88 13 00 00 - PRICE, new value for for this article , 1388hex = 5000/100 =50,00
00 b3 - CRC, control check sum

Printer: 06 02 02 7f 00 00 81

06 - ACK
02 - STX
02 - LEN
7f 00- successfully executed comand
00 81 - CRC

Non fiscal text printing:

NOTE:
For printing non fiscal text long command is used which can hold maximum 512 bytes. Text is printed with latinic letters and can not containt reserved words for fiscal documents (dnevni izveštaj...)

Command for non fiscal text printing is 26 (0x1A).

[SOH<1>][LEN<2>][CMD<1>][FOOTER <1>][(N)*{ [TYPE <1>][TEXT <32>]}][CRC<2>]

SOH – communication start
LEN – command lenght
CMD - command
FOOTER - means does after received block footer will be printed - cuts the paper (0-ne, 1 -da)
TYPE - font type, can be

- regular letters – font 0xFF
- bold letters 0xFF&~0x10 ie 0xEF

TEXT – regards to string of characters in one line
CRC – control check sum

Example 1: non fiscal text printing.

From PC: 03 44 00 1a 01 ff 4f 76 6f 20 6a 65 20 6e 65 66 69 73 6b 61 6c 6e 69 20 74 65 6b 73 74 20 6b 6f 6a 69 20 73 6c 75 ff 7b 69 20 7a 61 20 74 65 73 74 69 72 61 6e 6a 65 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 16 88

03- SOH
44 00 -LEN
1a - CMD, 1ahex = 26dec
01 -FOOTER, in this example footer value is 01 ie. After printing of non fiscal text paper will be cut
ff - TYPE , type of printing is ff ie small letters are in non fiscal text
4f 76 6f 20 6a 65 20 6e 65 66 69 73 6b 61 6c 6e 69 20 74 65 6b 73 74 20 6b 6f 6a 69 20 73 6c 75 - TEXT, look at the code table at the end of document 4fhex=79dec(O);76hex=118dec(v);6fhex =111 dec(o)....
ff -TYPE

7b 69 20 7a 61 20 74 65 73 74 69 72 61 6e 6a 65 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 - TEXT
16 88 - CRC , control check sum

Printer: 06 02 02 7f 00 00 81

06 - ACK
02 - STX
02 - LEN
7f 00 - successfully executed comand
00 81 - CRC

Commands for direct memory reading:

NOTE:

For reading memory, long command is used with maximum 512 bytes. Request is sent for start memory address. Printer responds with data from given address to the end of command.

Command for reading RAM memory 9 (0x09).

[SOH<1>][LEN<2>][CMD<1>][MEMORY_ADDRESS <4>][CRC<2>]

SOH – start of communication
LEN – command length
CMD - command
MEMORY_ADDRESS, address from which RAM memory is read - start address
CRC – check sum

Command for reading fiscal memory 8 (0x08).

SOH – start of communication
LEN – command length
CMD - command
MEMORY_ADDRESS, address from which fiscal memory is read - start address
CRC – check sum

Command for reading FLASH memory 76 (0x4C).

SOH – start of communication
LEN – command length
CMD - command
MEMORY_ADDRESS, address from which FLASH memory is read - start address
CRC – check sum

ASCII commands:

NOTE:

Commands are used to read data from fiscal memory in ASCII format, which is provided for sending data to tax office server.

Command for reading daily report data in ASCII format 91 (0x05b).

PC:

[STX<1>][LEN1<1>][CMD<1>][DAILY NUM <2>][CRC<2>]

STX – start of communication
LEN – command length
CMD - command
DAILY NUM – number of daily report for which data is read
CRC – check sum

Printer response:

[STX<1>][LEN1<1>][CMD<1>][ASCII STRING DAILY <35>][CRC<2>]

STX –start of communication
LEN – command length
CMD - command
ASCII STRING DAILY – daily report data in format as it would be sent to tax office server
CRC – check sum

Command for reading reset data in ASCII format 92 (0x05c).

