MCP

Author: Vgk
Status: Preliminary
Release: 1.0.7

FHMAS/ADHME

MAT SA

TABLE OF CONTENTS

1. Purpose of this document

The purpose of this document is to provide the necessary specification to software designers interested in communicating directly with all MAT fiscal signature devices with a network (IEE802.11) interface.

This document assumes that the reader is familiar with basic UDP and TCP/IP communication concepts, such as sockets, timeouts, etc. Also assumes that the reader is familiar with fiscal POS/ECR functioning and procedures.

2. Goals

The developer will have all necessary information for implementing all protocol layers, thus be able to:

- -Keep track of all transaction operations (sales, voids, refunds etc.)
- -Produce a SHA-1 digital signature for any number of invoices.
- -Update the device's list of daily signatures in case of fatal failure.
- -Expand the available local database of items to arbitrary numbers
- -Perform the ECR/POS configuration (setup) remotely
- -Issue receipts and all reports via protocol commands

3. Design approach and compatibility issues

Developers should take into consideration future additions or expansions to this specification. The goal is that an application designed using an older revision specs will function correctly in newer revision protocol.
In order to do so, the developers *must* check responses only for the presence of the known information and 'quietly' discard the information that is unknown. The designers of this protocol guarantee that the extensions of this protocol will not alter the position or the type of the information (unless absolutely unavoidable). Extra fields will always be added to the right of the reply strings. Specifically, these are the rules that deliver the highest compatibility:

a) Check the protocol version number. This information guarantees safety towards new commands. For example (hypothetically):
   In protocol revision '01.02' and higher the command '#' is supported, so reading a revision '01.00' indicates that the command '#' will fail.
b) Always assume correct a reply that has more fields than expected.

For example:
Reply expected: "/1/AAAAA/BBBB/CCCC/"
Reply received: "/1/AAAAA/BBBB/CCCC/DDDDD"
(Field 'DDDDD' is unexpected, but should not generate an Error because all the expected fields are present. So this field *should* be silently discarded.)

c) Always assume correct a 'FLAGS' field that is longer than expected.
For example:
Reply expected: "/1001001001/"
Reply received: "/1001001001001/"
(Three extra bits in the 'FLAGS' field are unexpected. The application must discard them without generating Errors).

d) It is an excellent design approach not to be very strict with numerical ranges or string lengths expected. This guarantees that the application will be compatible with other ECR/POS devices that use this protocol but having different resources to operate with. For example, an ECR/POS having more memory is probable to support a wider local item base, reporting higher index numbers. Or, a different printer mechanism may limit, for example, a header line length. Having a flexible design promises maximum compatibility with different hardware requiring very little (or no) changes to application source code.

3.1. Further information

The implementers are encouraged to study and/or use parts of code examples which are part of this document. Also, they must keep informed of any changes in this specification due to the status of this document. Suggestions from developers may or may not influence details of the document until it reaches 'final' status.

4. Communications interface

The device communicates with host computer via an ethernet (LAN) connection, using the Transmission Control Protocol (TCP) as part of the TCP/IP protocol suite.
Device 'listens' to a fixed TCP port, in which the host should send all its TCP packet frames. This port must be set (by default) to value 9101.

5. Protocol layers discussion

There are two different needs which the ECR/POS satisfies with two separate protocol layers. The first is the need of keeping track of the POS activity and the extension of the local database of items. The

second is the need to use the ECR/POS as a terminal device which we can call 'fiscal printer'.
The protocol layers for these needs respectively are:

- The 'online' protocol layer (It will be referred as 'online protocol')
- The 'command' protocol layer (It will be referred as 'command protocol')

Note that there is no such case where both layers are active at the same time due to the nature of the needs each layer deals with. To be clearer, the online protocol is required when it is desired to observe the POS device's activity when the operator of the ECR/POS issues receipts or any other document with it. The command protocol is required when is desired to use the device with a host computer application that issues the receipts and reports to the ECR as a fiscal printer.

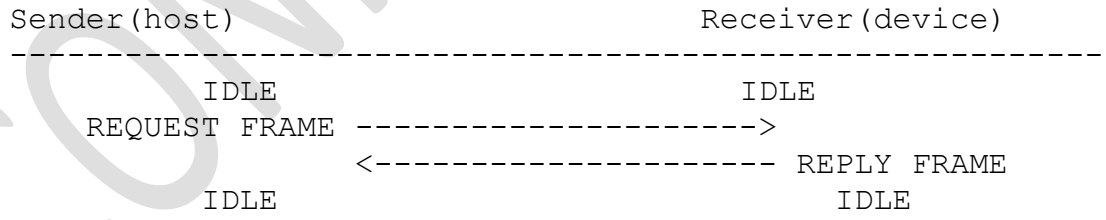
Although these two layers cannot coexist at the same time of POS operation, switching between them is allowed anytime. As expected, communication rules and procedures that layers use are the same.

A major difference between the online and command protocol is the origin of the communication. In the online protocol, the communication starts from the POS/ECR in contrast with the command protocol where the communication starts by the host computer.

6. Common rules

6.1. Model of data interchange

Both protocol layers share a common model of interchanging data with the host. The next scheme describes this model:



This scheme although describes the typical flow of data between the two communicating devices (Fiscal Device and host computer) does not include any other situation such as Errors in transmittance, re-transmittance etc. Note also that the 'sender' will be the ECR/POS and the 'receiver' will be the host in online protocol. In the command protocol, the 'sender' will be the host and the 'receiver' will be the ECR/POS.

Observe that this model includes two different packet transmittances, one from sender to receiver and one from receiver to sender. In the paragraphs to follow we will call the first packet 'request packet' and the second one 'reply packet' for simplicity. Reply packets are always sent by the ECR/POS when receiving command

protocol requests. Also reply packets may be sent in special cases by the host computer at online protocol.

6.2. States of protocol

For a better understanding of the previous paragraph and the communication flow, we can define states which communication 'sides' will enter.

- **Idle state**
  This is the state before any communication attempt takes place.
- **Packet transmittance state**
  The sender will enter this state to transmit a request packet and the receiver to transmit a reply packet.
- **Packet reception state**
  The receiver enters this state after acknowledging the senders enquire to get the request packet. The sender will enter this state right after verifying a positive acknowledge from the receiver, and only if the specific protocol case requires a reply packet.

Considering the above, the state flow for the sender and the receiver in a typical communication attempt will be:

Sender	Receiver
Idle	Idle
Packet transmittance state	Packet reception state
Packet reception state	Packet transmittance state
Idle	Idle

6.2.1. States definition -> Packet transmittance state

This state is the transmittance of either a request or a reply packet by the sender and the receiver respectively. Packets in both cases follow the rules described in a later paragraph. On completion of the packet transmittance, the sender or receiver advances to the next state, if any. During the packet transmittance state, the sender or receiver may also transmit control codes which will be transparent for the packet data, i.e. they will not be included in the data section of the packet.

6.2.2. States definition -> Packet reception state

The packet reception state is the process of receiving a request or reply packet. The sender will enter this state when receiving a reply packet and the receiver when receiving a request packet.

6.3. Packet purpose and structure

The actual communication data in both protocol layers are encapsulated in a 'packet'. As described above, there are request packets and reply packets. In simple words, request packets contain instructions that the sender wishes the receiver to follow or plain information. Reply packets are information which describe how receiver followed the instructions and/or plain information.
Request packets are always sent by the sender. Reply packets are always sent by the receiver. Request and reply packets have the same basic structure in both online and command protocol layers but differ in their contents.
Any valid packet is considered 'data' octet. Valid data octets must be between values '32' and '255' (decimal). Octets lower than '32' are considered 'control' codes [1] and MUST be interpreted specially. Valid data octets are forming the complete data section. Control codes are NOT part of the data.
The length of the data section is variable, due to its multifunctioning purpose. ECR/POS can accept data up to 250 octets of data in a single packet. Hosts MUST be able to accept at least the same amount of data in a single packet. ECR/POS will discard any further data if this limit is reached producing a negative acknowledge to the host.

Inside the data section of a packet, request or reply, are 'data fields':

Data				
Field 1	Field 2	Field 3	……..	Field N

Data fields form the total of the data section of a packet. Each field's size may vary. For this reason, a 'special' data character is defined to function as 'field separator'. In both protocol layers, the field separator character is the slash '/' (ASCII character 47 decimal, 057 octal, 2F hexadecimal). ECR/POS interprets this character as 'start of next field'. Host application must do the same. As a result of this character's special meaning, hosts MUST NOT include this character as part of field data but only as field separator. The reason for this is that the ECR/POS will incorrectly treat it as field separator and count one extra field in the packet, probably also shifting all other fields by one position to the right.
Fields vary in size and content. Various types of fields are described in a later paragraph in detail.

6.3.1. Fields - discussion

As already mentioned, fields are the building blocks of a data packet. In this paragraph we will examine all available types of fields and their basic restrictions and requirements.
In both layers, there are only two classes of fields: the string class and the numeric class. Further 'type' labelling was necessary to be

defined to document each type's ranges and restrictions. Understanding those is essential because when out of 'type' range fields are sent will be rejected by the ECR/POS on further packet processing.
Although fields of certain class and type have a range, the specific packet may REQUIRE a lower range for successful. Keeping this in mind, applying fields to a packet should be done following this scheme:
- Apply class restrictions checks
- Apply type restrictions and range checks
- Apply packet's specification for field's restrictions and range

6.3.2. Fields - classes

As mentioned, field classes are either string or numeric. These are the attributes of each class.

String class:
- Can contain any character of value 32 to 255 (decimal) except slash ('/')
- Can be of zero to any length that does not exceed the maximum packet size

Numeric class:
- Can contain any numeric character, a decimal point
- Can contain any 'A' to 'F' digit if hexadecimal (*)
- Can contain a minus as a first character
- Can have a total length of zero to 12 characters

(*) Hexadecimal values are only sent at command protocol reply to packets for device status map and fiscal status map fields.

6.3.3. Fields - types in detail

Field types are used as a method of generating or recognizing specific or generic fields for a use in a packet. The list that follows defines the ranges and restrictions of the specific types.

INTEGER type	
Class:	Numeric
Value range:	'-999999' to '999999'
Digit range:	1 to 6 digits
Notes:	Fields of this type must not contain any decimal part or decimal point. This type is usually used as a counter field or an index.

DATE6 type	
Class:	Numeric
Value range:	'010199' to '311240'
Digit range:	When required, must be 6 digits.

	When optional, may not be sent at all.
Notes:	Specifies a date. Date format is DDMMYY.
DATE8 type	
Class:	Numeric
Value range:	'01011999' to '31122040'
Digit range:	When required, must be 8 digits. When optional, may not be sent at all.
Notes:	Specifies a date. Date format is DDMMYYYY.

TIME type	
Class:	Numeric
Value range:	'000000' to '235959'
Digit range:	When required, must be 6 digits. When optional, may not be sent at all.
Notes:	Specifies a time. Time format is HHMMSS.

FLAGS type	
Class:	Numeric
Value range:	'0' to '1' or '2' for each flag in field
Digit range:	When required, must be as long as the packet requires. When optional, may not be sent at all.
Notes:	Flags type is used to minimize packet fields where a single "true"/"false" or "yes"/"no" type of information must be passed for various attributes. In case of '2' the specific digit should be ignored.

AMOUNT type	
Class:	Numeric
Value range:	'-99999999.99' to '99999999.99'
Digit range:	1 to 12 total 0 to 8 integer part 0 to 2 decimal part
Notes:	AMOUNT is usually used to specify prices, discounts, payment values, totals, etc. When used to specify payments, this type will always be expressed in the active note (i.e.: drachmas or euro)

QTY type	
Class:	Numeric
Value range:	'-99999.999' to '99999.999'
Digit range:	1 to 10 total 0 to 5 integer part 0 to 3 decimal part

Notes:	QTY is used to specify quantities of any kind.

RATE type	
Class:	Numeric
Value range:	'0.000000' to '9999.999999'
Digit range:	1 to 11 total 0 to 4 integer part 0 to 6 decimal part
Notes:	RATE is used to specify currencies of foreign notes or euro to drachmas rate and vice versa

PERCENT type	
Class:	Numeric
Value range:	"0.00" to "100.00"
Digit range:	1 to 6 total 0 to 3 integer part 0 to 2 decimal part
Notes:	PERCENTAGE is used to specify a discount percentage, a markup percentage etc.

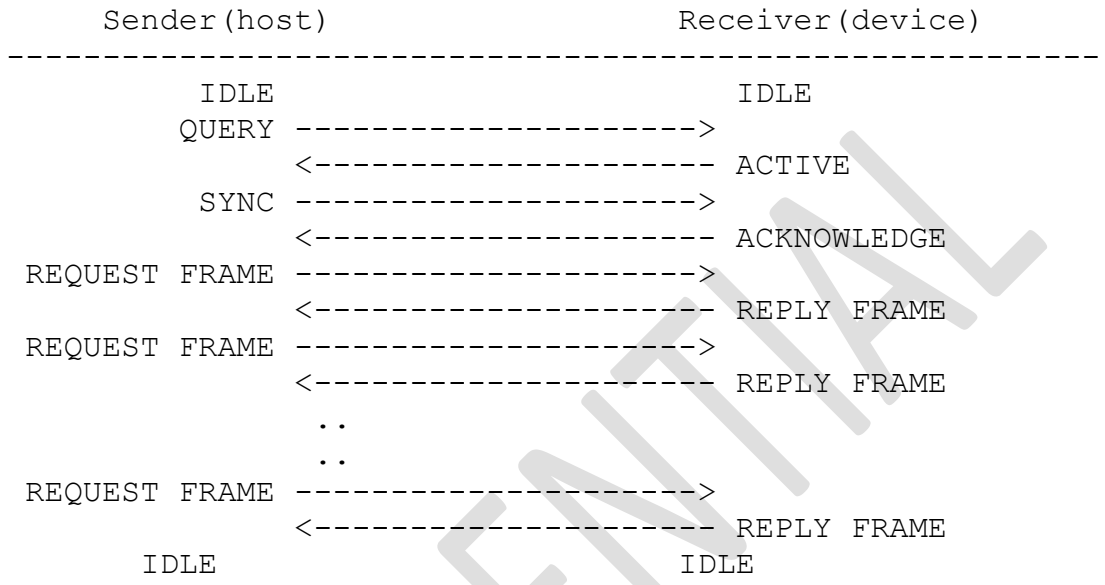
STRING type	
Class:	String
Value range:	-
Character range:	1 to 240 (if not exceeding max packet size)
Notes:	A normal string

7. SDNP protocol

Protocol discussion

7.1. Model of data interchange

The device uses the following model for communicating with the host:



This scheme although describes the typical flow of data between the two communicating sides (device and host computer) does not include any other situation such as errors in transmition, retransmition etc. Observe that this model includes two different packet transmitions, one from sender to receiver and one from receiver to sender. In the paragraphs to follow we will call the first 'request frame' and the second one 'reply frame' for simplicity. Reply frames are always sent by the device when receiving request frame.

7.2. States of protocol

For a better understanding of the previous paragraph and the communication flow, we can define states which communication 'sides' will enter.

- **Unsynchronized state**

This state is the device's initial state of this protocol. At this state, he device accepts only synchronization requests (ie, SYNC frames) or query requests (ie, QUERY frames).

- **Synchronized state**

This state is established right after the device has received and acknowledged a SYNC frame. This means that for a given time window the device considers itself 'synchronized', ie available for receiving data frames from host. After the expiration of this time window, provided that in the meantime no other data or synchronization frames have been received, the device enters the 'unsynchronized' state. The

SYNC frame provides the synchronization mechanism and is described in a later paragraph.

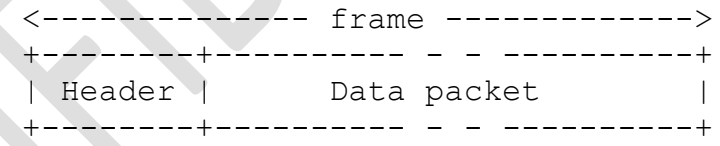
7.3. Frame types and organization

The actual communication data are exchaged by 'frames'. Specificaly, there are 3 types of frames:

* **Control frames**, that are responsible for correct delivery of request/reply frames. Control frames can be exchanged regardless of the state the device or the host is.
* **Request frames**, that contain instructions that sender (host) wishes the receiver (device) to follow. Request frames are successfully exchanged only in synchronized state.
* **Reply frames** contain information which describe how the device followed the instructions that specified by a request frame and other optional returned information that are directly relative to the action requested. So, a reply frame is only sent as a consequence of a request frame. This indicates that replies are sent only in synchronized state.

