CITIZEN CT-S601
PROTOCOL SPECIFICATION

Status: Final release

Release: V1.R2.T0

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 with fiscal ECR/POS models.

This document assumes that the reader is familiar with basic communication concepts, such as transmittances, receptions, 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)
- 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 line

The ECR/POS communicates with host computer via an asynchronous serial line of the RS-232C recommended standard. The serial line parameters are:

- Baud rate: 9600 baud
- Parity: none
- Data: 8
- Stop: 1
- Flow control: none

Note that because there is no flow control, only the RX/TX/GND signals are required for the cable configuration. The maximum length of cable is described in the 232C recommended standard for this baud rate. It is highly recommended that the maximum length is not exceeded to prevent drops in communication rate and undesirable retransmittances due to errors, or in worst case a total communication failure. When cable distance is unavoidably long, an extender may be used.

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 more clear, 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.

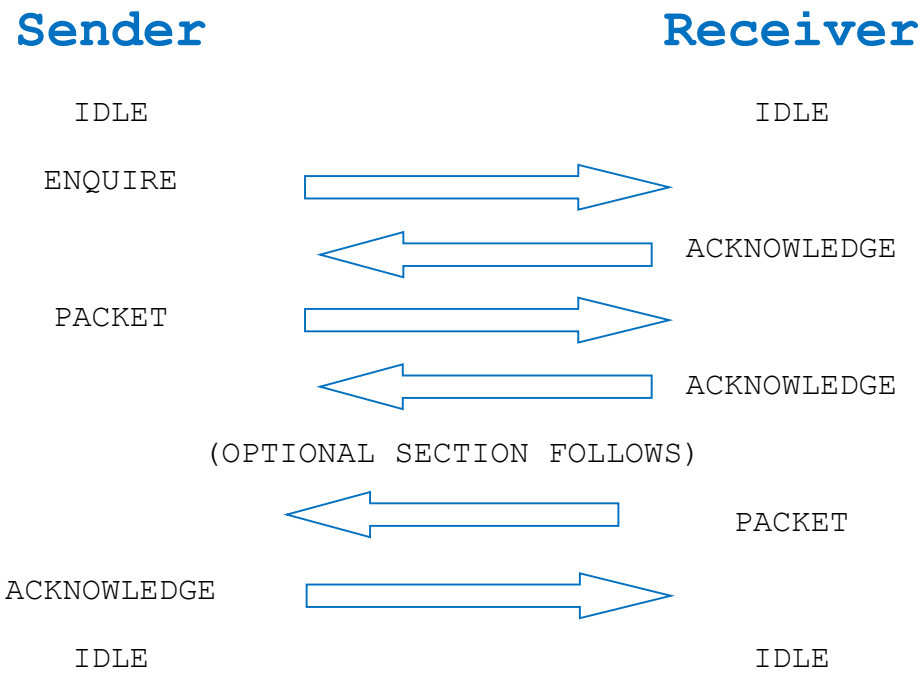
Although these two layers cannot coexist at the same time of POS operation, switching between them is allowed anytime. As expected, communication rules and procedures that layers use are the same.

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 (POS and host computer) does not include any other situation such as errors in transmittance, retransmittance 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.

- **Enquire state**

  The sender that wishes to initiate communication sends an inquiry to the receiver. The process of sending this inquiry is the enquire state so only the sender enters this state.

- **Acknowledge state**

The receiver will enter this state right after receiving an inquiry or after the verification of a request packet. The sender will enter this state after the verification of a reply packet.

- **Verify acknowledge state**

  The sender or receiver will enter this state after an enquire state or a packet transmittance state. The process of waiting the other end's positive or negative response is to verify acknowledge state.

- **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 sender's 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
Enquire state / Verify ack. State	Acknowledge state
Packet transmittance state	Packet reception state
Verify acknowledge state	Acknowledge state
Packet reception state	Packet transmittance state
Acknowledge state	Verify acknowledge state
Idle	Idle

6.2.1. States definition -> Enquire state

The enquire state is actually the transmittance of a single ASCII control code ENQ [CC1] by the sender. Doing this, the sender has concluded the enquire state. The purpose of this state is to find out if the receiver is able to reply, without flooding the communication line with too much data. After sending the ENQ code, the sender must wait for a response from the receiver, entering verify acknowledge state (see 6.2.2). It is highly recommended to clear the receiving buffer before entering an enquire state, so discarding any accidental data previously received in the serial communication's receive buffer, especially in cases where serial communication is interrupt driven.

Some synchronization needs may also require that before sending the ENQ code, hosts should send the CAN (cancel) [CC1] control code to cancel any waiting states in the ECR/POS side.

6.2.2. States definition -> Verify acknowledge

The verify acknowledge state is the reception of a response code which indicates that an action from one side has been accepted by the other. For this to work, the ASCII control codes ACK and NAK [CC1] are used to mean positive or negative acknowledgement respectively. In this state the sender or the receiver enters in the following cases:

- after an enquire state by the sender
- after a request packet transmittance by the sender
- after a reply packet transmittance by the receiver

In any of the above cases, the side which is in the verify acknowledge state must either accept ACK or NAK as valid responses within some specific time window. Any other received control values should be treated as NAK.

On reception of an ACK, the host must leave the verify acknowledge state and proceed to the next state, if any. This means that the previous state was successfully processed by the other side of the communication. On reception of a NAK, the host must leave the verify acknowledge state and repeat once more the previous state. For example, if the verify acknowledge state was for a previous enquire state, the enquire state must be repeated. If the request packet was not acknowledged, the packet must be retransmitted.

To prevent infinite communication loops, each of these cases mentioned are limited to a specific retransmittance count, which, when reached, indicates that the communication attempt causing the retransmittances was unsuccessful and further communication is not possible for some reason. The possible reasons for such a failure may be:

- Disconnection of serial cable
- Host computer or ECR/POS fatal error
- Too noisy communication line

6.2.3. States definition -> Acknowledge state

The acknowledge state is the transmittance of either ACK or NAK control codes after a previous enquire or packet reception. ACK must be transmitted when the enquire is accepted or the packet is verified successfully. This is 'positive acknowledge'. NAK must be transmitted when the enquire must be either delayed or rejected, or if the packet failed checksum verification. This is 'negative acknowledge'. Hosts must not transmit any other codes except ACK, NAK and CAN in this state.

6.2.4. 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 [see 6.3]. 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, ie they will not be included in the data section of the packet.

6.2.5. 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. Packet reception is initiated with the reception of the STX control code [CC1]. Any reception of data before the reception of STX must be silently discarded. Packet reception is terminated with the reception of ETX control code [CC1]. Any data after the termination code (ETX) do not belong to this state. See next paragraph for packet handling and structure.

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.

The packet structure is the following:

STX	Data	ETX

Notice that the actual data is between STX and ETX fields which are simply the ASCII control codes STX and ETX [CC1]. By ASCII definition, the STX/ETX control codes indicate the start of data transmittance and the end of data transmittance respectively. Any valid octet between the STX and ETX 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 and this also applies for the STX/ETX control codes.

The length of the data section is variable, due to its multifunctioning purpose. ECR/POS is able to 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 has to 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. Packet verification - error detection

To ensure that a request or reply packet was received with no errors, both layers use a special field: the checksum. Checksum is always the last field in the packet in all cases of packet transmittances. It also must be separated from the previous field using the slash (/). Checksums are always a 2-digit decimal values and represent the modulo

100 of the 8-bit sum of all data octets in the packet except any control codes or the 2-digits checksum itself but including the field separators. All field separators are calculated in the checksum.

Example checksum calculation function in 'C':

```
BYTE CalcChecksum(BYTE *packet)
{
BYTE sum = 0;
int checklength = strlen(packet) - 2;
while(checklength--) sum += (BYTE) (*packet++);
return( (sum % 100) );
```

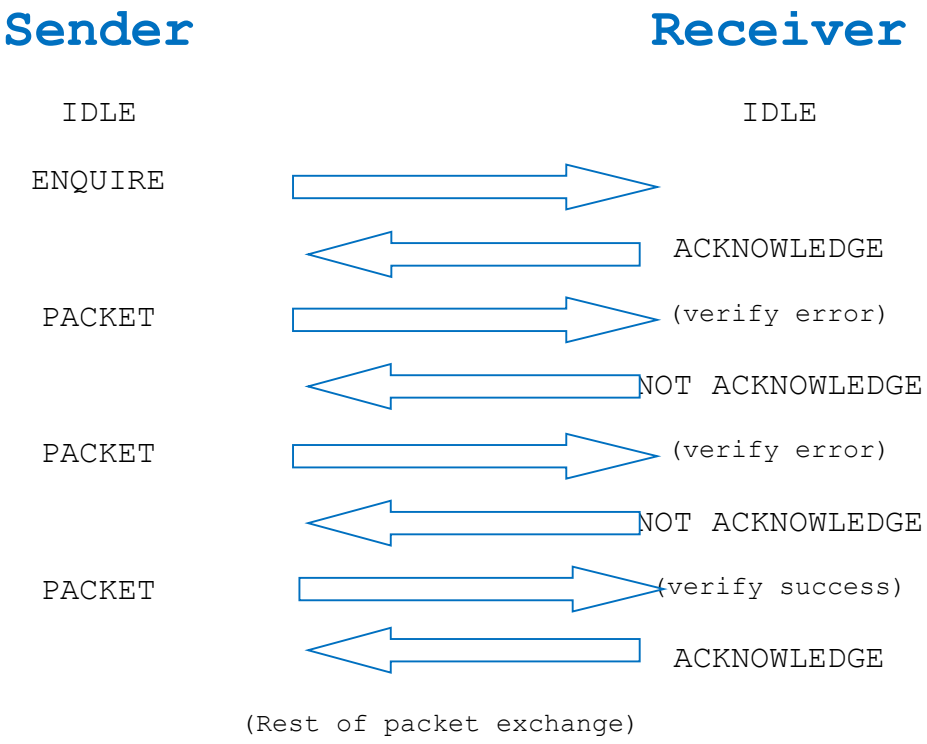
Example checksum calculation function in pseudo code:

```
Function Calculate_Checksum( parameter data_packet ) Returns BYTE
      Begin
        Declare CALCSUM, I as BYTE
        CALCSUM = 0
        For I = 0 to stringlength( data_packet ) - 2 Do
          CALCSUM  =  CALCSUM + ASCII( data_packet[ I ] ) )
        Next  I
        CALCSUM = CALCSUM mod 100
        Return CALCSUM
      End
```

The receiver of the packet must calculate this checksum locally, compare it with the transmitter's checksum and, if found equal, the packet is valid and a positive acknowledgement must be sent. Otherwise the packet was corrupted and a negative acknowledgement must be sent. The checksum will always be a numeric, 2-digit field in range 00-99.