PC:

[STX<1>][LEN1<1>][CMD<1>][RESET NUM <2>][CRC<2>]

STX –start of communication
LEN – command length
CMD - command

RESET NUM – reset number for which data is read
CRC – check sum

Printer response:

[STX<1>][LEN1<1>][CMD<1>][ASCII STRING RESET <15>][CRC<2>]

STX –start of communication
LEN – command length
CMD - command
ASCII STRING RESET- reset data in format as it would be sent to tax office server
CRC – check sum

Command for reading VAT values in ASCII format 93 (0x05d).

PC:

[STX<1>][LEN1<1>][CMD<1>][VAT NUM <2>][CRC<2>]

STX - start of communication
LEN - command length
CMD - command
VAT NUM- VAT number for which data is read (indexing 0-8)
CRC – check sum

Printer response:

[STX<1>][LEN1<1>][CMD<1>][ASCII STRING VATS <33>][CRC<2>]

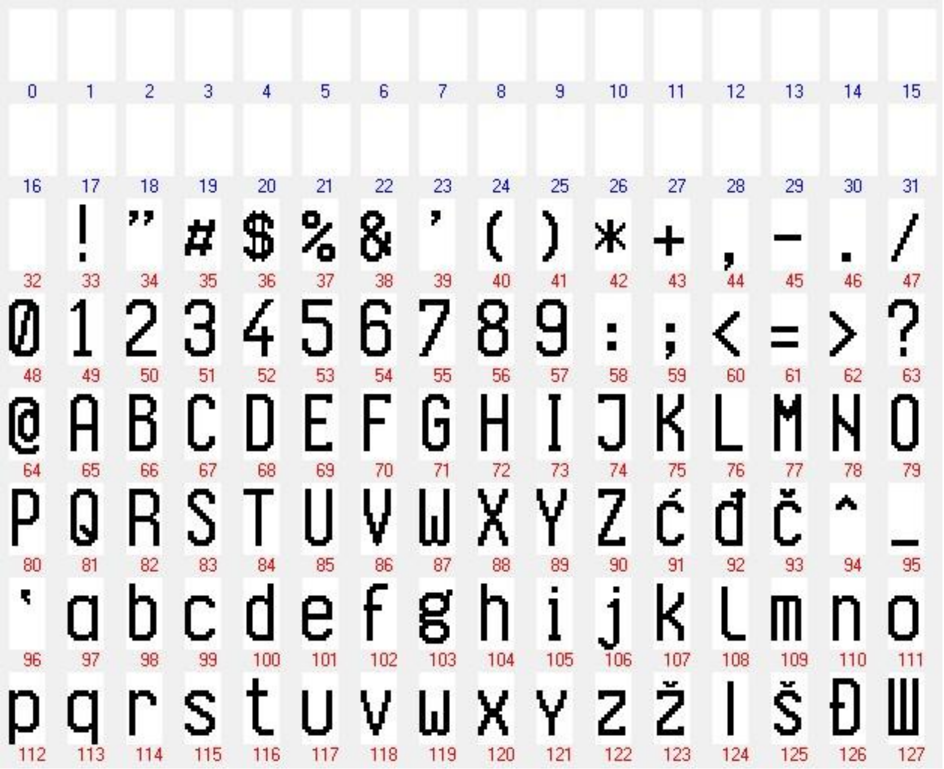
STX –start of communication
LEN – command length
CMD - command
ASCII STRING VATS - VAT data in format as it would be sent to tax office server