Request frames are always sent by the host. Reply frames are always sent by the device. Control frames are sent by both communication sides. All frames have the same basic structure but differ in their contents.

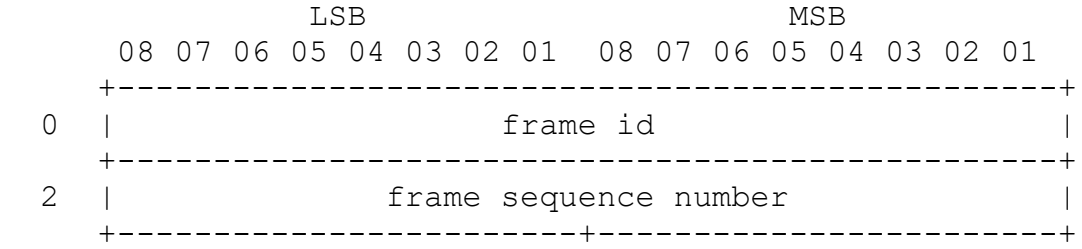
The frame generic structure is the following:

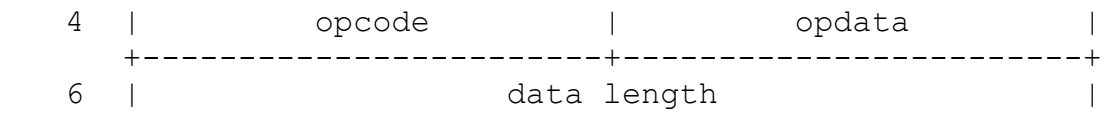


Header and data packets are explained in the following paragraphs.

7.3.1. Frame header structure

The SDNP header consists of 12 octets, that some of them are organized in 16-bit (WORD) values and some in 8-bit (BYTE) values. All WORDS in the header are in INTEL order (ie, least significant byte first).





Members of the header are as follows:

FRAME ID
========
A 16-bit (WORD) constant value, that specifies the SDNP protocol and is fixed to value 0x7A2D (decimal 31277) in frames sent by the devices, and to value 0xE18F (decimal 57743) at frames sent by hosts. Frames received by a host that do not have the expected value must be silently discarded in all cases. Frames received by the device not having the expected value are silently discarded.

FRAME SEQUENCE NUMBER (FSN)
===========================
A 16-bit (WORD) value that numbers the request/reply pairs when in synchronized state. After each successful request/reply transmission, this counter must be advanced by one in each communicating side. The FSN member of a SYNC frame is considered to be IFSN (Initial FSN). In control frames sent by the device, the FSN indicates the sequence number of the frame that caused the transmission of the control frame. For example, in reception of a SYNC frame with FSN=100, the device will respond with an ACK frame, with the FSN also set to 100. In data frames sent by the device the device REPLY frame's FSN will match the host's REQUEST frame's FSN.
After each synchronization, the expected FSN of the next REQUEST frame is IFSN+1.

OPCODE
======
An 8-bit (BYTE) value that specifies the frame type. We can define the following frame types:

QUERY

Name	Opcode	Opdata	Direction
QUERY	0x00	0x00	Host to device

A frame sent by the host in order to determine all devices connected in a network. When received by a device, it responds with an ACTIVE frame, indicating that this device is active and ready. Usually, this

frame is sent by the host to the 'broadcast' IP address, so all available devices will receive the QUERY and each produce a response. The FSN member of the QUERY frame is copied by the device to the ACTIVE response.

ACTIVE

Name	Opcode	Opdata	Direction
ACTIVE	0x01	0x00	Device to Host

This frame is sent by the device as a response to a QUERY frame, to indicate that this device is active and ready. This response is transmitted back to the IP address of the host sent the request. A valid ACTIVE response is a response which satisfies QUERY(FSN) = ACTIVE(FSN). ACTIVE frames received by a host with unexpected FSN must be silently discarded (ignored).

RST

Name	Opcode	Opdata	Direction
RST	0x10	0x00	Device to Host

This frame is sent by the device as a response to a REQUEST frame in order to indicate that the device needs to be (re)synchronized before accepting requests. It is a negative acknowledgement, ie the request received has been discarded. The host must repeat the synchronization process and then resend the request. A later paragraph describes the synchronization process in detail. There are two cases when the device responds with RST: a) when the device receives a request frame with unexpected FSN, and b) when the device receives a request in unsynchronized state.

SYNC

Name	Opcode	Opdata	Direction
SYNC	0x11	0x00	Host to Device

This frame is sent by the host to cause the (re)synchronization of the device. The FSN of this frame, when received by the device, is marked as the IFSN (Initial FSN). The FSN of the next REQUEST frame must be IFSN+1.
The device receiving a SYNC request, responds with ACK(IFSN).

ACK

Name	Opcode	Opdata	Direction
ACK	0x12	0x00	Device to Host

This frame is sent by the device as a positive acknowledgement to a SYNC frame. The host receiving this frame in the synchronization process, must assume the connection synchronized.

NAK

Name	Opcode	Opdata	Direction
NAK	0x13	0x00	Both directions

The NAK frame is used in any communication side in order to indicate failure of the data packet in frame validation. NAK must NOT be sent by the host when header checksum does not validate. This should be sent only if the data checksum is incorrect, provided that the header checksum is correct.

REQUEST

Name	Opcode	Opdata	Direction
REQUEST	0x21	0x00	Host to Device

This frame is sent by the host and it is the actual data bearing frame, containing the request data to be executed. The FSN of this frame must be set to Last FSN plus one, or to Initial FSN plus one if this is the first frame after synchronization process. Upon successful reception of a REQUEST frame and transmission of its REPLY frame, the device renews its synchronization timer.

REPLY

Name	Opcode	Opdata	Direction
REPLY	0x22	0x00	Device to Host

This frame is sent by the device to host as a result to a REQUEST frame, usually containing the results of the executed request. Upon

reception of a valid REPLY frame from the device, the host must advance its next expected FSN by one. Additionally the host must renew its synchronization timer.

OPDATA
======
An 8-bit (BYTE) value that accompanies the opcode as additional frame description. When not required, this value must be set to zero and will be set to zero if the frame is sent by the device.

DATA LENGTH
===========
A 16-bit value indicating the number of octets that follow the SDNP header ie the data packet length. Usually, in control frames, the data length is zero. This size does NOT include the size of the header.

DATA CHECKSUM
=============
A 16-bit value which is the 16-bit sum of all octets contained in the data packet, plus the value 0xAA55. When a data packet does not verify correctly by the device, a NAK control frame will be transmitted back to host to indicate reception failure and trigger retransmission as soon as possible.
Same approach must be performed by the host on data validation failure. When the data section is missing (data length = 0), the data checksum must be set to zero. When the host receives a frame with no data (data length = 0), this check must not be performed.

HEADER CHECKSUM
===============
A 16-bit value which is the 16-bit sum of the first 10 header octets, plus the value 0xAA55. A frame that does not validate correctly using the header checksum mechanism MUST be silently discarded as if it was never received.

7.3.2. Data link operations

For a better understanding of all SDNP protocol operations, we can define the following procedures which are analyzed in steps in this specification using pseudocode. These procedures are defined as a host implementation of the SDNP protocol. For this purpose, we must use some resources, such as timers, counters, etc. Also, a 'connection' structure is assumed that holds the required resources during the communication lifetime, as follows:

```
  STRUCTURE Connection
    IPADDRESS     DeviceAddress; //   IP address of the device
```

```
  VALUE          state;          //   UNSYNCHRONIZED or SYNCHRONIZED
  TIMER          SYNC;           // Timer holding connection
                                     expiration
  WORD           NextFSN;        // Next FSN to be used in REQUEST
                                    frames
  END
```

Query procedure
==========

It is the process of discovering all (or one) device(s) in the network.
Upon completion, the calling entity can have a list of the IP addresses of the devices that are able to be addressed by means of the SDNP protocol.
Minor modifications to this procedure can be performed in order to query a device that is in a known IP address.

```
SDNP_QUERY()
1   Clear device ip address list;
2   Set a timer T0 to 3 or more seconds;
3   Set a timer T1 to 100 milliseconds;
4   Select an arbitrary FSN;
5   Send QUERY(FSN) frame to the broadcast IP address;
6   Do until T0 expires:
7     If valid frame received then:
8       If frame type is ACTIVE then:
9         If ACTIVE(FSN) = QUERY(FSN) then:
10          Add IP address of sender to device list;
11        End;
12      End;
13      Discard frame;
14    End;
15    If T1 expired then;
16      Send QUERY(FSN) frame to the broadcast IP address;
17      Renew T1 timer;
18    End;
19  End;
END SDNP_QUERY()
```

Synchronization procedure
==========

The synchronization procedure establishes the initial FSN with the device. If the synchronization fails, the connection is broken, ie the device cannot be addressed and requests/replies cannot be exchanged. Upon procedure success, the request/reply exchange will succeed.

```
SDNP_SYNC()
1  Set connection state to UNSYNCHRONIZED;
2  For at least 6 times do:
3    Set timer T0 to 500 milliseconds;
```

```
  4    Select a random initial FSN (IFSN);
  5    Send SYNC(IFSN) frame to connection IP address;
  6    Do until T0 expires:
  7      If a valid frame received then:
  8        If frame type is ACK
  9          If ACK(FSN) = IFSN then:
  10           Set connection NextFSN = IFSN + 1;
  11           Set connection state to SYNCHRONIZED;
  12           Set connection SYNC timer to 4 seconds;
  13           Return sync success;
  14         End;
  15       End;
  16       Discard frame;
  17     End;
  18   End;
  19 End;
  20 Return sync failure;
  END SDNP_SYNC()
```

Send request procedure

==========

The send request procedure describes all steps needed for a single frame exchange process. Upon successful completion of this process, the reply is available to the calling entity.

```
  SDNP_SEND_REQUEST(IN STR RequestDataPacket, OUT STR
ReplyDataPacket)
  1  For at least 6 times do:
  2    If state is UNSYNCHRONIZED or connection SYNC timer expired
then:
  3      Perform SDNP_SYNC() procedure
  4      If failed then return request failure;
  5    End;
  6    Send REQUEST(Connection's NextFSN) using 'RequestDataPacket';
  7    Set T0 timer to 800 milliseconds;
  8    Do until T0 expires:
  9      If a valid SDNP frame received then do:
  10       If received frame's FSN <> Request frame's FSN
  11         Discard frame;
  12       Else
  13         Test received frame's opcode;
  14         Case RST:
  15           Set connection's state to UNSYNCHRONIZED;
  16           goto step 2;
  17         End;
  18         Case NAK:
  19           goto step 6;
  20         End;
  21         Case REPLY:
```

```
  23           If received frame's data packet does not validate
okay then:
  24           Create and send NAK frame with FSN set to received
FSN;
  25           Else
  26             Extract data packet from reply frame to
'ReplyDataPacket';
  27             Renew connection's SYNC timer;
  28             Advance connection's NextFSN by one;
  29             Return request transmittion success;
  30           End;
  31         End;
  32         Case all others:
  33           Discard frame;
  34         End;
  35       End;
  36     End;
  37   End;
  38 End;
  39 Return request transmittion failure;
  END SDNP_SEND_REQUEST()
```

Frame verification procedure

==========

This procedure performs basic checking in received frames from UDP socket.

Every frame reception implementation must perform this process.

```
  SDNP_FRAME_CHECK()
  1   If not in SDNP_QUERY() process then:
  2     If IP address of frame sender <> Connection's IP address
then:
  3       Discard frame and return failure;
  4     End;
  5   End;
  6   If size of UDP frame < size of SDNP header then:
  7     Discard frame and return failure;
  8   End;
  9   If size of UDP frame > 512 then:
  10    Discard frame and return failure;
  11  End;
  12  If SDNP header checksum does not validate okay then:
  13    Discard frame and return failure;
  14  End;
  15  If UDP frame size <> SDNP header data length +  SDNP header
size then:
  16    Discard frame and return failure;
  17  End;
  18  If frame id in SDNP header <> SDNP device protocol id then:
  19    Discard frame and return failure;
```

```
20  End;
21  Return success;
END SDNP_FRAME_CHECK()
```

8. Command protocol

The command protocol is initiated by the host computer, when the host wants to instruct the device to process a specific command. Due to the number of commands this layer supports, they can be grouped as:

- Request information commands
- Setup commands
- Fiscal printer commands
- System commands

8.1. Command protocol packets

In the command protocol there are always both packets present in the communication: the request packet and the reply packet. The general form of the request and reply packets follow this model:

Request packet: [Request code] <[Request data]>
Reply packet: [Reply code] / [device status] / [fiscal status]/ <[Reply data]>

In request packets, the request data are not always required (notice that 'request data' are inside <>). Additionally in reply packets, the reply data are not always present. All other sections are always present.

8.1.1. A more detailed form of command protocol request packet

Data		
	Optional Section	
Request code	Field 1 / Field 2 / Field 3 / ... / Field N	

This defines 2 sections of a request packet:

- The request code section
- The data field section

8.1.1.1. Request code

In online protocol packets we dealt with 'packet descriptor' which was a special field for identifying the packet type. In command protocol, the first field is called 'request code' and has the same functionality, although the request code is now sent to the ECR/POS rather than received by it. The request code is always a simple STRING field of one-character fixed length.

8.1.1.2. Request packet data fields

Data fields are not always required in all command's request packets. When not a requirement, data fields section is totally omitted, and the checksum section follows directly after the request code.

8.1.2. A more detailed form of command protocol reply packet

Packet Data			
		Optional Section	
Reply code	Status	Field 1 / Field 2 / ... / Field N	

This defines 3 sections of a reply packet:

- The reply code section
- The status section (device & fiscal)
- The data field section

8.1.2.1. Reply code section.

Reply code is a single numeric field of 2 hexadecimal characters identifying the result of the command execution by the ECR/POS. A zero-reply code ('00') indicates that the command has been executed successfully. A non zero reply code indicate an Error in command execution. Error codes returned are explained in detail in a later section. Receiving a nonzero reply code means that the command has NOTbeen executed. Receiving a zero-reply code means that the command has been or will be successfully executed. Commands that require very little time to executed, such as information retrieve, will be executed before the reply packet is transmitted. This is because the reply packet data fields depend on the command execution itself. Commands that take long time to execute, such as report issuing, will be only checked, a reply packet will be sent, and then will be executed.

8.1.2.2. Status section

Status is a section consisting of two numeric 2-character hexadecimal fields:

Device status	Fiscal status

Status section is returned by the ECR/POS to reflect the hardware & fiscal firmware states which must be considered by the host application.

8.1.2.2.1. Device status

Device status informs the host application of some hardware related events of the ECR/POS. The byte that this field forms must be mapped in bits in this way:

MSB							**LSB**
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
---	BATWARN	FDISC	IN MENU	PCONN	PP.END	SD DISC	BUSY

Bit 0: Device busy
This bit when set to '1' indicates that the ECR/POS is currently busy executing a previous command or other task. When busy, the ECR/POS may execute some noncritical commands and refuse to execute others replying an Error 'Device busy -- Unable to execute' (See Error codes).
The host must check this bit (requesting a 'status') before issuing any critical commands, or, must keep sending the command until the command is executed (or failed by other reason). BUSY state is a temporary state but, due to very different tasks the ECR/POS may cause the BUSY state, the time which the BUSY flag will be found set is varying from a few milliseconds to few minutes. A host may inform the user after (for example) one minute that the device is busy in other task and ask for a 'retry' or 'cancel' of the requested operation. An example in which a BUSY flag will be set for long time is a fiscal report issuing: When the host (or the ECR/POS user) requests a fiscal report with many records, the report will take long time to finish, thus keeping the BUSY flag set for long. It is highly recommended though that a host should NOT produce a 'device busy' Error message to the application user before (at least) twenty (20) seconds. It is also recommended that the host application must allow the user to cancel or retry the operation.

Bit 1: SD Disconnection.
This bit indicates that (when set to one) that Electronic Journal SD is disconnected.

Bit 2: Printer Paper End
This bit indicates (when set to one) that the printer is out of paper, and must be replaced before the previous task has completed its printing duty. Usually, when this flag is set, the 'device busy' flag may be set also, if a previous command that used the printer caused the paper end Error. So, it is recommended that the paper end bit MUST be checked before the busy bit. Host application may inform the user of the need to insert a new role of paper to the printing mechanism. After doing so, this bit will be cleared and the command (that detected the paper end) may be retransmitted normally.

Bit 3: Printer offline
This bit indicates (when set to one) that the printing device is not responding to printing commands. Recommended action is to power off the printer and on again and retry the command. If the problem persists, the ECR/POS needs to be serviced.

Bit 4: In Menu
This bit indicates (when set to one) that the ECR/POS is in Menu. Recommended action is to press Cancel in order to move in Issue Signatures.

Bit 5: Fiscal disconnection
This bit indicates (when set to one) that the printing device is not responding to printing commands. The fiscal is disconnected. When this happens, the ECR/POS is unable to issue receipts, reports of any kind except the fiscal periodical report. Recommended action is to call Technician.

Bit 6: Battery warning
This bit indicates (when set to one) that the printing device is not responding to printing commands. Recommended action is to power off the printer and on again and retry the command. If the problem persists, the ECR/POS needs to be serviced.

Bit 7: (Reserved)

Example: Assume device status field is '41'. This hexadecimal value, when converted to binary will be '00010001'. The '1's mean that the printer is offline (bit 4) and the device is busy (bit 0).

8.1.2.2.2. Fiscal status

Fiscal status is a 2-digit numeric hexadecimal field which informs the host about several states of the fiscal firmware inside the ECR/POS. The byte that this field forms must be mapped in bits in this way:

MSB							**LSB**
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FFULL	MAX INV	---	---	---	SIGNOPEN	DAYOPEN	---

Bit 0: (Reserved)

Bit 1: Day is open
This flag indicates that there is an open day in the ECR/POS. This means that one or more receipts or reports have been issued after a Z clearing report. The day open flag will be zero after the issuing of a Z report and before printing anything else, reports or receipts. A 'day' is defined in the fiscal firmware as the period between two Z closures.

Bit 2: Signature Open
This flag is indicating that a signature is currently in 'open' state in the ECR/POS.

Bit 3: (Reserved)

Bit 4: (Reserved)

Bit 5: (Reserved)

Bit 6: Max Daily Invoices
This flag indicates that ECR/POS already issued the max daily invoices. Recommended action is to issue Z report.

Bit 7: Fiscal file full
This bit indicates (when set to one) that the printing device is not responding to printing commands. The fiscal file used to store daily data after a 'Z' closure report is now full. When this happens, the ECR/POS is unable to issue receipts, reports of any kind except the fiscal periodical report. So, when the host detects this, it must not try to issue receipts or do any other printing.

Example: Assume fiscal status field is '16'. This hexadecimal value, when converted to binary will be '00010110'. The '1's mean that the ECR/POS has a day in open state (bit 1), a receipt is open (bit 2) and the open receipt is in payment state (bit 3).

8.2. Command packets groups

8.2.1. Read Device identification [a]

This command returns information about which ECR/POS unit is communicating with.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**a**	1 (Counting request code)	0 (Without request code)	"a/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'a' for this command.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, device status & fiscal status)	1 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "AAA00000001/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	ECR Registration number	INTEGER	Default	It is the official ECR/POS registration number and it is not programmable. This number is unique to each ECR/POS.
	FIELD 2	ECR number	INTEGER	1-2 digits	The programmable number of the ECR/POS assigned.
	FIELD 3	Registration owner letters	STRING	Fixed, 2 chars	The ECR's registration characters.
	FIELD 4	ECR MODEL	STRING	Default	The ECR's Model.
	FIELD 5	ECR Firmware Version	STRING	Default	The ECR's Firmware version.

8.2.2.Read version/device info [v]

This command will return version information for protocol and firmware of the ECR/POS. Also returns the device capabilities.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**v**	2 (Counting request code)	0 (Without request code)	"v/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'v' for this command.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	12 (Counting reply code, device status & fiscal status)	9 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "MAT /RBS 101-FMS(DMU)/V1 R1 T24_30-04-2024/6/2/5/8/6/10/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	MANUFACTURER	STRING	Default	MAT
	FIELD 2	ECR MODEL	STRING	Default	The ECR's Model.
	FIELD 3	Firmware Version/Date	STRING	Default	The ECR/POS firmware version and firmware date separated with '_'.
	FIELD 4	Key's total number	INTEGER	1-2 digits	Maximum keys
	FIELD 5	LCD lines	INTEGER	1 digit	LCD lines
	FIELD 6	Total VAT's number	INTEGER	1-2 digits	Maximum VAT categories
	FIELD 7	Total Header Lines	INTEGER	1-2 digits	Header lines

	FIELD 8	Total Footer Lines	INTEGER	1-2 digits	Footer lines
	FIELD 9	Total payment's number	INTEGER	1-2 digits	Maximum payments

8.2.3. Read device parameters [s]

This command will return various ECR/POS parameters programmed.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**s**	1	0	"s"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 's' for this command.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	25 (Counting reply code, device status & fiscal status)	22 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "111100/2/MHXANH-1 /997981320/1/5/6/7/2/1/1/1/9101/192.168.0.150/255.255.255.0/192.168.0.1/192.168.0.1/192.168.0.1/0.0.0.0/1/5000/4 /"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Setup Flags allowed flag value 0: Disable 1: Enable 2: Skip	FLAGS	Fixed, 6 digits	The flag settings are mapped as follows (left to right) 1st = Date Mode 2nd = Print Header/Footer when print sign short doc 3rd = Print Short Doc. after Sign 4th = Print Total Qty (Only when Device Print Sign Receipt) 5th = Print VAT Rates Analysis (Only when FHMAS Print Sign Receipt)

					6th = Print Machine info (Only when FHMAS Print Sign Receipt)
	FIELD 2	Device No	INTEGER	Default	Device Number
	FIELD 3	Device Description	STRING	1-15 chars	Device Description
	FIELD 4	Owner TIN	STRING	Fixed 9	TAX ID
	FIELD 5	Serial Port Usage	INTEGER	Default	0 - No Used 1 - CMD Protocol 2 - External Serial Printer
	FIELD 6	Serial Port Baud Rate	INTEGER	Default	0 - 9600 1 - 19200 2 - 38400 3 - 57600 4 - 115200 (Default) 5 - 230400
	FIELD 7	Print Tone	INTEGER	1-2 digits	Range 1-10
	FIELD 8	Print Spacing	INTEGER	1-2 digits	Range 1-20
	FIELD 9	Company Type	INTEGER	Default	Future Use
	FIELD 10	Ethernet	INTEGER	Default	0 = disable 1 = enable
	FIELD 11	DHCP	INTEGER	Default	0 = disable 1 = enable
	FIELD 12	Protocol	INTEGER	Default	0 = UDP 1 = TCP
	FIELD 13	Port	INTEGER	1-5 digits	Communication Port (Default 9101)
	FIELD 14	Static IP	STRING	1-15	Static IP
	FIELD 15	Mask	STRING	1-15	Mask
	FIELD 16	Gateway	STRING	1-15	Gateway
	FIELD 17	DNS1	STRING	1-15	DNS1
	FIELD 18	DNS2	STRING	1-15	DNS2

	FIELD 19	Remote IP	STRING	1-15	Remote IP
	FIELD 20	Protocol in Ethernet	INTEGER	Default	0 – Disable 1 – Enable
	FIELD 21	Safe Time delay after Sign	INTEGER	1-5 digits	Future Use
	FIELD 22	Fee Category	INTEGER	Default	0 – No Fee 1 – Παρεπιδημούντων 2 – Εστίαση 3 – Κέντρα Διασκέδασης 4 – Καζίνο 5 – Λοιπά

8.2.4. Program Device Parameters [S]

This command programs various 'setup' information in the Device. All fields are optional so the host can selectively modify specific fields.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**S**	23 (Counting request code)	22 (Without request code)	Set TIN: "S////997981320//////////////////" SetEthernet/DHCP/TCP/PROTOCOLINETHERNET: "S/////////1/1/1/9101///////1///" SetEnableShotDoc.PrintafterSign: "S/221222/////////////////////" SetFeeCategoryCasino: "S/////////////////////4/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Setup Flags allowed flag value 0:Disable 1:Enable 2:Skip	FLAGS	Fixed, 6 digits	The flag settings are mapped as follows (left to right) 1st = Date Mode 2nd = Print Header/Footer when print sign short doc 3rd = Print Short Doc. after Sign 4th = Print Total Qty (Only when Device Print Sign Receipt) 5th = Print VAT Rates Analysis (Only when FHMAS Print Sign Receipt)

					6th = Print Machine info (Only when FHMAS Print Sign Receipt)
	FIELD 2	Device No	INTEGER	Default	Device Number
	FIELD 3	Device Description	STRING	1-15 chars	Device Description
	FIELD 4	Owner TIN	STRING	Fixed 9	TAX ID
	FIELD 5	Serial Port Usage	INTEGER	Default	0 - No Used 1 - CMD Protocol 2 - External Serial Printer
	FIELD 6	Serial Port Baud Rate	INTEGER	Default	0 - 9600 1 - 19200 2 - 38400 3 - 57600 4 - 115200 (Default) 5 - 230400
	FIELD 7	Print Tone	INTEGER	1-2 digits	Range 1-10
	FIELD 8	Print Spacing	INTEGER	1-2 digits	Range 1-20
	FIELD 9	Company Type	INTEGER	Default	Future Use
	FIELD 10	Ethernet	INTEGER	Default	0 = disable 1 = enable
	FIELD 11	DHCP	INTEGER	Default	0 = disable 1 = enable
	FIELD 12	Protocol	INTEGER	Default	0 = UDP 1 = TCP
	FIELD 13	Port	INTEGER	1-5 digits	Communication Port
	FIELD 14	Static IP	STRING	1-15	Static IP
	FIELD15	Mask	STRING	1-15	Mask
	FIELD 16	Gateway	STRING	1-15	Gateway
	FIELD 17	DNS1	STRING	1-15	DNS1

	FIELD 18	DNS2	STRING	1-15	DNS2
	FIELD 19	Remote IP	STRING	1-15	Remote IP
	FIELD 20	Protocol in Ethernet	INTEGER	Default	0 - Disable 1 - Enable
	FIELD 21	Safe Time delay after Sign	INTEGER	1-5 digits	Future Use
	FIELD 22	Fee Category	INTEGER	Default	0 - No Fee 1 - Παρεπιδημούντων 2 - Εστίαση 3 - Κέντρα Διασκέδασης 4 - Καζίνο 5 - Λοιπά

	TOTAL FIELD COUNT	**DATA FIELD COUNT**	**EXAMPLE REPLY**
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.5. Read GSIS settings [,]

Read the programmable GSIS settings

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	,	3	2	",/3//"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request string	STRING	Fixed, 5 character	Must be ',/3//' for this command.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	10 (Counting reply code, device status & fiscal status)	7 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "1/0/0/http:~~wldcl.ece.ntua.gr~myweb~websend.php/80/A349970610D0257DB18C6BA9C796A8096D4960B79CF579DF9E735963AE3E41FC/http:~~wldcl.ece.ntua.gr~myweb~q1.php/

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Activate Send	INTEGER	Default	0- Disable 1 - Enable
	FIELD 2	Reserved	INTEGER	Default	Reserved
	FIELD 3	Reserved	INTEGER	Default	Reserved
	FIELD 4	GSIS Url or IP	STRING	1-80	GSIS Url or IP
	FIELD 5	Port	INTEGER	1-2 digits	GSIS Server Port
	FIELD 6	Key	STRING	Fixed 64 chars	AES Key
	FIELD 7	QRCode	STRING	1-80	QRCode URL

8.2.6. Program GSIS Settings []

This command programs GSIS settings. All fields are optional so the host can selectively modify specific fields.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**]**	8 (Counting request code)	7 (Without request code)	"]/1///http:~~wldcl.ece.ntua.gr~myweb~websend.php/80/2020202020202020202020202020202020202020202020202020202020202020/http:~~wldcl.ece.ntua.gr~myweb~q1.php/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Activate Send	INTEGER	Default	0 - Disable 1 - Enable
FIELD2	Reserved	INTEGER	Default	Reserved
FIELD 3	Reserved	INTEGER	Default	Reserved
FIELD 4	GSIS Url or IP	STRING	1-80	GSIS Url or IP
FIELD 5	Port	INTEGER	1-2 digits	GSIS Server Port
FIELD 6	Key	STRING	Fixed 64 chars	AES Key
FIELD 7	QRCode	STRING	1-80	QRCode URL

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.7. Read Header info [h]

This command will return all information about the header.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**h**	1 (Counting request code)	0 (Without request code)	"h/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'h' for this command.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	21 (Counting reply code, device status & fiscal status)	18 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "4/HEADER LINE 1 ////////////////0/50/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Header line printing types	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 2	Header line 1	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 3	Header line printing types	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing 2 = Double height 3 = Double width

					4 = Double width and height 5 = Bold
	FIELD 4	Header line 2	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 5	Header line printing types	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 6	Header line 3	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 7	Header line printing types	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 8	Header line 4	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 9	Header line printing types	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold

	FIELD 10	Header line 5	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 11	Header line printing types	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 12	Header line 6	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 13	Header line printing types	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 14	Header line 7	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 15	Header line printing types	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 16	Header line 8	STRING	0-32	The text data for each line.

				(0-16) chars	Only 16 characters can be printed in case of double width characters
	FIELD 17	Header Changes	INTEGER	1-3 digits	Total Header Changes
	FIELD 18	Remain Header Changes	INTEGER	1-3 digits	Remaining header Changes

8.2.8. Program header [H]

This command programs the header in the Device. Lines that will not be passed in the command will not be printed. To program a blank line, the host must pass the line filled with spaces. The lines provided for header will NOT be centred automatically.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**H**	17 (Counting request code)	16 (Without request code)	"H/1/LINE1/1/LINE2/1/LINE3/1/LINE4/1/LINE5/1/LINE6/1/LINE7/1/LINE8/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'H' for this command.
FIELD 2	Header line printing types	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
FIELD 3	Header line 1	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
FIELD 4	Header line printing types	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height

					5 = Bold
	FIELD 5	Header line 2	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 6	Header line printing types	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 7	Header line 3	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 8	Header line printing types	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 9	Header line 4	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 10	Header line printing types	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing

					2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 11	Header line 5	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 12	Header line printing types	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 13	Header line 6	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 14	Header line printing types	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 15	Header line 7	STRING	0-32 (0-16) chars	The text data for each line.

					Only 16 characters can be printed in case of double width characters
	FIELD 16	Header line printing types	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 17	Header line 8	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.9. Read Footer info [K]

This command will return all information about the header.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**K**	1 (Counting request code)	0 (Without request code)	"K/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'h' for this command.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	16 (Counting reply code, device status & fiscal status)	13 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "1/ΕΥΧΑΡΙΣΤΟΥΜΕ - THANK YOU ////////////1/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Footer line printing types	INTEGER	0-1 digits	The printing type for each Footer line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 2	Footer line 1	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 3	Footer line printing types	INTEGER	0-1 digits	The printing type for each Footer line as: 1 = Normal printing 2 = Double height 3 = Double width

				4 = Double width and height 5 = Bold
FIELD 4	Footer line 2	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
FIELD 5	Footer line printing types	INTEGER	0-1 digits	The printing type for each Footer line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
FIELD 6	Header line 3	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
FIELD 7	Footer line printing types	INTEGER	0-1 digits	The printing type for each Footer line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
FIELD 8	Footer line 4	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
FIELD 9	Footer line printing types	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold

	FIELD 10	Footer line 5	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 11	Footer line printing types	INTEGER	0-1 digits	The printing type for each Footer line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 12	Footer line 6	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 13	Active Footer Lines	INTEGER	0-1 digits	Number of active footer lines

8.2.10. Program footer [F]

This command programs the header in the Device. Lines that will not be passed in the command will not be printed. To program a blank line, the host must pass the line filled with spaces. The lines provided for header will NOT be centred automatically.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**F**	13 (Counting request code)	12 (Without request code)	"F/1/LINE1/1/LINE2/1/LINE3/1/LINE4/1/LINE5/1/LINE6/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'F' for this command.
FIELD 2	Footer line printing types	INTEGER	0-1 digits	The printing type for each Footer line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
FIELD 3	Footer line 1	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
FIELD 4	Footer line printing types	INTEGER	0-1 digits	The printing type for each Footer line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height

					5 = Bold
	FIELD 5	Footer line 2	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 6	Footer line printing types	INTEGER	0-1 digits	The printing type for each Footer line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 7	Footer line 3	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 8	Footer line printing types	INTEGER	0-1 digits	The printing type for each Footer line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 9	Footer line 4	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 10	Footer line printing types	INTEGER	0-1 digits	The printing type for each Footer line as: 1 = Normal printing

					2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 11	Footer line 5	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters
	FIELD 12	Footer line printing types	INTEGER	0-1 digits	The printing type for each Footer line as: 1 = Normal printing 2 = Double height 3 = Double width 4 = Double width and height 5 = Bold
	FIELD 13	Footer line 6	STRING	0-32 (0-16) chars	The text data for each line. Only 16 characters can be printed in case of double width characters

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.11. Program EFTPOS Parameters [\/11/]

This command programs the required parameters for EFTPOS Communication in the Device.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	\	19 (Counting request code)	18 (Without request code)	"\/11/1/1/7900/180/172.21.20.64/EFTPOS1/1/0////1111111111/0/0/0/0/0/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request Code	STRING	Fixed, 1 character	Must be '\' for this command.
FIELD2	Parameter Type	INTEGER	2 digits	Always '11'
FIELD 3	Index	INTEGER	1-2 digits	Index of total 12 EFTPOS configurations
FIELD 4	Activate	INTEGER	0-1 digits	0 - No Disable 1 - Local Network 2 - Remote Middleware
FIELD 5	Port	INTEGER	1-5 digits	Listening port of EFTPOS
FIELD 6	Timeout	INTEGER	1-3 digits	10-240 seconds timeout for EFTPOS response
FIELD 7	IP	STRING	1-15	IP address for local EFTPOS or remote Middleware server
FIELD 8	Description	STRING	1-25	Name Description of The EFTPOS
FIELD 9	Protocol Variant	INTEGER	0-1 digits	0 = EFTPOS PRINTING 1 = ECR PRINTING
FIELD 10	Main BMP Index	INTEGER	0-1 digits	1-9 BMP pre-programmed bitmaps if the protocol variant is 2
FIELD 11	Reserved	INTEGER	Default	Reserved

	FIELD 12	ACQ ID	STRING	1-5	ACQ ID When Middleware is Enabled
	FIELD 13	POS TID	STRING	1-5	EFTPOS TID When Middleware is Enabled
	FIELD 14	Setup Flags allowed flag value 0: Disable 1: Enable	FLAGS	Fixed, 10 digits	The flag settings are mapped as follows (left to right) 1st = Sale 2nd = Sale with instalments 3rd = Refund 4th = Void 5th = Pre-approval registration 6th = Mail order 7-10 = Reserved
	FIELD15	2nd BMP Index	INTEGER	0-1 digits	1-9 BMP pre-programmed bitmaps if the protocol variant is 2
	FIELD 16	3rd BMP Index	INTEGER	0-1 digits	1-9 BMP pre-programmed bitmaps if the protocol variant is 2
	FIELD 17	4th BMP Index	INTEGER	0-1 digits	1-9 BMP pre-programmed bitmaps if the protocol variant is 2
	FIELD 18	5th BMP Index	INTEGER	0-1 digits	1-9 BMP Pre-programmed bitmaps if the protocol variant is 2
	FIELD 19	6th BMP Index	INTEGER	0-1 digits	1-9 BMP pre-programmed bitmaps if the protocol variant is 2

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.12. Delete EFTPOS Parameters [\/12/]

This command Deletes EFTPOS parameters from the Device.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	\	19 (Counting request code)	18 (Without request code)	"\/12/1/////////////////"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request Code	STRING	Fixed, 1 character	Must be '\' for this command.
	FIELD2	Parameter Type	INTEGER	2 digits	Always '12'
	FIELD 3	Index	INTEGER	1-2 digits	Index of total 12 EFTPOS configurations
	FIELD 4-19	Reserved	INTEGER	Default	Reserved

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.13. Check EFTPOS Connection [\/13/]

This command Checks EFTPOS Connection (And Pair if not yet).

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	\	19 (Counting request code)	18 (Without request code)	EFTPOS Pair Procedure: "\/13/1/0/////////////////" EFTPOS Check Connection: "\/13/1/////////////////"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request Code	STRING	Fixed, 1 character	Must be '\' for this command.
FIELD2	Parameter Type	INTEGER	2 digits	Always '13'
FIELD 3	Index	INTEGER	1-2 digits	Index of total 12 EFTPOS configurations
FIELD 4	ECHO or INIT EFTPos	INTEGER	0-1 digits	0 - Echoing EFTPos 1 - Init EFTPos
FIELD 5-19	Reserved	INTEGER	Default	Reserved

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.14. Get GSIS AES Key [\/14/]

This command Reads and sets the AES key from GSIS.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	\	19 (Counting request code)	18 (Without request code)	"\/14/////////////////"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request Code	STRING	Fixed, 1 character	Must be '\' for this command.
	FIELD2	Parameter Type	INTEGER	2 digits	Always '14'
	FIELD 3-19	Reserved	INTEGER	Default	Reserved

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.15. Get Last Z and Last Send in GSIS [\/15/]

This command reads the last z-report number and gets the last successful post in GSIS.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	/	19 (Counting request code)	18 (Without request code)	"\/15/////////////////"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request Code	STRING	Fixed, 1 character	Must be '\' for this command.
	FIELD2	Parameter Type	INTEGER	2 digits	Always '15'
	FIELD 3-19	Reserved	INTEGER	Default	Reserved

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	5 (Counting reply code, device status & fiscal status)	2 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "49/48/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Last Z-Report	INTEGER	1-4 digits	Last Z-REPORT of the device
	FIELD 2	Last GSIS Post	INTEGER	1-4 digits	Last Z-REPORT of the device that is successfully posted on the GSIS Server

8.2.16. Read EFTPOS Parameters [,]

This command will return all information about the header.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	,	3 (Counting request code)	2 (Without request code)	",/11/1/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be ',' for this command.
FIELD 2	Parameter	STRING	2 chars	Always '11'
FIELD 3	Index	INTEGER	1-2 digits	Index of total 12 EFTPOS configurations

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	20 (Counting reply code, device status & fiscal status)	17 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "1/192.168.20.64/7900/EFTPOS EMULATOR/1/180/0/DMK9901A/0///1111110000/0/0/0/0/0/

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Index	INTEGER	1-2 digits	Index of total 12 EFTPOS configurations
FIELD 2	Activate	INTEGER	0-1 digits	0 - No Disable 1 - Local Network 2 - Remote Middleware
FIELD 3	Port	INTEGER	1-5 digits	Listening port of EFTPOS
FIELD 4	Timeout	INTEGER	1-3 digits	10-240 seconds timeout for EFTPOS response
FIELD 5	IP	STRING	1-15	IP address for local EFTPOS or remote Middleware server
FIELD 6	Description	STRING	1-25	Name Description of The EFTPOS

	FIELD 7	Protocol Variant	INTEGER	0-1 digits	0 = EFTPOS PRINTING 1 = ECR PRINTING
	FIELD 8	Main BMP Index	INTEGER	0-1 digits	1-9 BMP pre-programmed bitmaps if the protocol variant is 2
	FIELD 9	Reserved	INTEGER	Default	Reserved
	FIELD 10	ACQ ID	STRING	1-5	ACQ ID When Middleware is Enabled
	FIELD 11	POS TID	STRING	1-5	EFTPOS TID When Middleware is Enabled
	FIELD 12	Setup Flags allowed flag value 0: Disable 1: Enable	FLAGS	Fixed,10 digits	The flag settings are mapped as follows (left to right) 1st = Sale 2nd = Sale with instalments 3rd = Refund 4th = Void 5th = Pre-approval registration 6th = Mail order 7-10 = Reserved
	FIELD 13	2nd BMP Index	INTEGER	0-1 digits	1-9 BMP pre-programmed bitmaps if the protocol variant is 2
	FIELD 14	3rd BMP Index	INTEGER	0-1 digits	1-9 BMP pre-programmed bitmaps if the protocol variant is 2
	FIELD 15	4th BMP Index	INTEGER	0-1 digits	1-9 BMP pre-programmed bitmaps if the protocol variant is 2
	FIELD 16	5th BMP Index	INTEGER	0-1 digits	1-9 BMP Pre-programmed bitmaps if the protocol variant is 2
	FIELD 17	6th BMP Index	INTEGER	0-1 digits	1-9 BMP pre-programmed bitmaps if the protocol variant is 2

8.2.17. Print fiscal report (Z to Z) [z]

This command prints Fiscal report from start Z number to end Z number.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**z**	4 (Counting request code)	3 (Without request code)	"z/155/166/0/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'z' for this command.
FIELD 2	Start Z Number	INTEGER	1 - 4 digits	The start Z number of report.
FIELD 3	End Z Number	INTEGER	1 - 4 digits	The end Z number of report.
FIELD 4	Type of printing report	INTEGER	0-1 Digit	0= Summary 1= Detailed 2= Only signatures

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.18. Print fiscal report (Date to Date) [f]

This command prints Fiscal report from start Date to end Date.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**f**	4 (Counting request code)	3 (Without request code)	"f/150213/160213/1/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'f' for this command.
FIELD 2	Start Date	DATE6	Fixed, 6 digits	The start date of report.
FIELD 3	End Date	DATE6	Fixed, 6 digits	The end date of report.
FIELD 4	Type of printing report	INTEGER	0-1 Digit	0= Summary 1= Detailed 2= Only signatures

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.19. Get Invoice info [=]

This command will return all information about Invoice requested By DATE/Z and by UID/Daily.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**=**	5 (Counting request code)	4 (Without request code)	**search by Z and UID:** "=/10//3a01dfb7-28d0-44ca-8d35-f9dc07f68e01//" **search by Date and UID:** "=//080224/3a01dfb7-28d0-44ca-8d35-f9dc07f68e01//" **search by Date and Daily Index:** "=//080224//1/" **search by Z and Daily Index:** "=/10///1/" **search by Z and Date and UID:** "=/10/080224/3a01dfb7-28d0-44ca-8d35-f9dc07f68e01//" **search by Z and Date and Daily Index:** "=/10/080224//1/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be '=' for this command.
FIELD 2	Z number	INTEGER	1-4 digits	Number off Z-Report
FIELD 3	Date	INTEGER	Fixed,6 digits	Date of Invoice with format "ddMMyy"
FIELD 4	UID	STRING	1-60	Unique Invoice ID
FIELD 5	Daily Index	INTEGER	1-4 digits	Daily Index of invoice

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4-11 (Counting reply code, device status	8 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "1/3a01dfb7-28d0-44ca-8435-f4dc07a11e29/1/210524/143600/31A2D3A582B62DCE36593DB27F4E7B14C806F37F/2652FA4DCCF0F7A34A254669BA37EB5C4B2552AF/"

	& fiscal status)				
		DESCRIPTION		**TYPE**	**LENGTH**	**NOTES**
		FIELD 1	Result	INTEGER	1 digit	0: Not Found 1: Found
		FIELD2	Daily Number of Signed document	INTEGER	1-6 digits	Daily Number of Signed document
		FIELD 3	Invoice Date	INTEGER	Fixed,8 digits	Invoice Date in the form “yyyyMMdd”
		FIELD 4	Invoice Time	INTEGER	Fixed, 4 digits	Invoice Issued Time in the form “hhmm”
		FIELD 5	Invoice Signature _b	STRING	1-40 chars	Invoice Signature _b
		FIELD 6	Invoice Signature _e	INTEGER	1-40 chars	Invoice Signature _e

8.2.20. Get Last Invoice Info [9]

This command will return all information about last issued Invoice.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**9**	1 (Counting request code)	0 (Without request code)	**"9/"**

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be '9' for this command.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3-13 (Counting reply code, device status & fiscal status)	10 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "01406/00001/20240521/1647/E4A04534B2B5B6D897711F37F413918B5D661E43/997981320;998927533;DMU77000002;;20240521164 7;1;1406;77;172;;100;0.00;0.00;8.06;0.00;0.00;0.00;0.00; 1.94;0.00;10.00;0;0;0.00;;(24.00,8.06,2,0);B3FF2796AD;3F 28CFFDCA78634D32C434AC3436F5F433877ECB /DMU77000002/0077/http:\\wldcl.ece.ntua.gr\myweb\q1.php? SIG=DMU770000020000014063F28CFFDCA78634D32C434AC3436F5240 521164710.00/3a01dfb7-28d0-44ca-8d35-f9dc07f68e0c/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Total Signed documents	INTEGER	1-6 digits	Total Signed documents
	FIELD2	Daily Signed documents	INTEGER	1-6 digits	Daily Signed documents
	FIELD 3	Invoice Date	INTEGER	Fixed,8 digits	Invoice Date in the form "yyyyMMdd"
	FIELD 4	Invoice Time	INTEGER	Fixed, 4 digits	Invoice Issued Time in the form "hhmm"
	FIELD 5	Invoice Signature _b	STRING	1-40	Invoice Signature _b
	FIELD 6	Invoice _e line data	INTEGER	1-230	Invoice _e line data

	FIELD 7	Device Serial Number	STRING	Fixed, 11 chars	Device Serial Number
	FIELD 8	Issued Z number	INTEGER	1-5	issued Z number
	FIELD 9	QR Code data	STRING	Default	QR Code data
	FIELD 10	UID	STRING	Default	UID of last Invoice

8.2.21. Read daily totals [0]

This command is used to read the daily totals accumulated in one day.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**0**	1 (Counting request code)	0 (Without request code)	"0/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be '0' for this command.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	31 (Counting reply code, device status & fiscal status)	28 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "0.00/0.00/88.66/0.00/0.00/0.00/0.00/21.34/0.00/0.00/0.00/0.00/0.00/0.00/0.00/0.00/0.00/0.00/85.00/0.00/0.00/0.00/3/0/0/3/0/55/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Daily NET A	AMOUNT	Default	Daily net sums belonging to VAT A category
	FIELD 2	Daily NET B	AMOUNT	Default	Daily net sums belonging to VAT B category
	FIELD 3	Daily NET C	AMOUNT	Default	Daily net sums belonging to VAT C category
	FIELD 4	Daily NET D	AMOUNT	Default	Daily net sums belonging to VAT D category
	FIELD 5	Daily NET E	AMOUNT	Default	Daily net sums belonging to VAT E category
	FIELD 6	Daily VAT A	AMOUNT	Default	Daily sums belonging to VAT A category
	FIELD 7	Daily VAT B	AMOUNT	Default	Daily sums belonging to VAT B category
	FIELD 8	Daily VAT C	AMOUNT	Default	Daily sums belonging to VAT C category

	FIELD 9	Daily VAT D	AMOUNT	Default	Daily sums belonging to VAT D category
	FIELD 10	Daily Credits Net A	AMOUNT	Default	Daily credits sum belonging to VAT A category
	FIELD 11	Daily Credits Net B	AMOUNT	Default	Daily credits sum belonging to VAT B category
	FIELD 12	Daily Credits Net C	AMOUNT	Default	Daily credits sum belonging to VAT C category
	FIELD 13	Daily Credits Net D	AMOUNT	Default	Daily credits sum belonging to VAT D category
	FIELD 14	Daily Credits Net E	AMOUNT	Default	Daily credits sum belonging to VAT E category
	FIELD 15	Daily Credits VAT A	AMOUNT	Default	Daily credits sum belonging to VAT A category
	FIELD 16	Daily Credits VAT B	AMOUNT	Default	Daily credits sum belonging to VAT B category
	FIELD 17	Daily Credits VAT C	AMOUNT	Default	Daily credits sum belonging to VAT C category
	FIELD 18	Daily Credits VAT D	AMOUNT	Default	Daily credits sum belonging to VAT D category
	FIELD 19	EFTPOS Daily total	AMOUNT	Default	EFTPOS Daily total sum
	FIELD 20	EFTPOS Daily Credit total	AMOUNT	Default	EFTPOS Daily credits total sum
	FIELD 21	Fees total	AMOUNT	Default	Fees total
	FIELD 22	Withholdings Tax Total	AMOUNT	Default	Withholdings Tax Total
	FIELD 23	Signatures	INTEGER	1-6 digits	Total number of issued invoices.
	FIELD 24	Number of Credits	INTEGER	1-6 digits	The total Number of Credit documents issued
	FIELD 25	Voids count	INTEGER	1-6 digits	The number of all voids during the day
	FIELD 26	EFTPOS Sales count	INTEGER	1-6 digits	The number of all EFTPOS Sale transactions during the day

	FIELD 27	EFTPOS credits count	INTEGER	1-6 digits	The number of all EFTPOS credit transactions during the day
	FIELD 28	Issued Z number	INTEGER	1-4 digits	Issued Z number

8.2.22. Read EFTPOS last Invoice info [5/99/]

This command is used to read the EFTPOS Transactions info of last Invoice.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**5**	8 (Counting request code)	7 (Without request code)	"5/99/////// "

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be '5' for this command.
FIELD 2	Parameter	INTEGER	1-2 digits	Always '99'
FIELD 3-8	Reserved	INTEGER	Default	Reserved

Reply in case of no available payment info or last payment in progress or error occurred before send to EFTPOS

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, device status & fiscal status)	1 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "1/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Status Of Payment	INTEGER	1 digit	0= No available last payment info 1= Last payment is in progress 2= Error occurred in last payment and before send request to EFTPOS (Field 2

					contains the error description)

Reply in case of successful payment with EFTPOS or error received from EFTPOS			
	TOTAL FIELD COUNT	**DATA FIELD COUNT**	**EXAMPLE REPLY**
REPLY PACKET	4-7 (Counting reply code, device status & fiscal status)	1-4 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) Example with 1 EFTPOS Successful payment: "1|QS9TRDIxRTlBL0Y1MDAwOjk3ODoyL0QyMDI0MDUyMTE2NTk0OC9SRE1VNzcwMDAwMDIvSDEvVDE0MDcvTTAvUTFGN0U0QTE2|00|01|50.00|Mastercard|123412******1234|RDMU7702|1|1|787032|133030119089|20240521165453|0|001|0|0.00|/" Example with 2 EFTPOS Successful payment: "1|QS9TNjVDMkM5L0YyMDAwOjk3ODoyL0QyMDI0MDUyMTE3MjA1NS9SRE1VNzcwMDAwMDIvSDEvVDE0MDgvTTAvUThBRjYyMjE3|00|01|20.00|Mastercard|123412******1234|RDMU7702|1|1|787032|133030119089|20240521171559|0|001|0|0.00|/2|QS9TOUJFMjhBL0YzMDAwOjk3ODoyL0QyMDI0MDUyMTE3MjEwMS9SRE1VNzcwMDAwMDIvSDEvVDE0MDgvTTAvUUU4MTk3RkY0|00|01|30.00|Mastercard|123412******1234|RDMU7702|1|1|787032|133030119089|20240521171605|0|001|0|0.00|/" Example with 1 EFTPOS Unsuccessful payment: "1|QS9TRDFFMUIxL0Y1MDAwOjk3ODoyL0QyMDI0MDUyMTE3NDkxNy9SRE1VNzcwMDAwMDIvSDEvVDE0MDkvTTAvUTIyQzBCNTBF|33|01|||||||||||||TRANSACTION WAS DECLINED/"
	Field 1 may be repeated for each successful payment (maximum 4 payments per invoice. The sub-fields separated with '|' character.		

	DESCRIPTION	**TYPE**	**LENGTH**	**NOTES**
SUBFIELD 1	Index of EFTPOS payment result	INTEGER	1 digit	Index of EFTPOS payment result (1-4)

SUBFIELD 2	Command sent in EFTPOS	STRING	Default	The command sent in EFTPOS in base64 url encoded format
SUBFIELD 3	rsp-code/Error Code	INTEGER	1-2 digits	Response code received from EFTPOS OR Error Code received from EFTPOS
SUBFIELD 4	EFTPOS Index	INTEGER	1-2 digits	Index of total 12 EFTPOS configurations
SUBFIELD 5	amount-final	AMOUNT	Default	amount-final (Empty on Error)
SUBFIELD 6	card-type	STRING	Default	card-type (Empty on Error)
SUBFIELD 7	cardpan-masked	STRING	Default	cardpan-masked (Empty on Error)
SUBFIELD 8	terminalId	STRING	Default	terminalId (Empty on Error)
SUBFIELD 9	batch-num	STRING	Default	batch-num (Empty on Error)
SUBFIELD 10	stan	STRING	Default	stan (Empty on Error)
SUBFIELD 11	authcode	STRING	Default	authcode (Empty on Error)
SUBFIELD 12	rrn	STRING	Default	rrn (Empty on Error)
SUBFIELD 13	trans-datetime	STRING	Default	trans-datetime (Empty on Error)
SUBFIELD 14	txn-type	STRING	Default	txn-type (Empty on Error)
SUBFIELD 15	bankId	STRING	Default	bankId (Empty on Error)
SUBFIELD 16	txn-ecr-status	STRING	Default	txn-ecr-status (Empty on Error)
SUBFIELD 17	amount-tip	AMOUNT	Default	amount-tip (Empty on Error)
SUBFIELD 18	Error description	STRING	Default	Error description (Empty on Success)

8.2.23. Upload/Check/Print Bitmap [|]

This command uploads, checks or print a B/W bitmap in the Device.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	|	4 (Counting request code)	3 (Without request)	"|/0/1/8254"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be '|' for this command.
FIELD 2	Type	INTEGER	0-1 digits	0 = Start Upload Bitmap 1 = Check if Bitmap uploaded successfully 2 = Print a Bitmap
FIELD 3	Bitmap Index	INTEGER	0-1 digits	Bitmap Index position 1-9
FIELD 4	Bytes	INTEGER	1-5 digits	Bitmap Total Bytes: >1 and <20000 bytes After Command "|/0" you must send Bitmap raw binary data until reach the total number of bytes

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.24. Program Bmp Position [~]

This command programs the position of Bmp.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	~	4 (Counting request code)	3 (Without request code)	"~/1/2/1/

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be '~' for this command.
	FIELD 2	Bitmap Index	INTEGER	0 - 1 digits	The index of bitmap to setup
	FIELD 3	Bitmap position	INTEGER	0 - 1 digits	0: top 1: bottom 2: disable bitmap
	FIELD 4	Bitmap quality	INTEGER	0 - 1 digits	0: normal 1: double height

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.25. Read Bmp information [I]

This command reads all information about a programmed Bmp.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**I**	2 (Counting request code)	1 (Without request code)	"I/1/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'I' for this command
FIELD 2	Bitmap Index	INTEGER	0 - 1 digits	The index of bitmap to read

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	6 (Counting reply code, device status & fiscal status)	3 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "1/0/1/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Bitmap Index	INTEGER	0 - 1 digits	The index of bitmap to read
FIELD 2	Bitmap position	INTEGER	0 - 1 digits	0: top 1: bottom 2: disable bitmap
FIELD 3	Bitmap quality	INTEGER	0 - 1 digits	0: normal 1: double height

8.2.26. Program payment type [Y]

This command programs a payment type in the Device. All fields except the request code and the payment number are optional. When not provided, the information in the payment type will not be updated.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**Y**	6 (Counting request code)	5 (Without request code)	"Y/2/CARD 1/1/10001/CARD PAYMENT COMMENTS/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'Y' for this command.
FIELD 2	Payment type code	STRING	1 - 2 digits	The payment type code
FIELD 3	Payment type description	INTEGER	1 to 15 chars	The payment type description
FIELD 4	Payment Type	INTEGER	1 digit	Payment Type: 0 = Cash 1 = Card 2 = Pending 3 = QR Payment 4 = Cheque 5 = Credit
FIELD 5	Payment flags	FLAGS	Fixed, 5 digits	The flag settings for PAYMENT TYPE are mapped as follows (left to right) (1:YES and 0:NO): 1st = Active Payment 2nd = Can be used in pay-out (cash out) 3rd = Can be used in receive on account (cash in) 4th = Payment can give change 5th = Print payment comments
FIELD 6	Payment comments	STRING	1 to 30 chars	Payment Comment line

REPLY PACKET	**TOTAL FIELD COUNT**	**DATA FIELD COUNT**	**EXAMPLE REPLY**
	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.27. Read Payment info [y]

This command will return all information about a programmed payment.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**y**	2 (Counting request code)	1 (Without request code)	"y/2/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'y' for this command.
FIELD 2	Payment code	INTEGER	1-2 digits	The payment code

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	11 (Counting reply code, device status & fiscal status)	8 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "CARDS /1/10001/CARD PAYMENT COMMENTS/0.00/0.00/0.00/0.00/

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Payment description	STRING	Fixed, 15 chars	The payment's programmed type description
FIELD 2	Payment type code	INTEGER	1 - 2 digits	The payment type code
FIELD 3	Payment type flags	FLAGS	Fixed, 5 digits	The flag settings for PAYMENT are mapped as follows (left to right) (1:YES and 0:NO): 1st = Active Payment 2nd = Can be used in pay-out (cash out) 3rd = Can be used in receive on account (cash in) 4th = Payment can give change 5th = Print payment comments

	FIELD 4	Payment comments	STRING	0 - 30 chars	A Comment line for this payment
	FIELD 5	Payment daily sum	AMOUNT	Default	It is the daily sum of this payment
	FIELD 6	Payment cash ins	AMOUNT	Default	It is the sum of cash ins for the specific payment type
	FIELD 7	Payment cash outs	AMOUNT	Default	It is the sum of cash outs for the specific payment type
	FIELD 8	Payment total sum	AMOUNT	Default	It is the total sum of this payment

8.2.28. Program USERNAME/PASSWORD(HTTP-POST) [\/16/]

This command sets the username & password for RESTful API posts.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	\	19 (Counting request code)	18 (Without request code)	"\/16////USerName/Pass/////////////"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request Code	STRING	Fixed, 1 character	Must be '\' for this command.
FIELD 2	Parameter Type	INTEGER	2 digits	Always '16'
FIELD 3-5	Reserved	INTEGER	Default	Reserved
FIELD 6	Username	STRING	1-20	Username
FIELD 7	Password	STRING	1-20	Password
FIELD 8-19	Reserved	INTEGER	Default	Reserved

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.29. Read USERNAME/PASSWORD(HTTP-POST) [,/16/]

This command reads the username & password for RESTful API posts.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	,	2 (Counting request code)	1 (Without request code)	",/16"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request Code	STRING	Fixed, 1 character	Must be ',' for this command.
	FIELD 2	Parameter Type	INTEGER	2 digits	Always '16'

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	5 (Counting reply code, device status & fiscal status)	2 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "UserName/Pass/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Username	STRING	1-20	Username
	FIELD 2	Password	STRING	1-20	Password

8.2.30. Read Device status [?]

This command is used to retrieve the status of the Device. Because this status information is always sent in the reply packet, the status command doesn't need any additional information to receive or return.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**?**	1 (Counting request code)	0 (Without request code)	"?/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be '?' for this command.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.31. Program VAT rates [b]

This command is used to program the VAT rates of the Device. For this command to succeed, a day must not be open. Important notice: max 50 changes available.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**b**	7 (Counting request code)	6 (Without request code)	"b/6.50/13.00/23.00/36.00///"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'b' for this command.
FIELD 2	Vat A rate	AMOUNT	0-5 digits, range 0-100	The VAT A rate to program.
FIELD 3	Vat B rate	AMOUNT	0-5 digits, range 0-100	The VAT B rate to program.
FIELD 4	Vat C rate	AMOUNT	0-5 digits, range 0-100	The VAT C rate to program.
FIELD 5	Vat D rate	AMOUNT	0-5 digits, range 0-100	The VAT D rate to program.
FIELD 6	Vat E rate	AMOUNT	0-5 digits	The VAT E rate to program.

				, range 0-100	
	FIELD 7	Exemption reason	INTEGER	0-2 digits	Exemption reason (0-31)

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.32. Program Real Time Clock [e]

This command is used to program the ECR/POS real time clock (i.e.: time and date). For this command to succeed, the 'clock' jumper must be short, otherwise the command will fail. Also, the date must not be prior to the last fiscal record's date.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**e**	3 (Counting request code)	2 (Without request code)	"e/110324/161800/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'e' for this command.
FIELD 2	System date	DATE6	Default	The date to set in RTC (Real time clock) "ddMMyy"
FIELD 3	System time	TIME	Default	The time to set in RTC "hhmmss"

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.33. Read Real Time Clock [t]

This command will is used to read the ECR's/POS's real time clock.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**t**	1 (Counting request code)	0 (Without request code)	"t/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 't' for this command.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	5 (Counting reply code, device status & fiscal status)	2 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "250224/160913/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	System date	DATE6	Fixed, 6 digits	The current date in device "ddMMyy"
	FIELD 2	System time	TIME	Fixed, 6 digits	The current time in device "hhmmss"

8.2.34. Issue report [x]

This command is used to issue the standard reports supported by the device.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**x**	2 (Counting request code)	1 (Without request code)	"x/1/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'x' for this command.
FIELD 2	Report type	INTEGER	2 digits, range 1-11, 17	The report type can be: 1 = X sales total report 2 = Z closure report 3 = Copy of last Z 4 = Daily Cash Check

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.35. Read Z report record [R]

This command is used to read the Device's Z report record.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**R**	2 (Counting request code)	1 (Without request code)	"R/1"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'R' for this command.
FIELD 2	Z number	INTEGER	1-4 digits	Z number

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	11 (Counting reply code, device status & fiscal status)	8 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "1408/0/210524/174800/150142252B6434D1A96863881756B8385BA905CE/32DB42805DD67B1745BC4A6B82100B1932FB47B8/DMU77000002/77/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Accumulated Signatures counter	INTEGER	Default	Accumulated signatures counter
FIELD 2	Daily Signatures counter	INTEGER	Default	Daily signatures counter
FIELD 3	Date issued	DATE6	Fixed, 6 digits	Z report date issued
FIELD 4	Time issued	TIME	Fixed, 6 digits	Z report time issued
FIELD 5	Signature _c	STRING	Fixed 40	Daily Signature _c
FIELD 6	Signature _d	STRING	Fixed 40	Daily Signature _d
FIELD 7	Serial Number	STRING	Fixed 11	Device serial number
FIELD 8	Z number	INTEGER	Default	Number of requested Z

8.2.36. Read VAT Rates [V]

This command is used to read the VAT rates of the Device.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**V**	1 (Counting request code)	0 (Without request code)	"V/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'V' for this command.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	9 (Counting reply code, device status & fiscal status)	6 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "6.00/13.00/24.00/36.00/0.00/27/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Vat A rate	AMOUNT	0-5 digits, range 0-100	The VAT A rate to read.
	FIELD 2	Vat B rate	AMOUNT	0-5 digits, range 0-100	The VAT B rate to read.
	FIELD 3	Vat C rate	AMOUNT	0-5 digits, range 0-100	The VAT C rate to read.
	FIELD 4	Vat D rate	AMOUNT	0-5 digits, range 0-100	The VAT D rate to read.

	FIELD 5	Vat E rate	AMOUNT	0-5 digits, range 0-100	The VAT E rate to read.
	FIELD 6	Exemption reason	INTEGER	1-2 digits	Exemption reason

8.2.37. Read Device counters [Z]

This command is used for getting the current counters/totals of the device .

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**Z**	1 (Counting request code)	0 (Without request code)	"Z/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'Z' for this command.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	8 (Counting reply code, device status & fiscal status)	5 (Without reply code, device status & fiscal status)	(reply code) (device status) (fiscal status) "5/3/0/0/10000/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	LAST Z NUMBER	AMOUNT	Default	LAST Z NUMBER
	FIELD 2	TOT.SIGNS	AMOUNT	Default	TOT.SIGNS
	FIELD 3	DAILY SIGNS	AMOUNT	Default	DAILY SIGNS
	FIELD 4	DATA COUNTER	AMOUNT	Default	Receipt's sums belonging to VAT D category
	FIELD 5	REMAINING INVOICES	AMOUNT	Default	REMAINING INVOICES

8.2.38. Device write [7]

This command is used to print user messages to display or to the printer.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**7**	3 (Counting request code)	2 (Without request code)	"7/1/text/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be '7' for this command.
FIELD 2	Display type	INTEGER	Default	Types can be: 0 = Clear 1 = Print to 1st line only 2 = Print to 2nd line only
FIELD 3	Display line	STRING	0-32 chars	The line may be up to 16 characters long

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.39. Begin Signature Block [{]

This command is used start the parse and signing document procedure.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**{**	19 (Counting request code)	18 (Without request code)	"{/////////////////0/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be '{' for this command.
FIELD 3-17	Reserved	INTEGER	Default	Reserved
FIELD 18	Invoice Type	INTEGER	Default	0 = Json Data Only 1 = Combined Raw Data & Json (the 2 sections are separated with the "#$#$#$#$#$/" as a single command) 2 = Raw Data Only

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.40. Sign Data Block [@]

This command is used to send the data block for signing. The max packet in bytes is 1000.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	@	2 (Counting request code)	1 (Without request code)	"@/Data"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be '@' for this command.
FIELD 2	Data block	STRING	1-1000	Data block

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.41. End Of Signature Block [}]

This command is used to end the signing procedure and reply the signature results.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**}**	1 (Counting request code)	0 (Without request code)	"}/"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be '}' for this command.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.42. EFTPOS Transactions(Invoice,TaxFree,Prepayment, Tokens [6]

This command will send the requested amount to the EFTPOS

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**6**	11 (Counting request code)	10 (Without request code)	"6/1/ABC12345/1.00/INVOICE/1/1/////" INVOICE "6/2//2.00/TAX FREE/2/1/////" TAXFREE "6/3//3.00/PREPAYMENT/1/1/////" PREPAYMENT - **ecrToken Examples** - **INVOICE DOC. ecrToken CLOSE (162=Invoice Services):** 6/4//100.00///1/AA999001/3a01dfb7-28d0-44ca-8d35-f9dc07f68108/162/UE9TMDExMFIvUzg2QjRDMi9SRE1LOTkwMDAwMDEvVFRLMDAwMi9NMC9DMDAvRFZpc2EgQ3JlZGl0OjAwOjQ5MTc5MSoqKioqKjM0ODk6MTAwMDA6MTAwMDA6MDowOjA6MTE6OTkwMDAwNDE6MjE2OjIyMjIyMjEwMDQwMDoxNDQ2OjEyMzQ2MzoyMDI0MDQxNzEyMzAzMDow/ **EXAMPLE INVOICE DOC. ecrToken CANCEL (161=Invoice Products):** 6/5//100.00///1/AA999002/3a01dfb7-28d0-44ca-8d35-f9dc07f68905/161// **EXAMPLE INVOICE DOC. ecrToken CANCEL Pending (162=Invoice Services):** 6/5//1.00///5/RDMU7702/3a01dfb7-28d0-44ca-8d35-f9dc07001023/162// **EXAMPLE RECEIPT DOC. ecrToken CANCEL (172=Receipt Products):** 6/5//1.00///1/RDMU7702/3a01dfb3-28d0-44ca-8d35-f9dc07002002/172// **EXAMPLE ecrToken CANCEL All open Tokens:** 6/6//////////

DESCRIPTION	TYPE	LENGTH	NOTES

	FIELD 1	Request code	STRING	Fixed, 1 character	Must be '6' for this command.
	FIELD 2	Transaction Type	INTEGER	Fixed,1 character	1. Invoice 2. TaxFree 3. Prepayment 4. Close Invoice Token 5. Cancel Invoice Token 6. Cancel All Open Tokens
	FIELD 3	Invoice Number	STRING	Fixed, character	1: The invoice number. The letter **must** use the English alphabet
	FIELD 4	Amount to send to EFTPOS	Amount	Default	
	FIELD 5	Description of Payment	STRING		
	FIELD 6	EFTPOS Index	INTEGER	Fixed,1 character	Index of predefined EFTPOS 1-12 (Empty in Type 6)
	FIELD 7	EFTPOS Transaction Type	INTEGER	Fixed,1 character	1=Sale 2=Sale with installments 3=Pre-approval registration 4=Mail order 5=Pending (REGRECEIPT)
	FIELD 8	EFTPOS TID	STRING		EFTPOS TID (Empty in Type 6)
	FIELD 9	UID	STRING		UID of document corellated with ecrToken to be close/cancel (In Types 4 or 5)
	FIELD 10	DOC.CODE	STRING		Document Code corellated with ecrToken to be close/cancel (In Types 4 or 5)

	FIELD 11	BASE64URL EFTPOS REPLY	STRING		EFTPOS reply data BASE64 url encoded (In Types 4 or 5)

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4-7 (Counting reply code & status)	1-4 (Without reply code, status)	(reply code)(status) "1|QS9TMDY1NTQyL0YxMDA6OTc4OjIvRDIwMjQwMzI5MTkyMDMwL1JETVU3NzAwMDAwMS9IMS9UVFBEQS0wMS9NMC9RNkNDNkQ5MjE|00|01|1.00|Mastercard|123412******1234|DMUEFT01|1|1|787032|133030119089|20240329192037|0|001|0|0.00|/"
	Reply like 8.2.22.Read EFTPOS last Invoice info [5/99/]		

8.2.43. Request For ecrToken [_](POL 1155)

This command is used in order to get a unique ecrToken for EFTPos payments or get info about pending ecrTokens.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**—**	9 (Counting request code)	8 (Without request code)	"_/0/3a01dfb7-28d0-44ca-8d35-f9dc07001001/162/1.00/001/TK001/DMK9901A/1/" "_/1///////" "_/2///////" "_/3/3a01dfb7-28d0-44ca-8d35-f9dc07f68100//////" "_/4///////" "_/5/////// "

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be '_' for this command.
FIELD 2	Request type	INTEGER	Fixed, 1 char	**0**= Get ecrToken - - - - - - - - - - - - - - **1**= Return counters of registered tokens. - - - - - - - - - - - - - - **2**= Return Pending Tokens one by one. - - - - - - - - - - - - - - **3**= Search by UID and Return Pending Token - - - - - - - - - - - - - - **4**= Start Search and Return List of Pending Tokens

				- - - - - - - - - - - - - **5**= Return next packet of List of Pending Tokens
FIELD 3	UID	STRING	1-60 chars	Document (Receipt/Invoice) UID
FIELD 4	DOC.CODE	STRING		Document (Invoice/Receipt) Code
FIELD 5	AMOUNT	Amount	Default	Amount of payment
FIELD 6	Operator Number	STRING	1-8 Chars	Operator Number
FIELD 7	Document Number	STRING	1-8 Chars	Document Number
FIELD 8	TID	STRING	Default	Terminal ID of EFTPOS
FIELD 9	EFTPOS Transaction Type	INTEGER	Fixed,1 character	1=Sale 2=Sale with installments 3=Pre-approval registration 4=Mail order 5=Pending (REGRECEIPT)

Reply in case of Get ecrToken (_/0)

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	6 (Counting reply code & status)	3 (Without reply code, status)	(reply code)(status) "0/RDMU7702/AE9FQ1IwMTEwQS9TMThBRkM0L0YxMDA6OTc4OjIvRDIwMjQwNTE1MTA1ODA4L1JETVU3NzAwMDAwMi9IMDAxL1RBTFAvTTAvUUUzNUE3MUEy/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Status	INTEGER	Fixed, 1 char	0= Successfully created new ecrToken
FIELD 2	EFTPOS TID	STRING	1-8 chars	TID of EFTPOS correlated with this Token
FIELD 3	base64 url encoded of ecrToken	STRING	Default	AE9FQ1IwMTEwQS9TMThBRkM0L0YxMDA6OTc4OjIvRDIwMjQwNTE1MTA1OD

					A4L1JETVU3NzAwMDAwMi9IMDAxL1R BTFAvTTAvUUUzNUE3MUEy

Reply in case of Return counters of registered tokens (_/1)

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	5 (Counting reply code & status)	2 (Without reply code, status)	(reply code)(status) "5/0/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Daily Total Tokens	INTEGER	Default	Number of Daily Total Tokens
FIELD 2	Pending Daily Tokens	INTEGER	Default	Number of Pending Daily Tokens

Reply in case of Return Pending Tokens one by one (_/2)

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4-13 (Counting reply code & status)	1-10 (Without reply code, status)	(reply code)(status) "0/3a01dfb3-28d0-44ca-8d35-f9dc07002003/1.00/162/A001/1/1/140524/174942/RDMU7702/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Status	INTEGER	Fixed, 1 digit	0= Token found 1= No more Tokens 2= Error
FIELD 2	UID	STRING	Default	UID of ectToken
FIELD 3	Amount	Amount	Default	Amount
FIELD 4	Document Code	INTEGER	Default	Document Code
FIELD 5	Document Number	STRING		Document Number
FIELD 6	EFTPOS Index	INTEGER	1-2 digits	Index of EFTPOS correlated with this ecrToken
FIELD 7	Transaction Type	INTEGER	1 digit	Transaction Type: 1=Sale 2=Sale with installments

					3=Pre-approval registration 4=Mail order 5=Pending (REGRECEIPT)
	FIELD 8	Date	DATE6	Fixed, 6 digits	ecrToken Issue Date
	FIELD 9	Time	TIME	Fixed, 6 digits	ecrToken Issue Time
	FIELD 10	EFTPOS TID	STRING	1-8 chars	TID of EFTPOS

Reply in case of Search by UID and Return Pending Token (_/3)

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4-13 (Counting reply code & status)	1-10 (Without reply code, status)	(reply code)(status) "0/3a01dfb3-28d0-44ca-8d35-f9dc07002003/1.00/162/A001/1/1/140524/174942/RDMU7702/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Status	INTEGER	Fixed, 1 digit	0= Token found 1= Token Not found/Error
FIELD 2	UID	STRING	Default	UID of ectToken
FIELD 3	Amount	Amount	Default	Amount
FIELD 4	Document Code	INTEGER	Default	Document Code
FIELD 5	Document Number	STRING		Document Number
FIELD 6	EFTPOS Index	INTEGER	1-2 digits	Index of EFTPOS correlated with this ecrToken
FIELD 7	Transaction Type	INTEGER	1 digit	Transaction Type: 1=Sale 2=Sale with installments 3=Pre-approval registration 4=Mail order 5=Pending (REGRECEIPT)
FIELD 8	Date	DATE6	Fixed, 6 digits	ecrToken Issue Date

FIELD 9	Time	TIME	Fixed, 6 digits	ecrToken Issue Time
FIELD 10	EFTPOS TID	STRING	1-8 chars	TID of EFTPOS

Reply in case of Return List of Pending Tokens (_/4 and _/5)

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4-n (Counting reply code & status) n=1+number of ecrTokens records Character '|' used as separator for each record subfields	1-n (Without reply code, status)	(reply code)(status) "0/3a01dfb3-28d0-44ca-8d35-f9dc07002001|1.00|172|ALP|1|1|170524|142245|RDMU7702/3a01dfb3-28d0-44ca-8d35-f9dc07002002|2.00|172|ALP|1|1|170524|142250|RDMU7702/"

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Status	INTEGER	Fixed, 1 digit	0= Only one OR last packet of Pending Tokens 1= No more Tokens 2= Error 3= Exists more Tokens (call same command with Request type=5)
FIELD 2	UID	STRING	Default	UID of ectToken
FIELD 3	Amount	Amount	Default	Amount
FIELD 4	Document Code	INTEGER	Default	Document Code
FIELD 5	Document Number	STRING		Document Number
FIELD 6	EFTPOS Index	INTEGER	1-2 digits	Index of EFTPOS correlated with this ecrToken

	FIELD 7	Transaction Type	INTEGER	1 digit	Transaction Type: 1=Sale 2=Sale with installments 3=Pre-approval registration 4=Mail order 5=Pending (REGRECEIPT)
	FIELD 8	Date	DATE6	Fixed, 6 digits	ecrToken Issue Date
	FIELD 9	Time	TIME	Fixed, 6 digits	ecrToken Issue Time
	FIELD 10	EFTPOS TID	STRING	1-8 chars	TID of EFTPOS

8.2.44. Send Keyboard Key requests [)]

This command is used to send keyboard keys to device.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**)**	2 (Counting request code)	1 (Without request code)	Error (A1-Firmwware Upgrade confirmation): **")/131"** Error (A1-Firmwware Upgrade cancellation): **")/130"**

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be '@' for this command.
	FIELD 2	KEY CODE	INTEGER	1-3	Keyboard Key Codes: **KEY:** 15 **LEFT:** 132 **RIGHT:** 133 **UP:** 129 **CANCEL:**130 **OK:** 131

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.2.45. Send RESEND-ALL to EFTPOS [\/17/]

This command send command RESEND-ALL to selected EFTPOS.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	\	19 (Counting request code)	18 (Without request code)	"\/17/1/////////////////”

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request Code	STRING	Fixed, 1 character	Must be '\' for this command.
FIELD 2	Parameter Type	INTEGER	2 digits	Always ‘17’
FIELD 3	EFTPOS Index	INTEGER	1-2 digits	Index of EFTPOS (1-12) to send command RESEND-ALL.
FIELD 4-19	Reserved	INTEGER	Default	Reserved

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	3 (Counting reply code, device status & fiscal status)	0 (Without reply code, device status & fiscal status)	This command's reply packet does not contain additional information; only 1 field of reply code, 1 field of device status and 1 field of fiscal status.

8.3. RESTful API Reference

8.3.1. Schema

All API access is over HTTPS/HTTPS, and accessed from the local IP address of the device. Typical endpoint is as following:

http://192.1698.1.20/api/.

All data is sent and received as JSON.

All timestamps are returned in ISO 8601 format: YYYY-MM-DDTHH:MM:SSZ except those are described with different format in command protocol

Summary Representations - When you fetch a list of resources, the response includes a subset of the attributes for that resource. This is the "summary" representation of the resource.

Detailed Representations - When you fetch an individual resource, the response typically includes all attributes for that resource. This is the "detailed" representation of the resource.

8.3.2. HTTP Requests

API requests must be written as HTTP requests, and include the following components:

- **HTTP Method**: Describes the type of HTTP action (POST, GET, PUT or DELETE)
- **URL**: Describes the resource you are creating or accessing, along with any optional arguments.
- **HTTP Headers**: Specifies attributes of the request, including authentication, encoding and request format
- **Request Body**: Describes resources or specifies a call-control script.

8.3.3. Authorization

The API interface supports **Basic Authentication** in order to access the rest of commands. Requests that require authentication will return 401- Unauthorized, to protect private user information from unauthorized users.

The default username & password for the first login is:

UserName: **FHMAS** Password: **FHMAS** (to change the credentials send the command [\/16])

8.3.4. API reference

All post requests must be followed with the GetResult Post command to receive the full protocol command response.

Each post request except the GetResult responds to the successful reception of the data and with a result message.

It is mandatory to call the GetResult after each method in order to proceed with the next command otherwise the device will respond with corresponding message.

8.3.4.1.PostInvoice_00

Typical post for invoice with all the required items included in the json body.

Method	Purpose
Response formats	200-Success, 401-Unauthorized
Requires authentication?	Basic Authentication
Body Format	JSON

Request

The HTTP method and URL is as follows:

Method	URL
POST	http://Device_IP/api/PostInvoice_00

Parameters

This command doesn't include any optional parameter except the Json Body as its described in **8.4.**

Response

Name	Type	Description
result	long	Status code
description	string	Description of the process status

Sample Request


```
{
  "Doc": {
    "ReceiverHeader": {
      "CompanyTitle": "MAT S.A.",
      "Profession": "Profession",
      "Address": "Address",
      "TaxID": "997981320",
      "TaxOffice": "Tax Office",
      "PhoneNumber": "210565042",
      "PrintLine": ""
    },
    "InvoiceDetails": {
      "InvoiceUID": "3a01dfb7-28d0-44ca-8d35-f9dc07f68e03",
      "InvoiceType": 172,
      "PrintDevice": 1,
      "ReqForToken": 0,
      "CancelDevDailyNum": null,
      "InvoiceNo": "100",
      "InvoiceSeries": null,
      "InvoiceTotal": 10.00,
      "Operator": "user1",
      "Machine": "Shop",
      "PrintLine": ""
    },
    "TransactionLines": [
      {
        "LineNo": 1,
        "ClassCategory": 2,
        "ClassType": null,
        "FuelCode": null,
        "Code": "123456789",
        "Description": "Product #1",
        "ItemAmount": 10.00,
        "MeasurementUnit": 1,
        "SaleQty": 1,
        "NetAmount": 8.06,
        "VatAmount": 1.94,
        "GrossAmount": 10.00,
        "DMType": 0,
        "DMValue": 0,
        "DMPercentage": 0,
        "VatRate": "24.00",
        "VatExCategory": 0,
        "FeeCategory":0,
        "PrintLine": ""
      }
    ],
    "Payments": [
      {
```

```
        "Type": 0,
        "EftposTID": null,
        "EftposTransType": 0,
        "Value": 10.00,
        "PrintLine": ""
      }
    ],
    "InvoiceFooter": {
      "Line1": "LINE1",
      "Line2": "LINE2",
      "Line3": "LINE3",
      "Line4": "LINE4",
      "Line5": "LINE5",
      "Line6": "LINE6",
      "PrintLine": ""
    }
  }
}
```

Sample Response

```
{
    "response":{
        "result": 0,
        "description": "Sign Process Started"
    }
}
```

8.3.4.2. PostInvoice_01

This method of posting Invoice supports big data print output and requires the raw data before the json data. These 2 sections are separated with a single line and with the following string separator:

"#$#$#$#$#$"

Method	Purpose
Response formats	200-Success, 401-Unauthorized
Requires authentication?	Basic Authentication
Body Format	Raw,JSON

Request

The HTTP method and URL is as follows:

Method	URL
POST	http://Device_IP/api/PostInvoice_01

Parameters

This command doesn't include any optional parameter except the Json Body as its described in **8.4.**

Response

Name	Type	Description
result	long	Status code
description	string	Description of the process status

Sample Request

```
    MAT S.A.
---------------
Product #1 10.00
---------------
TOTAL      10.00
---------------
    FOOTER
#$#$#$#$#$
{
  "Doc": {
    "ReceiverHeader": {
      "CompanyTitle": "MAT S.A.",
      "Profession": "Profession",
      "Address": "Address",
      "TaxID": "997981320",
      "TaxOffice": "Tax Office",
      "PhoneNumber": "2105695042",
      "PrintLine": ""
    },
    "InvoiceDetails": {
      "InvoiceUID": "3a01dfb7-28d0-44ca-8d35-f9dc07f68e07",
      "InvoiceType": 172,
      "PrintDevice": 0,
      "ReqForToken": 0,
      "CancelDevDailyNum": null,
      "InvoiceNo": "100",
      "InvoiceSeries": null,
      "InvoiceTotal": 10.00,
      "Operator": "user1",
      "Machine": "Shop",
      "PrintLine": ""
    },
    "TransactionLines": [
      {
        "LineNo": 1,
        "ClassCategory": 2,
        "ClassType": null,
        "FuelCode": null,
        "Code": "123456789",
        "Description": "Product #1",
        "ItemAmount": 10.00,
        "MeasurementUnit": 1,
        "SaleQty": 1,
        "NetAmount": 8.06,
        "VatAmount": 1.94,
        "GrossAmount": 10.00,
        "DMType": 0,
        "DMValue": 0,
        "DMPercentage": 0,
        "VatRate": "24.00",
```

```
        "VatExCategory": 0,
        "FeeCategory":0,
        "PrintLine": ""
      }
    ],
    "Payments": [
      {
        "Type": 0,
        "EftposTID": null,
        "EftposTransType": 0,
        "Value": 10.00,
        "PrintLine": ""
      }
    ],
    "InvoiceFooter": {
      "Line1": "LINE1",
      "Line2": "LINE2",
      "Line3": "LINE3",
      "Line4": "LINE4",
      "Line5": "LINE5",
      "Line6": "LINE6",
      "PrintLine": ""
    }
  }
}
```

Sample Response

```
{
    "response":{
        "result": 0,
        "description": "Sign Process Started"
    }
}
```

8.3.4.3. PostInvoice_02

This method of posting Invoice supports only raw data in body message in order to support and sign any required report:

Method	Purpose
Response formats	200-Success, 401-Unauthorized
Requires authentication?	Basic Authentication
Body Format	Raw Data

Request

The HTTP method and URL is as follows:

Method	URL
POST	http://Device IP/api/PostInvoice 02

Parameters

This command doesn't include any optional parameter except the Json Body as its described in **8.4.**

Response

Name	Type	Description
result	long	Status code
description	string	Description of the process status

Sample Request

```
                    Περίοδος Ισοζυγίου
Ημερ/νία-Ώρα εκτ.

ΠΡΑΤΗΡΙΟ ΥΓΡΩΝ ΚΑΥΣΙΜΩΝ
10/01/2024 00:02:00
11/01/2024 07:07
30ΧΛΜ Λ.
10/01/2024 23:55:00
```

ΑΤΤΙΚΗΣ
ΤΗΛ.: ΑΜΔΙΚΑ:
ΑΦΜ: ΔΟΥ:
(1)Εκροές Αντλιών

Είδος Καυσίμου	Α/Α Αντλίας		Α/Α Ακρ/σιου				
Μετρητής Έναρξης Μετρητής Λήξης	Μετρητής (Lt)	Μετρητής (Lt15)	Α/Α Δεξ/νής				
Πωλήσεις (Lt) Πωλήσεις(Lt15)	Λιτρομ/σεις (Lt)	Λιτρομ/σεις(Lt15)	Διαφορά (Lt)	Διαφορά (Lt15)			
Benzine 95RON	1	94490263-P-39-04264	1				
94490263-N-39-18447 1.185.774,08	1.186.503,26	729,18	727,80	1			
729,18 727,78	0,00	0,00	0,00	0,02			
Benzine 95RON	2	94490263-P-39-04264	1				
94490263-N-39-18448 1.791.918,83	1.792.681,72	762,89	761,36	1			
762,89 761,37	0,00	0,00	0,00	-0,01			
Benzine 100RON	3	94490263-P-39-04265	1				
94490263-N-39-18449 231.192,07	231.346,06	153,99	153,72	7			
153,99 153,72	0,00	0,00	0,00	0,00			
Benzine 95RON+	4	94490263-P-39-04265	1				
94490263-N-39-18450 562.653,68	562.814,05	160,37	159,66	2			
160,37 159,66	0,00	0,00	0,00	0,00			
Diesel	5	94490263-P-39-04266	1				
94490263-N-39-18451 1.136.378,89	1.137.257,80	878,91	869,54	3			
878,91 869,56	0,00	0,00	0,00	-0,02			
Diesel	6	94490263-P-39-04266	1				
94490263-N-39-18452 207.366,02	207.590,58	224,56	222,16	3			
224,56 222,17	0,00	0,00	0,00	-0,01			
Diesel Heating	7	94490263-P-39-04267	1				
94490263-N-39-50855 230.389,55	230.505,32	115,77	115,70	6			
115,77 115,69	0,00	0,00	0,00	0,01			
Diesel premium	7	94490263-P-39-04267	2				
94490263-N-39-50856 485.910,49	486.380,88	470,39	469,03	4			
470,39 469,03	0,00	0,00	0,00	0,00			
Diesel Heating	8	94490263-P-39-04267	1				
94490263-N-39-50857 29.506,47	29.719,07	212,60	212,43	6			
212,60 212,44	0,00	0,00	0,00	-0,01			
Diesel premium	8	94490263-P-39-04267	2				
94490263-N-39-50858 11.590,79	11.599,69	8,90	8,87	4			
8,90 8,87	0,00	0,00	0,00	0,00			
Diesel Heating	9	94490263-P-39-09291	1				
94490263-N-39-24598 2.444.575,70	2.450.952,00	6.376,30	6.370,08	6			
6.376,30 6.370,08	0,00	0,00	0,00	0,00			

Σύνολα ανά καύσιμο		TAXIS ID			
Μετρητής (Lt)	Μετρητής (Lt15)		Πωλήσεις (Lt)	Πωλήσεις(Lit15)	
Λιτρομ/σεις (Lt)	Λιτρομ/σεις(Lt15)	Διαφορά (Lt)	Διαφορά (Lt15)		
Benzine 95RON		10			
1.492,07	1.489,16		1.492,07	1.489,15	0,00
0,00	0,00	0,01			
Benzine 95RON+		11			
160,37	159,66		160,37	159,66	0,00
0,00	0,00	0,00			

Benzine 100RON
153,99 153,72
0,00 0,00 0,00
Diesel
1.103,47 1.091,70
0,00 0,00 -0,03
Diesel premium
479,29 477,90
0,00 0,00 0,00
Diesel Heating
6.704,67 6.698,21
0,00 0,00 0,00

(2)Μετρ. Δεξαμενών

Είδος Καυσίμου
Έναρξη (Lt) Έναρξη (Lt15)
Λήξη (Lt) Λήξη (Lt15) Λήξη Θερμ/σία Χωρητ/τα
Διαφορά Όγκου(Lt) Διαφορά Όγκου(Lt15)
Benzine 95RON
4.059,45 4.047,31
5.554,15 5.543,06 16,80 9.997,74
-1.494,70 -1.495,75
Benzine 95RON+
1.466,59 1.459,90
1.306,53 1.300,72 18,60 6.099,31
160,06 159,18
Diesel
5.029,02 4.973,64
3.924,34 3.882,13 27,90 6.958,55
1.104,68 1.091,51
Diesel premium
1.949,00 1.943,01
1.467,27 1.463,14 18,30 3.446,27
481,73 479,87
Diesel Heating
138,73 138,52
4.780,68 4.784,61 14,80 7.177,30
-4.641,95 -4.646,09
Diesel Heating
3.610,42 3.604,47
4.245,13 4.242,33 16,10 7.173,76
-634,71 -637,86
Benzine 100RON
1.369,31 1.366,71
1.213,72 1.211,70 16,30 3.602,10
155,59 155,01

Μετρ. Δεξαμενών ανά καύσιμο

Διαφορά Όγκου(Lt) Διαφορά Όγκου(Lt15)
Benzine 95RON
-1.494,70 -1.495,75
Benzine 95RON+
160,06 159,18
Benzine 100RON
155,59 155,01
Diesel
1.104,68 1.091,51
Diesel premium
481,73 479,87

Α/Α Δεξ/νής Α/Α Μητρώου Δεξ/νής
Έναρξη Θερμ/σία Λήξη (Lt) Λήξη Διαφορά Όγκου(Lt) Διαφορά
1 94490263-T-39-05580
2 94490263-T-39-05581
3 94490263-T-39-05582
4 94490263-T-39-05583
5 94490263-T-39-05584
6 94490263-T-39-05585
7 94490263-T-39-05586

TAXIS ID

10
11
12
20
21

12
153,99 153,72 0,00
20
1.103,47 1.091,73 0,00
21
479,29 477,90 0,00
30
6.704,67 6.698,21 0,00

Diesel Heating 30
-5.276,66 -5.283,95
(3)Μετρ. Μεταβ. Δεξαμενών
Είδος Καυσίμου A/A Δεξ/νής A/A
Μητρώου Δεξ/νής Lt Lt15 Χαρακτηρισμός
Benzine 95RON 1
94490263-T-39-05580 2.986,69 2.992,09 ΠΑΡΑΛΑΒΗ
Diesel Heating 5
94490263-T-39-05584 4.631,09 4.634,05 ΠΑΡΑΛΑΒΗ
Diesel Heating 6
94490263-T-39-05585 7.339,56 7.340,07 ΠΑΡΑΛΑΒΗ
(4)Παραστ. Παραλαβής
Είδος Καυσίμου TAXIS ID
Lt Lt15
Benzine 95RON 10
3.002,00 3.003,88
Diesel Heating 30
12.001,00 11.996,48
(5)Ημερήσια κίνηση δεξαμενών
Είδος Καυσίμου A/A Δεξ. A/A
Μητρώου Δεξ/νής Lt Lt15
Benzine 95RON 1
94490263-T-39-05580 1.491,99 1.496,34
Benzine 95RON+ 2
94490263-T-39-05581 160,06 159,18
Benzine 100RON 7
94490263-T-39-05586 155,59 155,01
Diesel 3
94490263-T-39-05582 1.104,68 1.091,51
Diesel premium 4
94490263-T-39-05583 481,73 479,87
Diesel Heating 5
94490263-T-39-05584 -10,86 -12,04
Diesel Heating 6
94490263-T-39-05585 6.704,85 6.702,21
Διαφορές ανά προιόν
(3-4). Διαφ. Μεταβολών
Είδος Καυσίμου Μεταβολές Δεξ. Lt
Δελτία Παραλ. Lt Διαφορά Lt Μεταβολές Δεξ. Lt15 Δελτία Παραλ. Lt15
Διαφορά Lt15
Benzine 95RON 2.986,69
3.002,00 -15,31 2.992,09 3.003,88
-11,79
Benzine 95RON+ 0,00
0,00 0,00 0,00 0,00 0,00
Benzine 100RON 0,00
0,00 0,00 0,00 0,00 0,00

```
Diesel                                          0,00
0,00            0,00              0,00                          0,00
0,00
Diesel premium                                  0,00
0,00            0,00              0,00                          0,00
0,00
Diesel Heating                                  11.970,65
12.001,00       -30,35            11.974,12                     11.996,48
-22,36
Διαφορές Εισροών-Εκροών
(5-1)
Είδος Καυσίμου                                 Μεταβολές Δεξ. Lt
Διανομές Lt     Διαφορά Lt        Απόκλιση %                    Μεταβολές Δεξ. Lt15
Διανομές Lt15                     Διαφορά Lt15     Απόκλιση %
Benzine 95RON                                   1.491,99
1.492,07        -0,08               -0,01                       1.496,34
1.489,15                          7,19             0,48
Benzine 95RON+                                  160,06
160,37          -0,31               -0,19                       159,18
159,66                            -0,48            -0,30
Benzine 100RON                                  155,59
153,99          1,60                1,03                        155,01
153,72                            1,29             0,83
Diesel                                          1.104,68
1.103,47        1,21                0,11                        1.091,51
1.091,73                          -0,22            -0,02
Diesel premium                                  481,73
479,29          2,44                0,51                        479,87
477,90                            1,97             0,41
Diesel Heating                                  6.693,99
6.704,67        -10,68              -0,16                       6.690,17
6.698,21                          -8,04            -0,12


~~~ESEND~~~
064949449;;;;;;;;501;;0;0.00;0.00;0.00;0.00;0.00;0.00;0.00;0.00;0.00;0.00;0;
```

Sample Response

```
{
    "response":{
        "result": 0,
        "description": "Sign Process Started"
    }
}
```

8.3.4.4. GetResult

This method returns the response of the previous method call and is necessary after each post command.

Method	Purpose
Response formats	200-Success, 401-Unauthorized
Requires authentication?	Basic Authentication
Body Format	JSON

Request

The HTTP method and URL is as follows:

Method	URL
POST	http://Device_IP/api/GetResult

Parameters

This command doesn't include any optional and required parameter.

Response

Name	Type	Description
result	long	Status code
description	string	Description of the process status

Sample Request

```
{
   "result": "?"
 }
```

Sample Responses

Cmd: GetStatus


```
{
 "response":{
  "result": 0,
  "description": "00/D0/00/"
 }
}
```


Post: PostInvoice_00


```
{
 "response":{
  "result": 0,
  "description":
"00/D1/06/00023/00001/20240303/1630/469465BD8DBBBB408B7031AEDD7CC266756B6689/997981320;998927533;
DMK99000001;;202403031630;1;23;8;172;;100;0.00;0.00;8.06;0.00;0.00;0.00;0.00;1.94;0.00;10.00;0;0;
0.00;;(24.00,8.06,2,0);7D983C87E2;C3094E43DF55CCE6EE5BA90719AF0C807BD34966
/DMK99000001/0008/"
 }
}
```

8.3.4.5. SendCmd

This post request is used to execute each of the procedures that are described at the MCP command protocol.

Method	Purpose
Response formats	200-Success, 401-Unauthorized
Requires authentication?	Basic Authentication
Body Format	JSON

Request

The HTTP method and URL is as follows:

Method	URL
POST	http://Device_IP/api/SendCmd

Parameters

This command doesn't include any optional parameter except the Json Body with the protocol command.

Body Message

CMD	Data
Data	Protocol Command

Response

Name	Type	Description
result	long	Status code
description	string	Description of the process status

Sample Request

```
{
  "cmd": {
    "data": "v/"
  }
}
```

Sample Response

```
{
    "response":{
        "result": 0,
        "description": "Command Received"
    }
}
```

8.3.4.6. Echo

This post request is used to echoing the device and check the connection.

Method	Purpose
Response formats	200-Success, 401-Unauthorized
Requires authentication?	Basic Authentication
Body Format	JSON

Request

The HTTP method and URL is as follows:

Method	URL
POST	http://Device_IP/api/Echo

Parameters

This command doesn't include any optional and required parameter.

Body Message

CMD	Data
Data	Command

Response

Name	Type	Description
result	long	Status code
description	string	Description of the process status

Sample Request

```
{
  "echo": 0
}
```

Sample Response

```
{
    "response":{
        "result": 0,
        "description": "Command Received"
    }
}
```

8.4. JSON Schema

8.4.1. Invoice

Json Parameters

ReceiverHeader

Name	Type	Description	Required
CompanyTitle	String	Επωνυμία Παραλήπτη	Optional
Profession	String	Επάγγελμα	No
Address	String	Διεύθυνση	No
TaxID	String	Α.Φ.Μ.	No
TaxOffice	String	Δ.Ο.Υ.	
PhoneNumber	String	Τηλέφωνο	Yes
PrintLine	String	Γραμμές εκτύπωσης	Yes

InvoiceDetails

Name	Type	Description	Required
InvoiceUID	String	Μοναδικό αναγνωριστικό του παραστατικού. Με αυτό το αναγνωριστικό θα μπορεί το εμπορικό να ζητήσει δεδομένα από παραστατικό που έχει στείλει προς έκδοση.	Yes
InvoiceType	Integer	Κωδικός Παραστατικού σύμφωνα με τον Πίνακα Δ της ΠΟΛ 1220/2012 και τους νέους κωδικούς της ΠΟΛ 1173/2022	Yes
PrintDevice	String	0 = ERP, 1 = Εσωτερικός Εκτυπωτής ΦΗΜΑΣ,	Yes

		2 = Εξωτερικός Εκτυπωτής Συνδεδεμένος στο ΦΗΜΑΣ	
ReqForToken	String	0 = έκδοση Παραστατικού 1 = Ολοκλήρωση Υπό έκδοση παραστατικού από λήψη ecrToken. 2 = Ακυρωτικό και ολοκλήρωση ecrToken με ταυτόχρονη έκδοση παραστατικού λιανικής με υποχρεωτική ενσωμάτωση γραμμών. 3 = έκδοση παραστατικού μόνο για ετεροχρονισμένη πληρωμή (προηγείται λήψη ecrToken ετεροχρονισμένης συναλλαγής)	Yes
CancelInvType	String	Κωδικός Παραστατικού που ακυρώνεται	Yes
CancelDevDailyNum	Integer	Στην περίπτωση Ειδικού Ακυρωτικού (6) θα πρέπει να περιέχει το ΑΑ ημερήσιου counter που πήρε το παραστατικό που θα ακυρωθεί. Σε αυτή την περίπτωση τα πεδία TransactionLines και Payments δεν λαμβάνονται υπόψη (μπορούν να είναι null)	Yes
CancelInvNo	Integer	Αριθμός Παραστατικού που ακυρώνεται	Yes
CancelInvSeries	String	Σειρά παραστατικού που ακυρώνεται	Yes
InvoiceNo	String	Αφορά τον αριθμό παραστατικού που στέλνει το εμπορικό	Yes
InvoiceSeries	String	σειρά παραστατικού από το εμπορικό. Else can be null	No
InvoiceTotal	Decimal	Σύνολο αξίας παραστατικού	
Operator	String	can be null or missing	NO

Machine	String	can be null or missing	NO
GasStationLicNum	Integer	Αριθμός άδειας LPG παραλήπτη/πρατηρίου (Μόνο για Διυλιστήρια και Εταιρείες εμπορίας πετρελαιοειδών προς πρατήρια καυσίμων)	NO
GasStationInstalNum	Integer	Αριθμός εγκατάστασης του λήπτη (Μόνο για Διυλιστήρια και Εταιρείες εμπορίας πετρελαιοειδών προς πρατήρια καυσίμων)	NO
InvWithholdingTaxTotal	Decimal	Ποσό παρακράτησης φόρου	NO
PrintLine	String		NO

TransactionLines

Name	Type	Description	Required
LineNo	String	Μοναδικό αναγνωριστικό του παραστατικού. Με αυτό το αναγνωριστικό θα μπορεί το εμπορικό να ζητήσει δεδομένα από παραστατικό που έχει στείλει προς έκδοση.	Yes
ClassCategory	Integer	1. ΕΜΠΟΡΕΥΜΑ 2. ΠΡΟΙΟΝ 3. ΥΠΗΡΕΣΙΑ 4. ΠΑΓΙΟ(???) 5. ΛΟΙΠΑ ΕΣΟΔΑ(???) 6. ΕΣΟΔΑ ΓΙΑ ΛΟΓΑΡΙΑΣΜΟ ΤΡΙΤΩΝ(???) 7. Έξοδα	Yes
ClassType	String	Υποχρεωτικό στην περίπτωση Τιμολογίου. Αποδεκτές τιμές από πίνακα MyData (πχ E3_561_003).	NO

FuelCode	String	Κωδ. Καυσίμου. Υποχρεωτικό μόνο στην περίπτωση παραστατικού καυσίμων. Else can be null or missing	Yes
Code	String	Κωδικός προϊόντος/εμπορεύματος. can be null or missing	Yes
Description	String	Περιγραφή προϊόντος/εμπορεύματος	Yes
ItemAmount	Decimal	Στην περίπτωση Ειδικού Ακυρωτικού (6) θα πρέπει να περιέχει το ΑΑ ημερήσιου counter που πήρε το παραστατικό που θα ακυρωθεί. Σε αυτή την περίπτωση τα πεδία TransactionLines και Payments δεν λαμβάνονται υπόψη (μπορούν να είναι null)	Yes
MeasurementUnit	Integer	1. Τεμάχια 2. Κιλά 3. Λίτρα	Yes
SaleQty	Float	Ποσότητα πώλησης στην περίπτωση Λίτρων θα περιέχει 3 δεκαδικά	Yes
NetAmount	Decimal	Καθαρή αξία	Yes
VatAmount	Decimal	Αξία Φ.Π.Α.	Yes
GrossAmount	Decimal	Μεικτή αξία	
DMType	String	Τύπος έκπτωσης/αύξησης. 1.ΕΚΠΤΩΣΗ ΠΟΣΟΥ 2.ΕΚΠΤΩΣΗ ΠΟΣΟΣΤΟΥ 3.ΑΥΞΗΣΗ ΠΟΣΟΥ 4.ΑΥΞΗΣΗ ΠΟΣΟΣΤΟΥ 5.ΑΛΛΑΓΗ/ΕΠΙΣΤΡΟΦΗ ΕΙΔΟΥΣ (Αφορά επιστροφή Είδους και τα ποσά της κίνησης πρεπει να έχουν θετικό πρόσημο)	NO
DMValue	Decimal	Αξία έκπτωσης/Αύξησης	NO

DMPercentage	Decimal	Ποσοστό έκπτωσης/αύξησης	NO
VatRate	String	Συντελεστής Φ.Π.Α.	NO
VatExCategory	Decimal	Στην περίπτωση ΦΠΑ 0% περιέχει τον κωδικό της κατηγορίας απαλλαγής Φ.Π.Α.	NO
FeeCategory	String		No
PrintLine			

Payments

Name	Type	Description	Required
Type	Integer	0. Μετρητά 1. Κάρτα 2. Προφορτωμένη 3. QRPayment 4. Επιταγή 5. Επί πιστώσει	Yes
Description	String	Περιγραφή Πληρωμής	Yes
EftposDescr	String	EFTPOS Description (Μόνο για πληρωμή μέσω EFTPOS)	Yes
EftposTID	String	EFTPOS TID (Μόνο για πληρωμή μέσω EFTPOS)	Yes
EftposTransType	String	EFTPOS Τύπος Συναλλαγής 0. Αγορά 1. Αγορά με δόσεις 2. Επιστροφή 3.Ακύρωση 4. Καταχώριση Προέγκρισης 5. Mail order 6.Προφορτωμένη (Μόνο για πληρωμή μέσω EFTPOS)	Yes
Value	Decimal	Αξία πληρωμής	Yes

EftposPayResult	String	EFTPOS Reply Data (μετά από συναλλαγή που πραγματοποιήθηκε με λήψη Token από το ΦΗΜΑΣ)	NO
PrintLine			

InvoiceFooter

Name	Type	Description	Required
Line1	String	Footer Line #1	No
Line2	String	Footer Line #2	No
Line3	String	Footer Line #3	No
Line4	String	Footer Line #4	No
Line5	String	Footer Line #5	No
Line6	String	Footer Line #6	No
PrintLine			

Json Example


```
{
   "Doc":{
      "ReceiverHeader":{
         "CompanyTitle":"",
         "Profession":"",
         "Address":"",
         "TaxID":"",
         "TaxOffice":"",
         "PhoneNumber":"",
         "PrintLine":""
      },
      "InvoiceDetails":{
         "InvoiceUID":"3a01dfb7-28d0-44ca-8d35-f9dc07f68e0b",
         "InvoiceType":0,
         "ReqForToken":0,
         "CancelInvType":null,
         "CancelDevDailyNum":null,
         "CancelInvNo":null,
         "CancelInvSeries":null,
         "InvoiceNo":"2",
         "invoiceSeries":null,
         "InvoiceTotal":0.00,
         "Operator":"user1",
         "Machine":"Shop",
         "GasStationLicNum":0,
         "GasStationInstalNum":0,
         "InvWithholdingTaxTotal":0.00,
         "PrintLine":"User1     shop\n01-01-2022  00:00:00\nΣΥΝΟΛΟ                    26.46E\n"
      },
      "TransactionLines":[
         {
            "LineNo":1,
            "ClassCategory":0,
            "ClassType":null,
            "FuelCode":null,
            "Code":"123456789",
            "Description":"Item #1",
            "ItemAmount":90.33,
            "MeasurementUnit":1,
            "SaleQty":1,
            "NetAmount":9.94,
            "VatAmount":2.39,
            "GrossAmount":12.33,
            "DMType":0,
            "DMValue":0,
            "DMPercentage":0,
            "VatRate":"24.00",
            "VatExCategory":0,
            "FeeCategory":0,
```

```
            "PrintLine":"Item #1                        12.33 24%\n"
         },
      ],
      "Payments":[
         {
            "Type":1,
            "Description":null,
            "EftposDescr":"EFTPOS-1",
            "EftposTID":null,
            "EftposTransType":0,
            "Value":14.13,
            "PrintLine":"KAPTA                            14.13E\n"
         }
      ],
      "InvoiceFooter":{
         "Line1":"LINE1",
         "Line2":"LINE2",
         "Line3":"LINE3",
         "Line4":"LINE4",
         "Line5":"LINE5",
         "Line6":"LINE6",
         "PrintLine":"LINE1\nLINE1\nLINE1\nLINE1\nLINE1\nLINE1\n"
      }
   }
}
```

8.5. Command Protocol ERRORS

HEX	DEC	Description
00	0	No errors - success
01	1	Wrong number of fields
02	2	Field too long
03	3	Field too small
04	4	Field fixed size mismatch
05	5	Field range or type check failed
06	6	Bad request code
07	7	FM Record Number
08	8	FM Record Type
09	9	Printing type bad
0A	10	Cannot execute with day open
0B	11	RTC programming requires jumper
0C	12	RTC date or time invalid
0D	13	No records in fiscal period
0E	14	Device is busy in another task
0F	15	No more header records allowed
10	16	Cannot execute with block open
11	17	Block not open
12	18	Bad data stream
13	19	Bad signature field
14	20	Z closure time limit
15	21	Z closure not found
16	22	Z closure record bad
17	23	User browsing in progress
18	24	Signature daily limit reached
19	25	Printer paper end detected
1A	26	Printer is offline
1B	27	Fiscal unit is offline
1C	28	Fatal hardware error
1D	29	Fiscal unit is full
1E	30	No data passed for signature
1F	31	Signature does not exist
20	32	Battery fault detected
21	33	Recovery in progress
22	34	Recovery only after CMOS reset
23	35	Real-Time Clock needs programming
24	36	Z closure date warning
25	37	Bad character in stream
26	**38**	**FM Initialization fail**
27	39	Filesystem failure
28	40	SD Disconnection
29	41	Invalid GSIS Key
2A	42	Invalid data stream number
2B	43	Invalid Issuer TIN
2C	44	Invalid Client TIN
2D	45	Wrong Checksum
2E	46	Sign of data stream amounts is wrong
2F	47	No daily signatures
30	48	Busy in clock synchronization from GSIS
31	49	Exceed RTC minutes

32	50	Same Header
33	51	Empty Header
34	52	Jumper is on
35	53	Wrong Clerk Index
36	54	Wrong Payment Index
37	55	Inactive Payment
38	56	Busy in payment
39	57	Payment does not allowed changes
3A	58	Open Cash In/Out trsansaction
3B	59	Wrong Discount/Markup Index
3C	60	Discount/Markup Limit
3D	61	Zero Dicount/Markup amount
3E	62	BMP wrong data
3F	63	Wrong BMP Index
40	64	Invalid VAT Rate
41	65	Wrong amount
42	66	Invalid sales operation
43	67	No more VAT Rates changes
44	68	Not Supported
45	69	Wrong BAUD Rate
46	70	Wrong Quantity
47	71	No transactions
48	72	Blank description
49	73	Busy in service operation
4A	74	SD Full
4B	75	Invalid SD
4C	76	Format SD Fail
4D	77	Cannot open SD File(WR)
4E	78	Cannot Write SD File
4F	79	Cannot open SD File(RD)
50	80	FM Communication failure
51	81	Negative total
52	82	Wrong Barcode data
53	83	Busy in error message from keyboard usage
54	84	Limit of invoice total
55	85	Limit of day total
56	86	Battery failure
57	87	Full EJ
58	88	Invalid IP Address
59	89	Invalid TIN
5A	90	Must program TIN
5B	91	Empty EJ
5C	92	Cannot open EJ file (RD)
5D	93	Cannot Open EJ File (WR)
5E	94	Cannot Write EJ File
5F	95	Amount limit
60	96	Inactive Clerk
61	97	Call Technician
62	98	Ethernet communication
63	99	GSIS Communication
64	100	Empty FM
65	101	Must set Quantity
66	102	Wrong activation code

67	103	Cannot write the change of Header in FM
68	104	Unregister Device
69	105	Zero invoice total
6A	106	Wrong Activation Key
6B	107	Wrong Password
6C	108	Invalid Company Category
6D	109	Invalid SD
6E	110	The day is closed
6F	111	Cannot set Fee
70	112	Busy in GSIS Communication
71	113	Wrong GSIS Key
72	114	Inactive GSIS Communication
73	115	FM is closed
74	116	Incative External Printer
75	117	Connection error with External Printer
76	118	Licence expiration
77	119	Incative EFTPOS
78	120	IP Reserved
79	121	EFTPOS Transaction failure
7A	122	EFTPOS Get Transactions error
7B	123	EFTPOS Transaction in progress
7C	124	Wrong NET Address
7D	125	Sign in progress
7E	126	Wrong state of sign procedure
7F	127	Wrong Sign data
80	128	Invalid sign character
81	129	No data to sign
82	130	Memory allocation error
83	131	Cannot parse JSON
84	132	No transactions lines inJSON
85	133	No payment lines in JSON
86	134	Wrong JSON Classification category
87	135	Wrong JSON Classification Type
88	136	Wrong JSON Fuel code
89	137	Amounts of JSON transaction line does not much
8A	138	Wrong JSON Measurement Unit
8B	139	Wrong JSON Discount/Markup type
8C	140	Wrong JSON Discount/Markup amount
8D	141	Wrong JSON Exemption category
8E	142	Wrong JSON Fee Category
8F	143	Wrong JSON Payment type
90	144	Wrong JSON Payment EFTPOS TID
91	145	Wrong JSON Payment EFTPOS Transaction type
92	146	Wrong JSON Payment total
93	147	Wrong JSON Invoice UID
94	148	Wrong JSON Printing Device
95	149	Wrong JSON Request for token
96	150	Wrong JSON Invoice type
97	151	Wrong JSON Invoice number
98	152	Wrong JSON Invoice series
99	153	Wrong JSON Invoice total
9A	154	Requested Invoice to Cancel invalid type
9B	155	Requested Invoice to Cancel not found

9C	156	Requested Invoice to Cancel zero total
9D	157	Not send 10 Z to GSIS
9E	158	Wrong JSON Payment EFTPOS Description
9F	159	Wrong/Inactive EFTPOS Transaction Type
A0	160	Wrong JSON Withholding Total
A1	161	Wait FW Upgrade Confirmation
A2	162	Pending Tokens exists
A3	163	Not valid EFTPOS Payment Result data
A4	164	Wrong Token information

9. UPDATES

DATE	Version	WHATs NEW
8/4/24	1.0.3	• New Command EFTPOS Transaction PREPAYMENT,INVOICE,TAXFREE)[6] added • New Command EFTPOS Result [5] added and returns all the payment information from EFTPOS terminal.
21/4/24	1.0.4	• 8.2.42 was modified and now supports 2 more states for cases of confirmation or cancellation incase of myData document through 355. • New command (8.2.44) added and supports requests for ecrTokens. • New command (8.2.45) added and supports keyboard key requests. • Error table appended with 4 more errors • Json Schema Modified with new DMtype (5) and now supports refund type transaction line. • Json Schema Modified with 2 new fields EftposPayResult & ClassType. • Json Schema modified with updated the field ReqForToken and now supports close and cancel invoice from ecrToken
29/4/24	1.0.5	• 8.2.42 update with additional EFTPOS Tid field. • 1155 Modifications regarding the pending payments functionality.
21/05/24	1.0.6	• 8.1.2.2.1 Update Device Status bits • 8.1.2.2.2 Update Fiscal Status • 8.2.1. Read Device identification [a], Update reply fields • 8.2.2.Read version/device info, Update reply fields • 8.2.19. Get Invoice info [=], Update reply fields • 8.2.20. Get Last Invoice Info [9], add 2 more fields at the end (qrcode data, uid) • 8.2.22. Read EFTPOS last Invoice info [5/99/], Update reply fields • 8.2.26. Program payment type [Y], update fields • 8.2.31. Program VAT rates [b], update fields • 8.2.35. Read Z report record [R] , Update reply fields • 8.2.36. Read VAT Rates [V], update fields • 8.2.42. EFTPOS Transactions(Invoice,TaxFree,Prepayment, Tokens [6], update fields, add choice to Cancel all Pending Tokens, allow cancel Receipt Token

		• **8.2.43. Request For ecrToken [_](POL 1155), add more choices to read Token by UID, read List of Daily Pending Tokens**
24/05/24	1.0.7	• **8.2.43. Request For ecrToken [_](POL 1155), add EFTPOS TID when reply info of Token** • **8.2.45. Send RESEND-ALL to EFTPOS [\/17/], new command**