Data section				
Layer fields				Checksum
Field 1	Field 2	Field 3	……..	CC

Remembering the state paragraphs above, negative acknowledgements in packet receptions cause retransmittances of the packet. The scheme that follows describes one such case where the packet failed checksum verification twice and succeeded in the third:



6.3.2. 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 in order 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 process. 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 fields restrictions and range

6.3.3. 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 packets for device status map and fiscal status map fields.

6.3.4. 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.

NUM type	
Class:	Numeric
Value range:	0 to 9
Digit Range:	1 digit
Notes:	Fields of this type must not contain any decimal part or decimal point.

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' 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.

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 (ie: 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. Online protocol

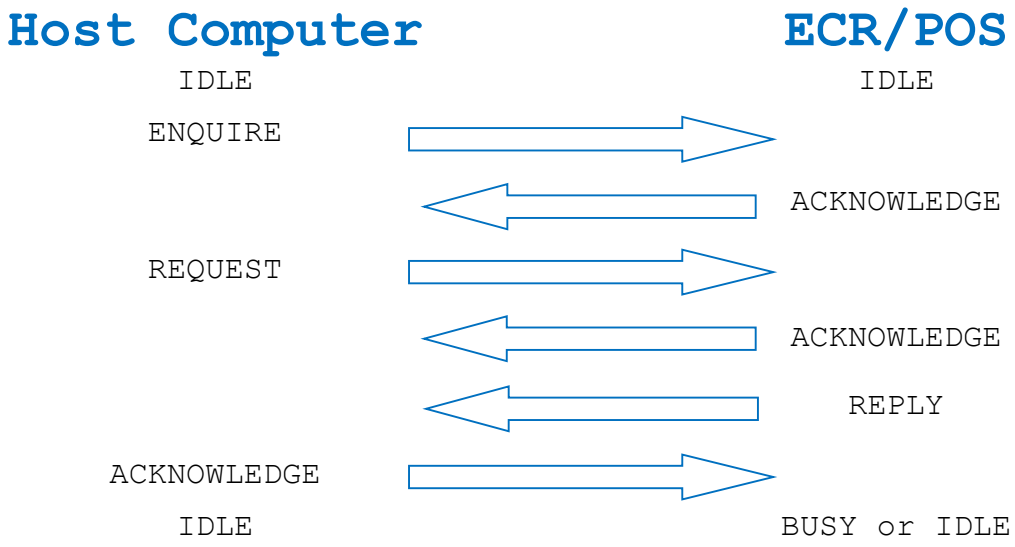
Not Available.

8. Command protocol

The command protocol is initiated by the host computer, when the host wants to instruct the ECR/POS 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

The model of the communication the command protocol follows is this:



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]> [checksum]

Reply packet: [Reply code] / [status fields] / <[Reply data]> [checksum]

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	Checksum

This defines 3 sections of a request packet:
- The request code section
- The data field section
- The checksum 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	Checksum

This defines 4 sections of a reply packet:
- The reply code section
- The status section
- The data field section
- The checksum 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 executed successfully. A non zero reply code indicates 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 was NOT executed. Receiving a zero reply code means that the command has been or will be successfully executed. Commands that require very little time to execute, 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
CUTTER	TMOUT	FFULL	PCONN	BATWARN	PP.END	FATAL	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 non-critical 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: Fatal error

This bit indicates that (when set to one) the ECR/POS detected a fatal hardware related error and cannot process most of the commands. Fatal errors may be a bad fiscal unit, a RAM integrity error or others. From application point of view, this bit means that other critical commands should not be sent, and a service to the ECR/POS is required.

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: 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 4: 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 5: 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.

Bit 6: Printer timeout

This bit indicates (when set to one) that the printing device is not responding to printing commands. This may be caused by printer's cover which may be open. User must check the cover and close it to continue printing operations. If this is not caused by an open cover and persists after a power off - power on, then the ECR/POS must be serviced.

Bit 7: Cutter Error

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.

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
--	EJOPEN	COUT	CIN	PAYM	TROPEN	DAYOPEN	DROPEN

Bit 0: Drawer is open

This flag indicates that the drawer is open.

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: Transaction (Receipt) Open

This flag is indicating that a receipt is currently in 'open' state in the ECR/POS. The flag will be set even if the receipt is in 'payment' state. When this bit is set, information related to an open receipt is valid. An application can prevent errors in commands by detecting this bit. For example, a command 'issue Z report' will fail if this bit is set.

Bit 3: Transaction in Payment

This flag indicates that ECR/POS has an open receipt in payment state. If it is set, the bit 2 (transaction open) will be also set.

Bit 4: Cash in open

This flag indicates that a cash in receipt is open

Bit 5: Cash out is open

This flag indicates that a cash in receipt is open

Bit 6: Electronic Journal Report Open.

This flag indicates that ECR/POS has an electronic journal report in progress.

Bit 7: (Reserved)

Example: Assume fiscal status field is '16'. This hexadecimal value, when converted to binary will be '00001110'. 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. Program header [H]

Programs the header in the device. The header is stored in the fiscal memory. 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 centered automatically.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**H**	18 (Counting request code & checksum field)	16 (Without request code & checksum field)	"H/1/HEADER LINE 1/2/HEADER LINE 2/1/HEADER LINE 3/1/HEADER LINE 4/3/HEADER LINE 5/4/HEADER LINE 6/1/HEADER LINE 7/1/HEADER LINE 8" (checksum)

	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 When printing double width, only 24 characters of the line are printed.
FIELD 3	Header line text	STRING	0-48 (0-24) chars	The text data for each line. (0-24) characters if double width character
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 When printing double width, only 24 characters of the line are printed.
	FIELD 5	Header line text	STRING	0-48 (0-24) chars	The text data for each line. (0-24) characters if double width character
	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 When printing double width, only 24 characters of the line are printed.
	FIELD 7	Header line text	STRING	0-48 (0-24) chars	The text data for each line. (0-24) characters if double width character
	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 When printing double width, only 24 characters of the line are printed.

	FIELD 9	Header line text	STRING	0-48 (0-24) chars	The text data for each line. (0-24) characters if double width character
	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 When printing double width, only 24 characters of the line are printed.
	FIELD 11	Header line text	STRING	0-48 (0-24) chars	The text data for each line. (0-24) characters if double width character
	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 When printing double width, only 24 characters of the line are printed.
	FIELD 13	Header line text	STRING	0-48 (0-24) chars	The text data for each line. (0-24) characters if double width character
	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 When printing double width, only 24 characters of the line are printed.
	FIELD 15	Header line text	STRING	0-48 (0-24) chars	The text data for each line. (0-24) characters if double width character
	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 When printing double width, only 24 characters of the line are printed.
	FIELD 17	Header line text	STRING	0-48 (0-24) chars	The text data for each line. (0-24) characters if double width character

	TOTAL FIELD COUNT	**DATA FIELD COUNT**	**EXAMPLE REPLY**
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.2. Read header [h]

Reads the current (active) header setting in the device. Also returns the times that this header is programmed and the times that are remaining for reprogramming.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**h**	2 (Counting request code & checksum field)	0 (Without request code & checksum field)	"h" (checksum)

	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	22 (Counting reply code, status & checksum field)	18 (Without reply code, status & checksum field)	(reply code)(status) "1/HEADER LINE 1/2/HEADER LINE 2/1/HEADER LINE 3/1/HEADER LINE 4/1/HEADER LINE 5/1/HEADER LINE 6/1/HEADER LINE 7/2/1/HEADER LINE 8/4/46" (checksum)

	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 When printing double width, only 24 characters of the line are printed.

	FIELD 2	Header line text	STRING	0-48 (0-24) chars	The text data for each line. (0-24) characters if double width character
	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 When printing double width, only 24 characters of the line are printed.
	FIELD 4	Header line text	STRING	0-48 (0-24) chars	The text data for each line. (0-24) characters if double width character
	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 When printing double width, only 24 characters of the line are printed.
	FIELD 6	Header line text	STRING	0-48 (0-24) chars	The text data for each line. (0-24) characters if double width character
	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 When printing double width, only 24 characters of the line are printed.
	FIELD 8	Header line text	STRING	0-48 (0-24) chars	The text data for each line. (0-24) characters if double width character
	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 When printing double width, only 24 characters of the line are printed.
	FIELD 10	Header line text	STRING	0-48 (0-24) chars	The text data for each line. (0-24) characters if double width character
	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 When printing double width, only 24 characters of the line are printed.
	FIELD 12	Header line text	STRING	0-48 (0-24) chars	The text data for each line.

					(0-24) characters if double width character
	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 When printing double width, only 24 characters of the line are printed.
	FIELD 14	Header line text	STRING	0-48 (0-24) chars	The text data for each line. (0-24) characters if double width character
	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 When printing double width, only 24 characters of the line are printed.
	FIELD 16	Header line text	STRING	0-48 (0-24) chars	The text data for each line. (0-24) characters if double width character
	FIELD 17	Count of header records written	INTEGER	1-2 digits	The number of times the title is programmed in fiscal memory.
	FIELD 18	Count of header records remaining	INTEGER	1-2 digits	The number of times the title remains to be programmed.

8.2.3. Program the Real-Time Clock/Calendar [T]

This command is used for programming the device's real time clock (ie: 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	**T**	4 (Counting request code & checksum field)	2 (Without request code & checksum field)	"T/110313/161800" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'T' for this command.
FIELD 2	System date	DATE6	Default (fixed 6)	The date to set in RTC (Real time clock).
FIELD 3	System time	TIME	Default (fixed 6)	The time to set in RTC.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.4. Read the Real-Time Clock/Calendar [t]

This command is used to read the device's real time clock.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	t	2 (Counting request code & checksum field)	0 (Without request code & checksum field)	"t" (checksum)

	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	6 (Counting reply code, status & checksum field)	2 (Without reply code, status & checksum field)	(reply code)(status) "210913/151020" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	System date	DATE6	Default (fixed 6)	The current date in device.
	FIELD 2	System time	TIME	Default (fixed 6)	The current time in device.

8.2.5. Read Device ID/S-N [a]

This command is used to read the device's serial number

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**a**	2 (Counting request code & checksum field)	0 (Without request code & checksum field)	"a" (checksum)

	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	5 (Counting reply code, status & checksum field)	1 (Without reply code, status & checksum field)	(reply code)(status) "ABC12312312345" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Device serial number	STRING	Fixed, 11 digits (3 letters, 8 digits)	The device's unique serial number

8.2.6. Display message [7]

This command is used to show a message to display unit. The messages appear in the external LCD or VFD connected to the port FM of the CITIZEN CT-S601's. The serial cable must have its pins directed as follow: 2--→3, 3--→2

The type of the LCD or VFD that is used, is defined by the field 13 of the command 'S' that programs the CITIZEN CT-S601's parameters (2.8.19)

0= is LCD type (Micrelec 2 x 16) 1= VFD (Epson compatible)

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**7**	8 (Counting request code & checksum field)	6 (Without request code & checksum field)	"7/1/TEST MESSAGE" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be '7' for this command.
FIELD 2	Line number	INTEGER	Fixed, 1 digit (0-2)	The display line to show the message. If zero, the display is cleared and the message in field 3 is ignored. Otherwise, it can be either 1 or 2 specifying the line.
FIELD 3	Message	STRING	1 to 24 chars	The text shown is limited by the display width, which may vary depending on model. The safest text size though is 16 characters, because it is guaranteed that all compatible models will support it.

REPLY PACKET	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.7. Read Version [v]

The read version commands in useful for retrieving the protocol version. See also paragraph [3] for version compatibility issues.

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

	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	7 (Counting reply code, status & checksum field)	3 (Without reply code, status & checksum field)	(reply code)(status) "MAT/CTS601G2\A/V1 R1 T7" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Vendor information	STRING	1-48 chars	A vendor information string.
	FIELD 2	Model information	STRING	1-48 chars	A model information string. This can be useful in determining specific physical information about the device (i.e. display width, max signatures in day etc).
	FIELD 3	Version	STRING	1-16 chars	Contain the protocol version.

8.2.8. Read Device Status [?]

This command has no additional input output data. It is only used for getting the status codes from the device. Otherwise it is a NOOP (no operation).

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**?**	2 (Counting request code & checksum field)	0 (Without request code & checksum field)	"?" (checksum)

	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	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.9. X/Z report [x]

This command is for validating the successful transfer after daily closure (Z) report or issuing copy of last Z or (X) statistical sales.

Caution: To transfer daily flash data, the electronic journal must be read first (see command 'A'-8.2.17 & 'Q' 8.2.18)

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**x**	3 (Counting request code & checksum field)	1 (Without request code & checksum field)	"x/1" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'x' for this command.
	FIELD 2	Report selector	INTEGER	1 digit	'1' = Daily to totals whit out zeroing data (X) '2'= Daily Flash data transfer was successful '5'= Daily Flash data transfer was unsuccessful '9' = Issue the last Z report

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.10. Fiscal report (date to date) [f]

This command is for issuing a date-to-date fiscal report. (Read data from fiscal memory)

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**f**	8 (Counting request code & checksum field)	6 (Without request code & checksum field)	"f/010113/310813" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'f' for this command.
	FIELD 2	Start date	DATE6	Default	The starting date that defines the requesting fiscal period.
	FIELD 3	End date	DATE6	Default	The ending date that defines the requesting fiscal period.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.11. Fiscal report (Z to Z) [z]

This command is for issuing a Z-to-Z fiscal report. (Read data from fiscal memory)

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**z**	4 (Counting request code & checksum field)	2 (Without request code & checksum field)	"z/150/320" (checksum)

	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 starting Z number that defines the requesting fiscal period.
	FIELD 3	End Z number	INTEGER	1-4 digits	The ending Z number that defines the requesting fiscal period.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.12. Item sale [3]

This command belongs to the fiscal printer commands. It is used to sale an item remotely. If a transaction is not open, the ECR/POS will open it. Not all fields in this command are optional.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**3**	12 (Counting request code & checksum field)	10 (Without request code & checksum field)	"3/S/PLU CODE/ITEM-1/ADDITIONAL INFO/BARCODE/1.000/100.00/1/4/CATEGORY CODE" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	1 character	Must be '3' for this command.
FIELD 2	Operation	STRING	1 character	The operation code must be one of the following: 'S' for positive sale, 'V' for void (negative) sale, and 'R' for refund
FIELD 3	INTERNAL PLU CODE (IT IS NOT OBLIGATORY)	NUM	0-3 digits	It is the INTERNAL PLU CODE (1-200)
FIELD 4	Item description	STRING	1-35 chars	The description of the item (required)
FIELD 5	Sale extended description line	STRING	0-35 chars	An extra information line printed below the 'sale' line (optional)
FIELD 6	Barcode (or other extra string)	STRING	0-16 chars	A barcode code or other printable string
FIELD 7	Sales quantity	QTY	1-8 digits	The item sale quantity
FIELD 8	Item unit price	AMOUNT	1-10 digits	The item's unit price for the sale

	FIELD 9	Vat code or Department code	INTEGER	1-2 digits	The vat code (1='A', 2='B'...5=’E’) (the vat code & department code is linked)
	FIELD 10	Item Vat rate	PERCENTAGE	1-5 digits	The VAT rate that applies to this item. This rate MUST be equal to the VAT rate is programmed in the EPSON 6000 (1=6.5%, 2=13%, 3=23%, 4=36%, 5=0%)
	FIELD 11	CATEGORY CODE (1-20)	INTEGER	0-2 digits	It is the sales category that the item can belong to (1-20) This field is optional. If not sent then the sale will occur according to the category the item’s department belongs.

	TOTAL FIELD COUNT	**DATA FIELD COUNT**	**EXAMPLE REPLY**
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.13. Discount or Markup [4]

This command is for issuing discounts or markups to the ΑΔΗΜΕ printer. A transaction must be open.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**4**	12 (Counting request code & checksum field)	10 (Without request code & checksum field)	"4/1/1/DISCOUNT IN SALES/EXTRA DESCR/12.75/0/0/0/0/0" (checksum) "4/2/1/DISCOUNT IN SUBT/EXTRA DESCR/12.75/2.75/2/3/1/4" (checksum) "4/3/1/MARKUP IN SALES/EXTRA DESCR/0.50/0/0/0/0/0" (checksum) "4/4/1/MARKUP IN SUBTOTAL/EXTRA DESCR/5.00/1/2/0/1/1" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be '4' for this command.
FIELD 2	Type of Discount Markup	INTEGER	1 digit	The type of discount markup 1=Discount in sales 2=Discount in subtotal 3=Markup in sales 4=Markup in Subtotal.
FIELD 3	The VAT Code in which the Discount/Markup according to the corresponding type will occur. (If the type is for Discount/Markup in subtotal, then it can always be 1).	INTEGER	1 digit	The VAT Code can be 1-5 (1=A, 2=B, 3=C, 4=D, 5=E)

	FIELD 4	The operation description	STRING	0-35 chars	Optional string for description of operation. If not passed, the default string will be used.(DISCOUNT, SUBTOTAL DISCOUNT, MARKUP, SUBTOTAL MARKUP)
	FIELD 5	Operation extended description	STRING	0-35 chars	Optional string for additional information printing of the operation. Prints one additional line below the operation printing lines.
	FIELD 6	Amount of operation	AMOUNT	1-10 digits	The amount of the discount/markup. An additional value will be subtracted from VAT Code' sale total (Field 3). **If the Discount/Markup is on the subtotal, then its value must be equal with the sum of the fields 7-11 (Allocation Total).**
	FIELD 7	Discount/Markup Allocation Amount. In VAT **A**	AMOUNT	0-10 digits	If Discount/Markup on subtotal then this is the amount that will be subtracted (if it exists) from the VAT A value
	FIELD 8	Discount/Markup Allocation Amount. In VAT **B**	AMOUNT	0-10 digits	If Discount/Markup on subtotal then this is the amount that will be abstracted (if it exists) from the VAT B value
	FIELD 9	Discount/Markup Allocation Amount. In VAT **C**	AMOUNT	0-10 digits	If Discount/Markup on subtotal then this is the amount that will be abstracted (if it exists) from the VAT C value
	FIELD 10	Discount/Markup Allocation Amount. In VAT **D**	AMOUNT	0-10 digits	If Discount/Markup on subtotal then this is the amount that

					will be abstracted (if it exists) from the VAT D value
	FIELD 11	Discount/Markup Allocation Amount. In VAT **E**	AMOUNT	0-10 digits	If Discount/Markup on subtotal then this is the amount that will be abstracted (if it exists) from the VAT E value

	TOTAL FIELD COUNT	**DATA FIELD COUNT**	**EXAMPLE REPLY**
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.14. Payments in receipt [5]

When a receipt is open, this command will force the ECR/POS firmware state to enter payment mode.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**5**	6 (Counting request code & checksum field)	4 (Without request code & checksum field)	"5/2/CREDIT CARD/DINERS-12345678/12.56" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be '5' for this command.
FIELD 2	Payment type (index)	INTEGER	1 digit	The payment code as follows: 1=CASH 2=CARDS 3=CREDIT These 3 codes are used only for the storing of the Payments modes, in order for a Z Report to be issued
FIELD 3	The operation description	STRING	1-35 chars	Optional string for description of operation. If not passed, the default string will be used. (CASH, CARD, CREDIT)
FIELD 4	The operation extra description	STRING	1-35 chars	Optional string for extra description of operation.
FIELD 5	Payment Amount	AMOUNT	1-10 digits	The amount for the payment. If the amount is 0 then the receipt is closed containing the whole sum.

REPLY PACKET	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
	5 (Counting reply code, status & checksum)	1 (Without reply code, status & checksum)	(reply code)(status) "-4.00" (checksum)"

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Balance to be paid (AMOUNT)	AMOUNT	Default	Balance to be paid. If negative then it is change.

8.2.15. Read transaction totals [9]

This command is used for getting the current transaction totals when a receipt is currently open. If a receipt is not open, the transaction totals will be zero.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**9**	2 (Counting request code & checksum field)	0 (Without request code & checksum field)	"9" (checksum)

	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	11 (Counting reply code, status & checksum field)	7 (Without reply code, status & checksum field)	(reply code)(status) "100.00/200.00/300.00/400.00/500.00/17/1500.00" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Receipt Accumulators	AMOUNT	Default	Receipt's sums belonging to VAT A category
	FIELD 2	Receipt Accumulators	AMOUNT	Default	Receipt's sums belonging to VAT B category
	FIELD 3	Receipt Accumulators	AMOUNT	Default	Receipt's sums belonging to VAT C category
	FIELD 4	Receipt Accumulators	AMOUNT	Default	Receipt's sums belonging to VAT D category
	FIELD 5	Receipt Accumulators	AMOUNT	Default	Receipt's sums belonging to VAT E category

	FIELD 6	Receipt number	INTEGER	1-6 digits	The receipt's number
	FIELD 7	Transaction Total	AMOUNT	Default	The amount that requires payment before the transaction can be closed. If the receipt is not in payment state, this amount equals to the sum of all VAT accumulators. When the receipts is in payment state, it shows the amount remain to be paid.

8.2.16. 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**	2 (Counting request code & checksum field)	0 (Without request code & checksum field)	"0" (checksum)

	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	22 (Counting reply code, status & checksum field)	18 (Without reply code, status & checksum field)	(reply code)(status) "22.00/0.00/0.00/0.00/0.00/22.00/7/4/0.00/0.00/1.00/0.00/22.00/0.00/0.00/0.00/0.00/0.00" (checksum)

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

	FIELD 6	Daily total	AMOUNT	Default	Daily total sum (the sum of fields 1 to 5)
	FIELD 7	Legal receipts total	AMOUNT	Default	The sum of all legal receipts during the day
	FIELD 8	Illegal receipts total	AMOUNT	Default	The sum of all illegal receipts during the day
	FIELD 9	Voids total	AMOUNT	Default	The sum of all voids during the day
	FIELD 10	Refunds total	AMOUNT	Default	The sum of all refunds during the day
	FIELD 11	Cancels total	AMOUNT	Default	The sum of all cancels during the day
	FIELD 12	CASH (Type of payment)	AMOUNT	Default	The sum of payment by cash
	FIELD 13	CARD (Type of payment)	AMOUNT	Default	The sum of payment by credit card
	FIELD 14	CREDIT (Type of payment)	AMOUNT	Default	The sum of payment by credit
	FIELD 15	Total of Amount Discounts	AMOUNT	Default	The sum of all discounts (on sales) during the day
	FIELD 16	Total of Amount Markups	AMOUNT	Default	The sum of all markups (on sales) during the day
	FIELD 17	Total of Subtotal Discounts	AMOUNT	Default	The sum of all discounts (on subtotal) during the day
	FIELD 18	Total of Subtotal Amount Markups	AMOUNT	Default	The sum of all markups (on subtotal) during the day

8.2.17. Start Read Flash Memory to Download [A]

This command is for issuing the daily closure (Z) report. If an error (51 hex 81des) occurs after the command is executed, then this means that either there is a 48 hours difference between the last and the current Z report, or that the CITIZEN CT-S601's clock is set to the wrong time (if that is the case then the machine's clock must be read with the help of the command 't'(8.2.4) and if its time reading is correct, then a new trial can be made but this time the option 2 must be put in the z report issue command)

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**A**	2 (Counting request code & checksum field)	0 (Without request code & checksum field)	"A" (checksum)

	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	6 (Counting reply code, status & checksum field)	2 (Without reply code, status & checksum field)	(reply code)(status) "0386/CCC990000011310100386000l_a.txt" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	It returns the number of issued Z.	INTEGER	1-4 digits	It returns the number of issued Z.
	FIELD 2	It returns the name of the file in which the data must be stored As _a the data and as_b the signature.			

8.2.18. Start Read line per line Flash Memory [Q]

This command is used to start to read flash memory to download files into pc.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**Q**	2 (Counting request code & checksum field)	0 (Without request code & checksum field)	"Q" (checksum)

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

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	6 (Counting reply code, status & checksum field)	2 (Without reply code, status & checksum field)	(reply code)(status) 0/0019/CCA880000050019131011138_c.txt (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Report Type	INTEGER	1	It is the type for the data to follow 0 = name of file to save into pc read next 1 = valid data to save into pc read next 2 = finish one file read next 3 = finish read flash memory
	FIELD 2	DATA to save into pc	STRING	0-500	If the type is 1 then the data must be stored in the PC (at the end of the data 0x13+0x10 must be added)

					If the type is 0, 1 or 2 read the next record If the type is 3 then the data are the electronic signature of all the data that were transmitted from the Flash Memory.
	General notes for the above 2 commands - To issue a z report (Daily sales zeroing and recording of the accumulators in the Fiscal Memory) the electronic journal must be first read and the results must be stored in the PC. - One of the files has the ending _a (e.g. name_a.txt) and contains the electronic journal's data, one of the files has the ending _b (e.g. name_b.txt) and contains the electronic signature of the receipt, one of the files has the ending _e (e.g. name_e.txt) and contains the amounts of receipt, one of the files has the ending _c (e.g. name_c.txt) and contains the _c electronic signature, one of the files has the ending _d (e.g. name_d.txt) and contains the _d electronic signature and the last file has the ending _s (e.g. name_s.txt) and contains the amounts of daily receipts.				

8.2.19. Programming of Parameters of ΑΔΗΜΕ [S]

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**S**	14 (Counting request code & checksum field)	12 (Without request code & checksum field)	"S/99/ECR99/CLERK99/1/0/0/1/0/0/1/1/1" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'S' for this command.
FIELD 2	The ECR number in the store 1-99	NUM	0-2 bytes (1-99)	The ECR number in the store (optional).
FIELD 3	ECR description	STRING	0-8 bytes	The ECR description (optional).
FIELD 4	Clerk Description	STRING	0-20 bytes	The Clerk description (optional).
FIELD 5	Departments will be printed in the Z-Report	NUM	0-1 byte	0 They will not be printed 1 They will be printed If the field is not filled, then the last record will appear.
FIELD 6	The drawer will automatically be opened at the end of a receipt.	NUM	0-1 byte	0 It won't be opened It will be opened If the field is not filled, then the last record will appear.

	FIELD 7	An illegal receipt will be printed every time the drawer is opened.	NUM	0-1 byte	0 It won't be printed (Default) 1 It will be printed If the field is not filled, then the last record will appear.
	FIELD 8	A VAT Analysis will be printed at the end of the receipt.	NUM	0-1 byte	0 An Analysis won't be printed 1 An Analysis will be printed If the field is not filled, then the last record will appear.
	FIELD 9	3 comment lines will be printed at the end of the receipt.	NUM	0-1 byte	0 They won't be printed 1 They will be printed If the field is not filled, then the last record will appear.
	FIELD 10	A graphic will be printed at the beginning of the receipt (if graphics are supported)	NUM	0-1 byte	0 No graphic will be printed 1-N The serial number of the graphic that will be printed at the beginning of the receipt. If the field is not filled, then the last record will appear.
	FIELD 11	A graphic will be printed at the end of the receipt (if graphics are supported)	NUM	0-1 byte	0 No graphic will be printed 1-N The serial number of the graphic that will be printed at the beginning of the receipt. If the field is not filled, then the last record will appear.

	FIELD 12	The paper will be automatically cut at the end of every receipt.	NUM	0-1 byte	0 The paper won’t be cut 1 The paper will be cut If the field is not filled, then the last record will appear.
	FIELD 13	The type of the LCD display that the machine supports	NUM	0-1 byte	0 MICRELEC LCD type 1 SERIAL VFD type (Epson compatible) If the field is not filled, then the last record will appear.

	TOTAL FIELD COUNT	**DATA FIELD COUNT**	**EXAMPLE REPLY**
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.20. Reading Parameters [s]

With this command the machine's parameters can be read.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**s**	2 (Counting request code & checksum field)	0 (Without request code & checksum field)	"s" (checksum)

	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	16 (Counting reply code, status & checksum field)	12 (Without reply code, status & checksum field)	(reply code)(status) "99/ECR99/CLERK99/1/0/0/1/0/0/0/1/1" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	The ECR number in the store 1-99	NUM	0-2 bytes	The ECR number in the store (optional).
	FIELD 2	ECR description	STRING	0-8 bytes	The ECR description (optional).
	FIELD 3	Clerk Description	STRING	0-20 bytes	The Clerk description (optional).
	FIELD 4	Departments will be printed in the Z-Report	NUM	0-1 byte	0 They will not be printed 1 They will be printed If the field is not filled, then the last record will appear.
	FIELD 5	The drawer will automatically will be	NUM	0-1 byte	0 It won't be opened

		opened at the end of a receipt.			1 It will be opened If the field is not filled, then the last record will appear.
	FIELD 6	An illegal receipt will be printed every time the drawer is opened.	NUM	0-1 byte	0 It won't be printed (Default) 1 It will be printed If the field is not filled, then the last record will appear.
	FIELD 7	A VAT Analysis will be printed at the end of the receipt.	NUM	0-1 byte	0 An Analysis won't be printed 1 An Analysis will be printed If the field is not filled, then the last record will appear.
	FIELD 8	3 comment lines will be printed at the end of the receipt.	NUM	0-1 byte	0 They won't be printed 1 They will be printed If the field is not filled, then the last record will appear.
	FIELD 9	A graphic will be printed at the beginning of the receipt (if graphics are supported)	NUM	0-1 byte	0 No graphic will be printed 1-N The serial number of the graphic that will be printed at the beginning of the receipt. If the field is not filled, then the last record will appear.
	FIELD 10	A graphic will be printed at the end of the receipt (if graphics are supported)	NUM	0-1 byte	0 No graphic will be printed 1-N The serial number of the graphic that will be printed at the beginning of the receipt. If the field is not filled, then the last record will appear.
	FIELD 11	The paper will be automatically cut at the end of every receipt.	NUM	0-1 byte	0 The paper won't be cut 1 The paper will be cut If the field is not filled, then the last record will appear.

	FIELD 12	The type of the LCD display that the machine supports	NUM	0-1 byte	0 MICRELEC LCD type 1 SERIAL VFD type If the field is not filled, then the last record will appear.

8.2.21. Printing string into select station [P]

This command is used to print a line to the printer.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**P**	4 (Counting request code & checksum field)	2 (Without request code & checksum field)	"P/PRINTING STRING/1" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'P' for this command.
FIELD 2	Printing string	STRING	1-30 bytes	The line to send to the printer.
FIELD 3	Font type	INTEGER	1 byte	Printing type: 1:PRNTYPE_NORMAL 30 bytes 2:PRNTYPE_DOUBLE 30 bytes 3:PRNTYPE_NORMAL | DBLWIDTH 20 bytes 4:PRNTYPE_DOUBLE | DBLWIDTH 20 bytes

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.22. Line Feed [F]

REQUEST PACKET	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
	F	3 (Counting request code & checksum field)	1 (Without request code & checksum field)	"F/1" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'F' for this command.
	FIELD 2	Line for feed	NUM	1-2 bytes	Number of lines to feed.

REPLY PACKET	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.23. Open a transaction or Close/Cancel an open transaction [O]

This command is for opening a new transaction or close/cancel any open transaction.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	O	3 (Counting request code & checksum field)	1 (Without request code & checksum field)	"O/1" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'O' for this command.
	FIELD 2	Open/ Close/ Cancel a transaction	INTEGER	Fixed, 1 digit	The type can be: 0 = Open transaction 1 = Close transaction 2 = Cancel transaction

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.24. Set VAT rates [b]

This command is used to program the VAT rates of the ECR/POS. For this command to succeed, a day must not be open.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**b**	6 (Counting request code & checksum field)	4 (Without request code & checksum field)	"b/6.5/13/23/36" (checksum)

	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.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY

REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.25. Read VAT rates [e]

This command is used to retrieve the current vat rates programmed into the ECR/POS.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**e**	2 (Counting request code & checksum field)	0 (Without request code & checksum field)	"e" (checksum)

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

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	9 (Counting reply code, status & checksum field)	5 (Without reply code, status & checksum field)	(reply code)(status) "6.50/13.00/23.00/36.00/0.00" (checksum)

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

8.2.26. Open cash in/out transaction [6]

This command is for opening a cash-in or cash-out transaction to the ECR/POS.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**6**	5 (Counting request code & checksum field)	3 (Without request code & checksum field)	"6/1/1.5/COMMENT" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be '6' for this command.
FIELD 2	Cash in / Cash out type	INTEGER	Fixed, 1 digit	The type can be: 0 = Open Cash in transaction 1 = Open Cash out transaction
FIELD 3	Cash in/Cash out Amount	AMOUNT	1-10 digits	It is the Cash in/Cash Amount that the CITIZEN CT-S601 stores as CASH
FIELD 4	Comments	STRING	0-35 chars	Comments

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.27. Open Drawer-Cut Paper [p]

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**p**	3 (Counting request code & checksum field)	1 (Without request code & checksum field)	"p/1" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'p' for this command.
	FIELD 2	Open Drawer / Paper Cut	NUM	1-2 bytes	1= Open Drawer 2= Paper Cut

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.28. Read last Z number [#]

Read last Z

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**#**	2 (Counting request code & checksum field)	0 (Without request code & checksum field)	"#" (checksum)

	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	6 (Counting reply code, status & checksum field)	2 (Without reply code, status & checksum field)	(reply code)(status) "24/6" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Last Z number	INTEGER	1-4 digits	The number of last Z.
	FIELD 2	Last receipt's number	INTEGER	1-4 digits	The number of last receipt.

8.2.29. Programming Footer [Y]

Programs the footer of 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 centered automatically.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**Y**	8 (Counting request code & checksum field)	6 (Without request code & checksum field)	"Y/1/FOOTERLINE1/2/FOOTERLINE2/1/FOOTERLINE3" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'Y' for this command.
FIELD 2	Footer line printing type	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing, 2 = Double height 3 = Double width, 4 = Double width/height When printing double width, only 24 characters of the line can be printed.
FIELD 3	Footer line text	STRING	0-48 chars	The text data for each line
FIELD 4	Footer line printing type	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing, 2 = Double height 3 = Double width, 4 = Double width/height

					When printing double width, only 24 characters of the line can be printed.
	FIELD 5	Footer line text	STRING	0-48 chars	The text data for each line
	FIELD 6	Footer line printing type	INTEGER	0-1 digits	The printing type for each header line as: 1 = Normal printing, 2 = Double height 3 = Double width, 4 = Double width/height When printing double width, only 24 characters of the line can be printed.
	FIELD 7	Footer line text	STRING	0-48 chars	The text data for each line

	TOTAL FIELD COUNT	**DATA FIELD COUNT**	**EXAMPLE REPLY**
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.30. Programming Category [K]

Programming CITIZEN CT-S601's Categories 1-20

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**K**	4 (Counting request code & checksum field)	2 (Without request code & checksum field)	"K/1/CATEGORY_01" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'K' for this command.
	FIELD 2	Category's serial number 1-20	NUM	0-2	Category's serial number 1-20
	FIELD 3	Category's description	STRING	0-35 chars	Category's description

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.31. Programming Departments [d]

Programming Departments 1-5

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**d**	5 (Counting request code & checksum field)	3 (Without request code & checksum field)	"d/1/DEPARTMENT_01/1" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'd' for this command.
FIELD 2	Department's Serial Number 1-5	NUM	1-2 digits	Department's Serial Number 1-5
FIELD 3	Department's description	STRING	1-35 chars	Department's description
FIELD 4	Category's serial number	NUM	0-2 digits	The serial number of the category, the department belongs to (1-20).

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.32. Read Sales per DEPARTMENTS [D]

Reading sales per department

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**D**	3 (Counting request code & checksum field)	1 (Without request code & checksum field)	"D/1" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'D' for this command.
	FIELD 2	Department's serial number 1-5	NUM	1-4	It is the serial number of the department, which data we want to read.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	9 (Counting reply code, status & checksum field)	5 (Without reply code, status & checksum field)	(reply code)(status) "TMHMA-01/1/2/0.00/0.000" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Department's description	STRING	1-35 chars	Department's description
	FIELD 2	VAT Code 1-5	NUM	1 digit	VAT Code 1-5
	FIELD 3	Category number 1-20	NUM	1-2 digits	Category number 1-20
	FIELD 4	Sales total	AMOUNT	Default	Sales total
	FIELD 5	Sales quantities	QTY	Default	Sales quantities

8.2.33. Read Sales per CATEGORY [k]

Reading sales per category

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**k**	3 (Counting request code & checksum field)	1 (Without request code & checksum field)	"k/1" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'k' for this command.
	FIELD 2	Category's serial number 1-20	NUM	1-4	It is the serial number of the category, which data we want to read.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	7 (Counting reply code, status & checksum field)	3 (Without reply code, status & checksum field)	(reply code)(status) "CATEGORY_01 /13.00/5.000" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Category's description	STRING	1-35 chars	Category's description
	FIELD 2	Sales total	AMOUNT	Default	Sales total
	FIELD 3	Sales quantities	QTY	Default	Sales quantities

8.2.34. Read last Z number and date time [*]

Reading of the last Z number and date time

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	*	2 (Counting request code & checksum field)	0 (Without request code & checksum field)	"*" (checksum)

	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	10 (Counting reply code, status & checksum field)	6 (Without reply code, status & checksum field)	(reply code)(status) "25/111013/140400/3/110045/134500" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Last Z number	INTEGER	1-4 digits	The number of last Z.
	FIELD 2	Last Z's issuing date	DATE6	Default (fixed 6)	The date of last Z.
	FIELD 3	Last Z's issuing time	TIME	Default (fixed 6)	The time of last Z.
	FIELD 4	Last receipt's number	INTEGER	1-4 digits	The number of last receipt.
	FIELD 5	Last receipt signature's date	DATE6	Default (fixed 6)	The date of last receipt.
	FIELD 6	Last receipt signature's time	TIME	Default (fixed 6)	The time of last receipt.

8.2.35. Read sales totals per payment [(]

Reading of the sales totals per payment

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

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be '(' for this command.
	FIELD 2	Payment number	NUM	1 byte	Payment number

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	5 (Counting reply code, status & checksum field)	1 (Without reply code, status & checksum field)	(reply code)(status) "50.00" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Payment amount	AMOUNT	Default	Total sales in each payment.

8.2.36. Read the free space of the FLASH [)]

It returns the free space of the Flash memory in blocks of 512 bytes

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**)**	2 (Counting request code & checksum field)	0 (Without request code & checksum field)	")" (checksum)

	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	5 (Counting reply code, status & checksum field)	1 (Without reply code, status & checksum field)	(reply code)(status) "115904 " (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Free memory in KB.	INTEGER	1-6 digits	The available capacity of flash memory.

8.2.37. Cancel Payments in receipt [c]

When a receipt is open, this command will force the ECR/POS firmware state to enter payment mode.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**c**	6 (Counting request code & checksum field)	4 (Without request code & checksum field)	"c/2/CREDIT CARD/DINERS-12345678/12.56" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'c' for this command.
FIELD 2	Payment type (index)	INTEGER	1digits	The payment code as follows: 1=CASH 2=CARDS 3=CREDIT These 3 codes are used only for the storing of the Payments in order for a Z Report to be issued
FIELD 3	The operation description	STRING	1-35 chars	Optional string for description of operation. If not passed, the default string will be used. (CASH, CARDS, CREDIT)
FIELD 4	The operation extra description	STRING	0-35 chars	Optional string for extra description of operation.
FIELD 5	Payment Amount	AMOUNT	1-10 digits	The amount for the payment If the amount is 0 then the receipt is closed containing the whole amount.

REPLY PACKET	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
	5 (Counting reply code, status & checksum field)	1 (Without reply code, status & checksum field)	(reply code)(status) "3.00" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Balance to be paid (AMOUNT)	AMOUNT	Default	It is the balance to be paid. If it is negative then it is change.

8.2.38. Set external serial ports for display data [[]

Defining the external Display that the sales will appear to

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**[**	5 (Counting request code & checksum field)	3 (Without request code & checksum field)	"[/1/1/2" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be '[' for this command.
FIELD 2	Display sales in an external display	INTEGER	1 digit	If the sales will be automatically displayed in an external display 1=NO 2=YES
FIELD 3	Display COM	NUM	1 digit	At the back of CITIZEN CT-S601 are 2 communication ports (DISPLAY – FM) 1 = Display Sales in the DISPLAY port 2 = Display Sales in the FM port
FIELD 4	External's display type	NUM	1 digit	The external's display type that will is plugged in the communication port 1 = LCD Micrelec 2 = VFD (Epson compatible)

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.39. Display data into external LCD or VFD [2]

Define in which external display the sales will be displayed

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**2**	7 (Counting request code & checksum field)	5 (Without request code & checksum field)	" 2/5/1/1/2/DISPLAY DATA" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be '2' for this command.
FIELD 2	The amount of time in which the message will be displayed in the external display	INTEGER	3 digits	The amount of time in which the message will be displayed in the external display (in seconds)
FIELD 3	Display COM	NUM	1 digit	At the back of CITIZEN CT-S601 are 2 communication ports (DISPLAY – FM) 0 = Display Sales in the DISPLAY port 1 = Display Sales in the FM port
FIELD 4	External's display type	NUM	1 digit	The external's display type that will is plugged in the communication port 1= LCD Micrelec 2 = VFD (Epson compatible)
FIELD 5	Display's Text line number	NUM	1 digit	In which text line of the Display the message will appear 1 = Text Line 1

					2 = Text Line 2 3 = clear display
	FIELD 6	DATA to be displayed	STRING	0-20 chars	These are the data that will appear in the Display

	TOTAL FIELD COUNT	**DATA FIELD COUNT**	**EXAMPLE REPLY**
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.40. Read any digital signature from fiscal memory [R]

Read any digital signature from fiscal memory.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**R**	3 (Counting request code & checksum field)	1 (Without request code & checksum field)	"R/" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'R' for this command.
	FIELD 2	Fiscal Memory's electronic signature number	NUM	1-4 bytes	This is the Fiscal Memory's electronic signature number which will be read. The amount of the electronic signatures stored in the fiscal memory, can be read with using the Command '#' (8.2.28).

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	9 (Counting reply code, status & checksum field)	5 (Without reply code, status & checksum field)	(reply code)(status) "00/02/02/141013/103400/C23F4A7BCCE8BDE74C4DB96EC89190E8D9836364/9B3D64857B891012324A1A8A680C06B24EB6D2F7/29" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Fiscal Memory's Electronic Signature Storing Date	DATE6	Default	It is the date in which the electronic signature was stored in the fiscal memory

	FIELD 2	Fiscal Memory's Electronic Signature Storing Time	TIME	Default	It is the time in which the electronic signature was stored in the fiscal memory
	FIELD 3	Electronic signature _c	STRING	40 chars	It is the _c signature of the daily electronic file
	FIELD 4	Electronic signature _d	STRING	40 chars	It is the _d signature of the daily electronic file
	FIELD 5	The serial Z number of the electronic signature	NUM	Default	It is the serial Z number of which the electronic signature was issued.

8.2.41. Automatic sales display in an external Display [1]

The sales are automatically displayed in an external Display which is defined from the type of the external Display that has been programmed using the field 13 of the command 'S' (8.2.19)

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**1**	2 (Counting request code & checksum field)	0 (Without request code & checksum field)	"1" (checksum)

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

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	5 (Counting reply code, status & checksum field)	1 (Without reply code, status & checksum field)	(reply code)(status) "0" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	CITIZEN CT-S601's status	NUM	Default	0 = The CITIZEN CT-S601"s sales are not displayed in an external Display 1 = The CITIZEN CT-S601's sales are automatically displayed in an external Display

8.2.42. Automatic Item's quantity printing at the end of the receipt [q]

This command is used to print or cancel the printing of the sales at the end of the receipt.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**q**	2 (Counting request code & checksum field)	0 (Without request code & checksum field)	"q" (checksum)

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

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	5 (Counting reply code, status & checksum field)	1 (Without reply code, status & checksum field)	(reply code)(status) "0" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	CITIZEN CT-S601's status	NUM	Default	0 = Do not print the quantity of the sold items 1 = Automatically prints the quantity of the sold items

8.2.43. Input of 3 comment lines to be automatically printed [m]

This command inputs 3 comment lines that will be printed at the end of the receipt

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**m**	5 (Counting request code & checksum field)	3 (Without request code & checksum field)	"m/COMMENTLINE1/COMMENTLINE2/COMMENTLINE3" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'm' for this command.
	FIELD 2	The 1st comment line to be printed	STRING	0-48 chars	The actual text that will be printed.
	FIELD 3	The 2nd comment line to be printed	STRING	0-48 chars	The actual text that will be printed.
	FIELD 4	The 3rd comment line to be printed	STRING	0-48 chars	The actual text that will be printed.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.44. Set top icons [Z]

This commands is used to set the top icon that will be printed in the receipt.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**Z**	3 (Counting request code & checksum field)	1 (Without request code & checksum field)	"Z/1" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'Z' for this command.
	FIELD 2	The top bitmap to be printed	INTEGER	0-1	The top bitmap to be printed

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.45. Set size of top and bottom icons [-]

This commands is used to set the size of the top icon that will be printed in the receipt.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**-**	3 (Counting request code & checksum field)	1 (Without request code & checksum field)	"-/1" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be '-' for this command.
	FIELD 2	The size of the top bitmap to be printed	INTEGER	Default	0=normal 1= double size and width

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.46. Read footer [{]

Reads the current (active) footer setting in the device.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**{**	2 (Counting request code & checksum field)	0 (Without request code & checksum field)	"{" (checksum)

	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	16 (Counting reply code, status & checksum field)	12 (Without reply code, status & checksum field)	(reply code)(status) "1/FOOTERLINE1/2/FOOTERLINE2/1/FOOTERLINE3/1/FOTERLINE4/1/FOOTERLINE5/1/FOOTERLINE6" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Footer line printing types	INTEGER	1	The printing type for each header line as: 1 = Normal printing, 2 = Double height 3 = Double width, 4 = Double width/height
	FIELD 2	Footer line text	STRING	0-48 chars	The text data for each line
	FIELD 3	Footer line printing types	INTEGER	1	The printing type for each header line as: 1 = Normal printing,

					2 = Double height 3 = Double width, 4 = Double width/height
	FIELD 4	Footer line text	STRING	0-48 chars	The text data for each line
	FIELD 5	Footer line printing types	INTEGER	1	The printing type for each header line as: 1 = Normal printing, 2 = Double height 3 = Double width, 4 = Double width/height
	FIELD 6	Footer line text	STRING	0-48 chars	The text data for each line
	FIELD 7	Footer line printing types	INTEGER	1	The printing type for each header line as: 1 = Normal printing, 2 = Double height 3 = Double width, 4 = Double width/height
	FIELD 8	Footer line text	STRING	0-48 chars	The text data for each line
	FIELD 9	Footer line printing types	INTEGER	1	The printing type for each header line as: 1 = Normal printing, 2 = Double height 3 = Double width, 4 = Double width/height
	FIELD 10	Footer line text	STRING	0-48 chars	The text data for each line
	FIELD 11	Footer line printing types	INTEGER	1	The printing type for each header line as: 1 = Normal printing,

					2 = Double height 3 = Double width, 4 = Double width/height
	FIELD 12	Footer line text	STRING	0-48 chars	The text data for each line

8.2.47. Set Receipt Client Card [}]

This command is used to set the size the number of client card in the receipt.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	}	3 (Counting request code & checksum fields)	1 (Without request code & checksum fields)	"}/123456789012345" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be '}' for this command.
	FIELD 2	The number of client card	STRING	1-19 chars	The number of client card for receipts.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.48. Subtotal in receipt [o]

This command is used to print subtotal in receipt.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**o**	2 (Counting request code & checksum fields)	0 (Without request code & checksum fields)	"o" (checksum)

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

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.49. Void Previous Transaction [V]

This command is used to void the previous transaction in an open receipt.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**V**	2 (Counting request code & checksum fields)	0 (Without request code & checksum fields)	"V" (checksum)

	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	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.50. Read/Print GGPS settings, Read Ethernet settings [,]

Read settings for send data to GGPS server OR read settings of Ethernet.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	,	3 (Counting request code & checksum fields)	1 (Without request code & checksum fields)	"," (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be ',' for this command.
	FIELD 2	Read or Print	INTEGER	1 DIGIT	0: Read GGPS settings 1: Print GGPS settings 2: Read Ethernet settings

Read GGPS settings:

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	8 (Counting reply code, status & checksum)	4 (Without reply code, status & checksum)	(reply code)(status) "1/http://147.102.24.100/myweb/websend.php/80/6697CF19399F2F1655AD6C5C0A80A5F51A1C91149F8E8A9B455CD2F401D738A6"(checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Active send	INTEGER	Default	0= Inactivate send data to GGPS 1= Activate send data to GGPS
	FIELD 2	GGPS Server	STRING	1-80 chars	Server to send to GGPS
	FIELD 3	GGPS Port	INTEGER	Default	Port to send data to GGPS
	FIELD 4	AES Key	STRING	0-64 digits	The AES key for send to GGPS

Print GGPS settings:			
REPLY PACKET	**TOTAL FIELD COUNT**	**DATA FIELD COUNT**	**EXAMPLE REPLY**
	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

Read Ethernet settings:			
REPLY PACKET	**TOTAL FIELD COUNT**	**DATA FIELD COUNT**	**EXAMPLE REPLY**
	12 (Counting reply code, status & checksum)	8 (Without reply code, status & checksum)	(reply code)(status) "0/192.168.0.10/192.168.0.11/9000/192.168.0.1/192.168.0.1/192.168.0.1/255.255.255.0" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	DHCP state	INTEGER	1 digit	0: Disabled 1: Enabled
	FIELD 2	IP address	STRING	up to 15 chars	Ethernet static IP address
	FIELD 3	Remote IP address	STRING	up to 15 chars	Ethernet remote IP address
	FIELD 4	Port Number	INTEGER	Default	Port Number
	FIELD 5	Gateway	STRING	up to 15 chars	Ethernet Gateway
	FIELD 6	Primary DNS	STRING	up to 15 chars	Ethernet Primary DNS
	FIELD 7	Secondary DNS	STRING	up to 15 chars	Ethernet Secondary DNS
	FIELD 8	MASK	STRING	up to 15 chars	Ethernet MASK

8.2.51. Programming GGPS settings []

Programming GGPS settings.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**]**	6 (Counting request code & checksum fields)	4 (Without request code & checksum fields)	"]/" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Request code	STRING	Fixed, 1 character	Must be ']' for this command.
	FIELD 2	Activate GGPS	INTEGER	0-1 digits	0: Disable GGPS send 1: Enable GGPS send
	FIELD 3	GGPS Server	STRING	1-80 chars	Server to send to GGPS In cases where the Server's address (URL) contains the character [/] which is the protocol's field seperator, it should be replaced with character [~].
	FIELD 4	GGPS Port	STRING	0-64 digits	Port to send data to GGPS
	FIELD 5	AES Key	STRING	0-64 digits	The AES key for send to GGPS

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code,	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

	status & checksum)		

8.2.52. Programming Parameters ADHME (new command) [B]

This command is used to set general parameters of printer.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**B**	20 (Counting request code & checksum fields)	18 (Without request code & checksum fields)	"B/1110000100100//////1/MACHINE_01//3//2///////" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'B' for this command.
FIELD 2	Flags	FLAGS	13 digits	**1st digit** **1** = Print departments on X report otherwise **0** **2nd digit** **1** = Clear PLU stats on Z report otherwise **0** **3rd digit** **1** = Print departments vat analysis on Z report otherwise **0** **4th digit** **1** = Print total quantity on receipt end otherwise **0** **5th digit** **1** = Print PLU codes in receipts otherwise **0** **6th digit** **1** = Check stock before PLU sale otherwise **0** **7th digit** **1** = print vat analysis on receipt end otherwise **0**

					8th digit **1** = Drawer open otherwise **0** **9th digit** **1** = Buzzer on Drawer open otherwise **0** **10th digit** **0** = print short date-time,/**1** = print full date-time (in receipt) **11th digit** **1** = show subtotal in display after every sale otherwise **0** **12th digit** **1** = active cutter otherwise **0** **13th digit** **0** = partial cut,/**1** = full cut
	FIELD 3	Active Clerks	INTEGER	0-2 digits	Number of Active Clerks.
	FIELD 4	Maximum item price	AMOUNT	0 or Default	A global maximum limit for item prices
	FIELD 5	Maximum sale quantity	QTY	0 or Default	A global maximum limit for sale quantities
	FIELD 6	Maximum total amount	AMOUNT	0 or Default	A global maximum limit for receipt total
	FIELD 7	Maximum daily sales amount	AMOUNT	Default	A global maximum limit for daily sales total
	FIELD 8	The ECR number in the store 1-99	NUM	0-2 bytes	The ECR number in the store (optional).
	FIELD 9	ECR description	STRING	0-25 bytes	The ECR description (optional).

	FIELD 10	Owner AFM	STRING	9 bytes	Owner’s AFM
	FIELD 11	Print Clerk/Machine	INTEGER	0-1 digits	0: Don’t print Clerk/Machine information 1: Print Machine number and description in receipt start 2: Print Clerk description in receipt start 3: Print Clerk description, Machine number and description in receipt start
	FIELD 12	N.A.	NUM	0-1 byte	
	FIELD 13	Number of drawer.	NUM	0-1 byte	Number of drawer open on receipt end. 0: Open Drawer 1 1: Open Drawer 2 2: Open Drawer 1 & 2
	FIELD 14	Serial Port 1 Type	INTEGER	0-1 digits	Command protocol or Not Used
	FIELD 15	Serial Port 2 Type	INTEGER	0-1 digits	Command protocol or Not Used
	FIELD 16	Baud Rate of Serial Port 1	INTEGER	0-1 digits	Baud Rate (0=2400, 1=9600, 2=19200, 3=28800, 4=57600, 5=115200, 6=230400)
	FIELD 17	Baud Rate of Serial Port 2	INTEGER	0-1 digits	Baud Rate (0=2400, 1=9600, 2=19200, 3=28800, 4=57600, 5=115200, 6=230400)
	FIELD 18	Protocol Lines	INTEGER	0-2 digits	Protocol buffer lines (1-80)
	FIELD 19	Protocol Time Out	INTEGER	0 - 3 digits	Protocol timeout to print buffer (100-3000)

REPLY PACKET	**TOTAL FIELD COUNT**	**DATA FIELD COUNT**	**EXAMPLE REPLY**
	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.53. Programming Advertising Message [.]

Programming the advertising message.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	.	6 (Counting request code & checksum fields)	4 (Without request code & checksum fields)	"./TEST/1/0/10" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be '.' for this command.
FIELD 2	Advertising Message	STRING	1-96 chars	Advertising Message
FIELD 3	Enable Advertising Message	INTEGER	0-1 digits	0= Disable Advertising Message 1= Enable Advertising Message
FIELD 4	Show Date/Time instead of message	INTEGER	0-1 digits	Show date/time in LCD instead of message.
FIELD 5	Time before show message	INTEGER	0-3 digits	Time before show message

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.54. Programming start receipt comments [j]

This command inputs 6 comment lines that will be printed at the start of the receipt

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**j**	8 (Counting request code & checksum fields)	6 (Without request code & checksum fields)	" j/LINE1/LINE2/LINE3/LINE4/LINE5/LINE6" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'j' for this command.
FIELD 2	The 1st comment line to be printed	STRING	0-48 chars	The 1st comment line to be printed
FIELD 3	The 2nd comment line to be printed	STRING	0-48 chars	The 2nd comment line to be printed
FIELD 4	The 3rd comment line to be printed	STRING	0-48 chars	The 3rd comment line to be printed
FIELD 5	The 4th comment line to be printed	STRING	0-48 chars	The 4th comment line to be printed
FIELD 6	The 5th comment line to be printed	STRING	0-48 chars	The 5th comment line to be printed
FIELD 7	The 6th comment line to be printed	STRING	0-48 chars	The 6th comment line to be printed

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.55. Read Advertise message [^]

Read advertising message.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	^	2 (Counting request code & checksum fields)	0 (Without request code & checksum fields)	"^" (checksum)

	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	8 (Counting reply code, status & checksum)	4 (Without reply code, status & checksum)	(reply code)(status) "1/0/10/ ICS-CITIZEN FISCAL" (checksum)

	DESCRIPTION		TYPE	LENGTH	NOTES
	FIELD 1	Active Advertising Message	INTEGER	Default	0= Inactive Advertising Message 1= Active Advertising Message
	FIELD 2	Show date-time instead of message.	INTEGER	Default	0= Inactive Show date-time 1= Active Show date-time
	FIELD 3	Time before show message.	INTEGER	0-3 digits	Time before show message.
	FIELD 4	Advertising Message	STRING	1-96 chars	Advertising Message

8.2.56. Coupon Discount [M]

This command is for issuing coupons. A transaction must be open.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**M**	7 (Counting request code & checksum fields)	5 (Without request code & checksum fields)	"M/1/1/COUPON/COUPON/1.00" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'M' for this command.
FIELD 2	The VAT Code in which the coupon according to the corresponding type will occur.	INTEGER	1 digit	The VAT Code can be 1-5 (1=A.5=E)
FIELD 3	CATEGORY CODE	INTEGER	0-2	It is the category number (1-20).
FIELD 4	Coupon Description.	STRING	0-35 chars	Description of coupon.
FIELD 5	Extended description	STRING	0-35 chars	Optional string for additional information printing of the operation. Prints one additional line below the operation printing lines.
FIELD 6	Amount of coupon.	AMOUNT	1-10 digits	The amount of the coupon.

	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.57. Print Barcode [C]

It prints a graphical Barcode. It also prints above or below the graphical barcode, the data of this barcode.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	**C**	9 (Counting request code & checksum fields)	7 (Without request code & checksum fields)	"C/10/3/2/67/12/803370678004/2" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be 'C' for this command.
FIELD 2	Height of the Graphical Barcode	NUM	1-2 bytes	This is the Height of the Graphical Barcode expressed in a number of dots It can only take values from 10-200
FIELD 3	Width of the Graphical Barcode	NUM	1 byte	This is the Width of the Graphical Barcode expressed in the density of dots It can only take quantized values from 2-6.
FIELD 4	Printing position of the Graphical Barcode data	NUM	1 byte	This is the Printing position of the barcode number 0=Data are not printed 1=TOP of the graphic 2=BOTTOM of the graphic 3=TOP&BOTTOM of graphic
FIELD 5	Graphical Barcode Printing code	NUM	2 bytes	This is the Graphical Barcode printing code. The following printing codes are supported: 65=UPC-A 66=UPC_2_SUPL

					67=JAN13 (EAN13) 68=JAN8 (EAN8) 69=CODE39 70=UPC_5_SUPL 73=CODE128
	FIELD 6	Graphical Barcode Printing LENGTH	NUM	Default	This is the Graphical Barcode printing length according to the selected printing code. The appropriate lengths according to the supported printing codes are as following UPC-A Fixed n=11 UPC_2_SUPL Fixed n=2 JAN13 (EAN13) Fixed n=12 JAN8 (EAN8) Fixed n=7 CODE39 Can be changed 1<n>127 UPC_5_SUPL Fixed n=5 CODE128 Can be changed 2<n>127
	FIELD 7	Graphical Barcode Data	NUM	Default	The Graphical Barcode data. The length must be according to the restrictions of field 6
	FIELD 8	Barcode Align	NUM	1 byte	This is the alignment of the barcode. 0=Left 1=Center 2=Right

	TOTAL FIELD COUNT	**DATA FIELD COUNT**	**EXAMPLE REPLY**
REPLY PACKET	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.58. Programming Ethernet settings [_]

Programming Ethernet settings.

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	—	10 (Counting request code & checksum fields)	8 (Without request code & checksum fields)	"_/0/8000/192.168.0.10//192.168.0.1/192.168.0.1//255.255.255.0" (checksum)

	DESCRIPTION	TYPE	LENGTH	NOTES
FIELD 1	Request code	STRING	Fixed, 1 character	Must be '_' for this command.
FIELD 2	DHCP state	INTEGER	1 digit	0: Disabled 1: Enabled
FIELD 3	Port Number	INTEGER	Default	Port Number
FIELD 4	IP address	STRING	up to 15 chars	Ethernet static IP address
FIELD 5	Remote IP address	STRING	up to 15 chars	Ethernet remote IP address
FIELD 6	Gateway	STRING	up to 15 chars	Ethernet Gateway
FIELD 7	Primary DNS	STRING	up to 15 chars	Ethernet Primary DNS
FIELD 8	Secondary DNS	STRING	up to 15 chars	Ethernet Secondary DNS
FIELD 9	MASK	STRING	up to 15 chars	Ethernet MASK

REPLY PACKET	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REPLY
	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of status and checksum.

8.2.59. Read Device Extra Status [;]

This command has no additional input output data. It is only used for getting the extra status codes from the device. Otherwise it is a NOOP (no operation).

	REQUEST CODE	TOTAL FIELD COUNT	DATA FIELD COUNT	EXAMPLE REQUEST
REQUEST PACKET	;	2 (Counting request code & checksum fields)	0 (Without request code & checksum fields)	";" (checksum)

	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	4 (Counting reply code, status & checksum)	0 (Without reply code, status & checksum)	This command's reply packet does not contain additional information; only 1 field of reply code and 2 fields of additional status and checksum.

ADDITIONAL STATUS 1

Bit 7	Bit 6	Bit 5	Bit4	Bit 3	Bit 2	Bit 1	Bit 0
RESERVED	GGPS LAST Z	COM.LINES	CASH OUT	CASHIN	IN PAYMENT	DAY OPEN	REC OPEN

Bit 0: Open Receipt in progress.
Bit 1: Day Open.
Bit 2: The receipt is in Payment Mode.
Bit 3: Cash In Receipt in progress.
Bit 4: Cash Out Receipt in progress.
Bit 5: Comments lines counter.
Bit 6: Last Z GGPS send error.
Bit 7: Reserved.

ADDITIONAL STATUS 2							
Bit 7	**Bit 6**	**Bit 5**	**Bit4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
RESERVED	**RESERVED**	**PRN TIME OUT**	**PRN DISCONNECT**	**CUTTER**	**COVER**	**PAPER END**	**RESERVED**
Bit 0: Reserved. **Bit 1**: Printer Paper End. **Bit 2**: Printer Cover Open. **Bit 3**: CUTTER ERROR. **Bit 4**: Printer Disconnect. **Bit 5**: Printer Time Out. **Bit 6**: Reserved. **Bit 7**: Reserved.							

9. Tables and miscellaneous definitions

9.1. Table 1, Reply codes / error codes

Hex	Meaning	Suggested Action
00	No errors - success	None
01	Wrong number of fields	Check the command's field count)
02	Field too long	A field is long: check it & retry
03	Field too small	A field is small: check it & retry
04	Field fixed size mismatch	A field size is wrong: check it & retry
05	Field range or type check failed	Check ranges or types in command
06	Bad request code	Correct the request code (unknown)
07	Fiscal Record Number error	The requested fiscal record number is wrong
08	Fiscal Record Type error	The requested fiscal record type is wrong
09	Printing type bad	Correct the specified printing style
0A	Cannot execute with day open	Issue a Z report to close the day
0B	RTC programming requires jumper	Short the 'clock' jumper and retry
0C	RTC date or time invalid	Check the date/time range. Also check if date is prior to a date of a fiscal record
0D	No records in fiscal period	No suggested action; the operation cannot be executed in the specified period
0E	Device is busy in another task	Wait for the device to get ready
0F	No more header records allowed	No suggested action; the header programming cannot be executed because the Fiscal memory cannot hold more records
10	Cannot execute with block open	The specified command requires no open signature block for proceeding. Close the block and retry
11	Transaction not opened	Open a transaction first
12	Sign Data Error	Error in signing the electronic data
13	Sign Error	error in signing
14	Z closure time limit	Means that 24 hours passed from the last Z closure. Issue a Z and retry

Hex	Meaning	Suggested Action
15	Z closure not found	The specified Z closure number does not exist. Pass an existing Z number
16	Z closure record bad	The requested Z record is unreadable (damaged). Device requires service
17	User browsing in progress	The user is accessing the device by manual operation. The protocol usage is suspended until the user terminates the keyboard browsing. Just wait or inform application user.
18	No more Invoice	Take a Z Report in order to continue issuing an invoice
19	Printer paper end detected	Replace the paper roll and retry
1A	Printer is offline	Printer disconnection. Service required
1B	Fiscal unit is offline	Fiscal disconnection. Service required
1C	Fatal hardware fiscal error	Mostly fiscal errors. Service required
1D	Fiscal unit is full	Need fiscal replacement. Service
1E	No Data for Signature	There are no data to be signed
1F	Signature not in range	The signature number is not in range
20	Battery fault detected	If problem persists, service required
21	Open day for signature reprint	Close the day to reprint signature
22	Reprint Signature CMOS error	Signature cannot be reprinted due to CMO error. Call service
23	Real-Time Clock needs programming (This means that the RTC has invalid Data and needs to be reprogrammed. As a consequence, service is needed).	This means that the RTC has invalid Data and needs to be reprogrammed. As a consequence, service is needed
24	JUMPERON	The Jumper are on, They must be removed for the operation to continue.
25	INVSALEOP	Error Sale type It must be S/V/R
26	DPTINDEXERR	Department’s code number out of range (1-5)
27	VATRATE	The VAT rate sent by the PC isn’t equal to the CITIZEN CT-S601’s one
28	PAYMENTINDEXERR	Payment’s code is out of range (1-3) 1=CASH, 2=CARD, 3=CREDIT

Hex	Meaning	Suggested Action
29	Printer Time Out	Connection with Printer Head cannot be established
2A	COVEROPEN	The printer tray is opened
2B	SLIP Printer Error	The slip printer is not ready
2C	Printer Head Error	The printer's Head is damaged
2D	Sensor Error	Sensor is damaged
2E	Sensor Reading Error	The Sensor cannot read
2F	NOTENDREADLEGAL	There are illegal receipts in the journal that must be read
30	NOTENDREADILEGAL	There are legal receipts in the journal that must be read
31	WRONGILEGALNUMBER	The requested illegal receipt doesn’t exist in the electronic journal
32	FLASHERROR	CARD reading problem
33	NOTFOUNDRECEIPT	The requested legal receipt doesn’t exist in the electronic journal
34	NOMOREILEGALRECEIP	There are no more receipts to be read in the CARD
35	NOTSTARTREAD	CITIZEN CT-S601 must first be told about the reading of the CARD before the CARD’s reading begins
36	NOTFINISHREADRECEIPTDATA	The CARD’s reading isn’t finished
37	NOTREADFORFOUNDRECEIPT	A record hasn’t been read
38	ENDREADFLAS	The CARD’s reading was successful
39	HWTRAYAGAN	Error reading the CARD, please try again
3A	NOTSTARTREADFLASH	CITIZEN CT-S601 must first be told about the reading of the CARD before the CARD’s reading begins
3B	NOTFOUNDOPENDAY	DAY isn’t opened and no transactions are present
3C	NOMOREINRECEIPTLINES	No more than 6 comment lines can be printed on the receipt
3D	NOTTRANSFERFLASH	The CARD’s data transfer to the PC isn’t over yet
3E	PRINTERDISCONECT	Printer is disconnected
3F	TRANSACTIONINPROGRES	Another CITIZEN CT-S601’s function is in progress
40	TRANSACTIONNOTOPEN	There is no opened receipt
41	TRANSACTIONISOPEN	There is an opened receipt
42	NOMOREVAT	No more VTA codes can be programmed in the fiscal memory

Hex	Meaning	Suggested Action
43	CASHINOPEN	Cash in is in progress
44	CASHOUTOPEN	Cash out is in progress
45	INPAYMENT	Payment is in progress
46	NOZERODM	No zero Discount/Markup is allowed
47	MAXDISCOUNTINVAT	Greater Discount than the CITIZEN CT-S601's VAT amount
48	MAXDMINTRANSTOTAL	The discount exceeds the minimum transaction amount
49	NOTEQUALDMGETSUM	VAT's allocation's totals do not match
4A	NEGATIVEVATSALES	No negative sales-transactions are allowed
4B	MUSTCLOSETRANSACTION	The receipt must be closed in order for the function to continue
4C	FLASHFULL	CARD is full, it must be read
4D	NOZEROVAT	The VAT rate cannot be 0
4E	NOSANEVATRATE	No equal VAT rates in different categories
4F	NOSALESZEROPRICE	Zero sale's price cannot occur
50	NODATAFORPRNX	There are no transactions-A X Report cannot be issued
51	WORNIGDATE	DATE/TIME Error. Call service
52	FLASSTOPWORK	CARD error. The CITIZEN CT-S601 cannot perform sales
53	NOTVALIDPLU	PLU Internal Code Error (1-200)
54	INVALIDCATEGORI	Category Code Error (1-20)
55	INVALID DPT	Department Code Error (1-5)
56	BMP Index Error	The BMP Index Number is not correct
57	Cutter Error	Turn off the CITIZEN CT-S601 and try again
58	Recover data from FLASH	The Flash CARD must be read. The machine is in an after-CMOS status
59	PAYMENT cannot be cancelled	There is no payment amount to be cancelled
5A	ZERO PAYMENT cannot be cancelled	A zero payment cannot be cancelled
5B	NOT in Payment Mode	The CITIZEN CT-S601 is not in payment mode
5C	Barcode Data Error	The Barcode Data are not valid
5D	BMP Data Error	The BMP Data are damaged
5E	Clerk index error	Wrong clerk index
5F	Clerk password error	Wrong clerk password

Hex	Meaning	Suggested Action
60	Price Error	Wrong Price
61	Invalid DM Type	Invalid Discount/Markup Type
62	DM Index	Wrong Discount/Markup Index
63	NO MORE SALES	Maximum Number of Sales in Receipt
64	Battery Error	Battery Li error
65	Clerk access problem	Access Denied for current clerk
66	Baud Rate	Wrong Baud Rate
67	Qty Error	Quantity error
68	In Ticket	After Ticket Discount
69	Inactive Ticket	The ticket is inactive
6A	DM Limit	Discount/Markup limit error
6B	Blank Description	Blank Description is not allowed
6C	Barcode Error	Error in barcode
6D	Negative Receipt Total	The receipt cannot close, negative total
6E	Client Index Error	Wrong Client index
6F	Client mot found	Wrong Client code
70	Payment no change	This Payment type cannot give change
71	Insert Payment amount	Must insert amount for payment
72	Same Header	The header is same with previous
73	In Error	There is an error and must use printer keyboard
74	Receipt Limit	Total of receipt exceed the limit
75	Day Limit	Daily total sales exceed the limit
76	Fiscal Communication Error	There is a problem with fiscal communication
77	NAND FULL	NAND memory is full
78	AFM Error	Wrong AFM
79	Empty EJ	The Electronic Journal is empty
7A	Invalid IP	Invalid IP Address
7B	Invalid Refund	Refund is not allowed
7C	Invalid Void	Void is not allowed
7D	Amount limit	Out of range amount
7E	Empty Header	The header must have at least 1 line
7F	Inactive Clerk	Clerk is inactive
80	No transactions	There are not daily transactions

Hex	Meaning	Suggested Action
81	Program AFM	You must programming AFM
82	Unformatted SD	Format SD fail, SD is unformatted
83	Time Error	Wrong Time
84	Call Technician	You must call Technician
85	Open EJ file	Cannot open EJ file
86	Write EJ file	Cannot write EJ file
87	Read EJ file	Cannot read EJ file
88	AES Code	Wrong AES Code
89	Wrong Coupon	Wrong Coupon Index/Barcode
8A	Ethernet Communication	Error in Ethernet communication
8B	Upload GGPS	Error while upload files in GGPS

9.2. Table 2, ASCII control codes [CC1]

Name	HEX	DEC	Purpose
ACK	06h	6	Aknowledge (positive)
NAK	15h	21	Not Aknowledge (negative)
STX	02h	2	Start of text
ETX	03h	3	End of Text
CAN	18h	24	Cancel
ENQ	05h	5	Enquire

9.3. Table 3, timeouts and retransmissions - minimum recommended values

Enquire Acknowledge timeout	3secs	3 retries
Packet Acknowledge timeout	3secs	3 retries
STX receive timeout	3secs	
In packet data timeout	1sec	