CRC – check sum

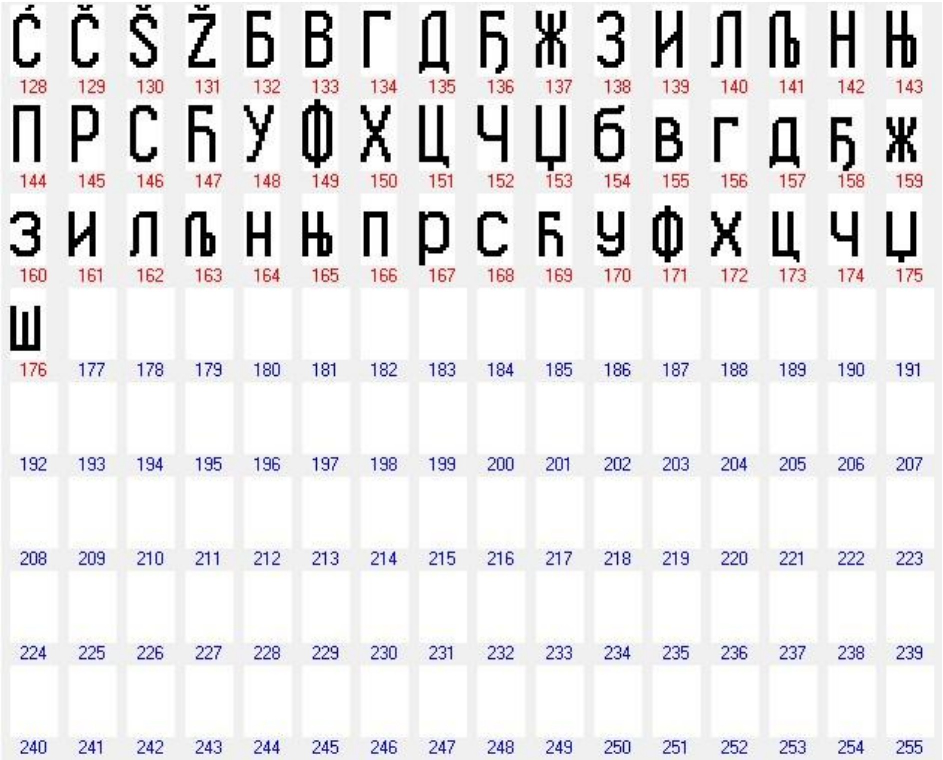
Error and character table:

ERROR TYPE	CODE
Article with given code and barcode already exist	10
Given article code is not valid	11
Given article price is not valid	12
Given article department is not valid (no departments in printer P2DS)	13
Given VAT is not valid	14
Given measure unit is not valid	15
Given code or barcode is not valid (no barcode in printer P2DS)	16
Given barcode is not valid (no barcode in printer P2DS)	17
Article does not exist	18
Article base is full	19
Article description is empty	20
Inadequate value (unique error when sending data to printer)	21
Inadequate value (unique error when reading data from printer)	22
Value is the same. (unique error when programming data to printer)	23
Value is deleted. (unique error when reading data from printer)	24
Test is successful.	25
Value is already defined. (unique error when sending data to printer)	26
Code exists	27
Value cannot be changed. (unique error when sending data to printer)	28
Fiscal bill started	31
Fiscal day started	32
Inadequate value for article sale	33
Fiscal bill must be closed	34
VAT is not defined. Article has undefined VAT value.	35
Fiscal value too small. Calculated tax for given price and quantity is zero.	36
Fiscal value too big. Calculated tax for given price and quantity is out of range.	37
Fiscal bill not started.	38
Daily report must be executed	39
Picture already defined.(no picture in printer P2DS)	42
Base is empty	43
Device is busy. Please try later	44
Journal printer head is risen.	65
Number of remaining resets critical	66
Number of remaining VAT changes critical	67
Number of remaining daily reports critical	68
Execute technical intervention	69
Jumper is not present	75

Time cannot be adjusted	76
Wrong time	77
Jumper is present	78
Code already exists	79
You must be logged in	80
You don't have permissions	81
Command does not exist	82
Option is not supported	83
Error in cash in/out flow	97
Voiding not finished	99
Device is busy	100
Command length error	101
Command does not exist	102
Command cannot be executed	103
Last artical in base	104
Thermal printer overheated	217
Error, no paper	218
Fiscal receipt printer head is risen	219
Memory error	220
Fisal memory full	221
Fiscal memory error	222
Fiscal memory is fiscalized	223
Fiscal memory PIB error	224
Display error	225
Keyboard error	226
GPRS modem error	227
GPRS modem not detected	228
GPRS modem is busy	229
GPRS modem is working	230
Reset must be executed	235



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
P	Q	R	S	T	U	V	W	X	Y	Z	ć	đ	č	^	_
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
p	q	r	s	t	u	v	w	x	y	z	ž	|	š	Đ	Ш
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127



Ć	Č	Š	Ž	Б	В	Г	Д	Ђ	Ж	З	И	Л	Љ	Н	Њ
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
П	Р	С	Ћ	У	Ф	Х	Ц	Ч	Џ	б	в	г	д	ђ	ж
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
з	и	л	љ	н	њ	п	р	с	ћ	у	ф	х	ц	ч	џ
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
ш															
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Table of content: