SPECIFICATION OF MESSAGE INTERFACE POS TO ECR

Version 1.27
21.05.2021

Revision History

Date	Author	Description
30/03/10	B.Obreshkov	Version 1.0 – Initial version of the specification
01/04/10	A.Petkov	Version 1.1 – Table with Optional Fields usage added
16.11.2010 г.	B.Obreshkov	Version 1.2 – Hold message and transmission number changed
25.11.2010 г.	B.Obreshkov	Version 1.21 – BIN added in POS reply message
1.02.2011 г.	B.Obreshkov	Version 1.22 – Loyalty ID and related messages
14.09.2011 г.	B.Obreshkov	Version 1.22 – Intercharacter timeout
06.08.2012 г.	B.Obreshkov	Version 1.22 – Error code 08
29.11.2012 г.	B.Obreshkov	Version 1.23 – Extended transaction types
3.06.2013 г.	B.Obreshkov	Version 1.24 – Card Type
01.03.2017 г 21.05.2021	B.Obreshkov B. Rangelov	Version 1.26 – Field a – Additional data Version 1.27 – Revision of used field and operation

Contents

.1 Introduction

This document specifies the communication protocol for message exchange between an EFT POS terminal and an Electronic Cash Register system.

.2 Abbreviations used

ECR	Electronic Cash Register system
POS	Point Of Service/Sale terminal
LRC	Longitudinal Redundancy Check
STX	Start of Text, ASCII control character, value 0x02
ETX	End of Text, ASCII control character, value 0x03
ACK	Acknowledge, ASCII control character, value 0x06
NAK	Negative Acknowledge, ASCII control character, value 0x15
FS	File Separator, ASCII control character, value 0x1C
sp	space, ASCII blank character, value 0x20

.3 Communication format

POS and ECR communicate via serial interface RS-232 using the following settings:

Baudrate : 2400 bps
Number of bits : 8
Parity: none
Stop bits : 1
Flow control: none

.4 Data Link Level Details

All data packets exchanged over the serial interface between ECR and POS, start with STX (0x02) and end with ETX (0x03), followed by LRC check value.

Link Level Frame:

<STX>	Message	<ETX>	<LRC>

LRC is calculated by XORing the entire data packet excluding STX and including ETX.

.5 Data Link Level Message flow

On reception of a data packet, the recipient should verify the LRC and send ACK (0x06) in case of good LRC, or NAK (0x15) in case of bad LRC. On NAK reception, the transmitting device should repeat the last data packet. In case of 3 consecutive NAKs, connection is terminated.

Message exchange of data packets with one bad LRC and repeat request is illustrated below:

(1)	POS	→	<STX> ...request... <ETX> <LRC>			
(2)				<NAK> ←	ECR	
(3)	POS	→	<STX> ...request... <ETX> <LRC>			
(4)				<ACK> ←	ECR	
(5)			<STX> ...reply ... <ETX> <LRC>		←	ECR
(6)	POS	→	<ACK>			

Retransmission Timer

After transmission of a data packet is completed, the sender shall wait for positive or negative acknowledge (ACK or NAK) for a time of 3 seconds. Upon expiry of this time, the data packet must be re-transmitted up to a maximum of three times. In case of 3 consecutive timeouts, connection is terminated.

.6 Message Format

Each message consists of a fixed length header and Optional Fields.
The Optional Fields are separated by FS (0x1C).
The first byte of each field is the Field ID followed by the actual data.

Message Format:

Header	<FS>	<Field ID>	<Field Data>	...	<FS>	<Field ID>	<Field Data>

Header data and Optional Field data are in ASCII representation. Data byte values must be in range from 0x20 to 0x7F.

Header Structure

All messages include a fixed length header.
The Header consists of several fields as specified in followed table:

Bytes	**Field Name**	**Description**
1 – 3	Version	Values from *000* to *999*. For the current specification must be *104*. It is assumed that POS may contain more than one implementation of the communication protocol, and ECR supports exactly one implementation of the protocol. If the version supported by ECR is not supported by POS terminal, connection is considered impossible and will be rejected by POS. ECR messages should contain the version currently supported. It is up to POS terminal to format the messages in a commonly supported protocol version.
4	Message Class	*0* if this is a Request Message, *1* for Response Messages.
5 – 6	Message Type	Instructs the device of the action that need to performed. Detailed description is provided in section "Message description". The set of valid values is listed below: *00* – Handshake *01* – Hold *10* – Sale transaction *11* – Cashback transaction *12* – Cash transaction *20* – Void of last txn *21* – End of Day Message type in reply, should always match the message type from request.

Header structure continued

Bytes	Field Name	Description
7 – 9	Error Code	Informs about the result of last Request Message. Valid values are: *000* – OK *001* – Protocol version not supported *002* – Message format error *003* – Transaction declined *004* – Currency not supported *005* – Unable to void *006* – Timeout to host *007* – User break *008* - Busy *088* – General error *100* – Message number mismatch *101* – Invalid message sequence *102* – Invalid card type *999* – NA – field should be ignored (used in request message)
10 – 12	Transmission Number	This is a 3 digits transmission number. It is incremented each time a new Request Message is sent. Initial value is *001*, max. value is *999*, after that it is reset to *001*. All Response Messages should contain the same transmission number as in request. If this is not the case, messages should be rejected with error code *100*. The purpose of this field is to detect duplicate transmissions and late responses. The message sequence itself can be ignored . It is not mandatory next message to have the current transmission number incremented by 1. **Note:** The Transmission Number is **NOT** incremented during repeated transmissions at Data Link Level.

Optional data fields

Field ID	Field Name	Description
B	**Amount**	Variable length from 1 to 16 bytes. Transaction amount in cents, no decimal point. It is assumed currency exponent is 2. Used in Sale and Cashback transactions.
C	**Secondary amount**	Variable length from 1 to 16 bytes. Cashback amount in cents, no decimal point. It is assumed currency exponent is 2. Used in Cashback transactions.
F	**Approval code**	Fixed length of 8 bytes. This field is left-justified and space (0x20) filled. Present in Response Messages if the transaction is approved. Also used in Void transactions.
T	**Currency code**	Fixed length of 3 bytes. Numeric currency code, e.g. '*975*' for Bulgarian Lev Should be present in Request Messages from ECR if fields *B* or *C* are present.
Q	**Terminal ID**	Fixed length of 16 bytes. This field is left-justified and space (0x20) filled. Present in Response Messages if transaction is approved. Optional for declined transactions. Also used in Void transactions.
K	**Hold time**	Fixed length of 6 bytes. Right justified. Leading zeros. Hold time, in seconds Used only in Hold message.

M	**System ID**	Fixed length of 8 bytes. Left justified and space (0x20) filled. This field contains the customer System ID. POS expects distinct System IDs for test and for production environment. Used only in Handshake message.
a	**Additional data**	Variable length from 1 to 80 bytes. Present in request message. Data inside this filed is tagged. Format is 3 bytes tag ,2 bytes length and data. Tags: INV – Invoice number Example: INV03123 – invoice number 123

Optional data fields usage

Followed table specifies the requirements to usage of Optional data fields per message type.
The "Field ID" columns in the table have following meaning:

- 'M' – Mandatory , field must be present,
- 'C' – conditional, field must be present if transaction was approved.

Field ID: **Message**	***B*** **Amount**	***C*** **Sec. amount**	***F*** **Approval code**	***T*** **Currency code**	***Q*** **Terminal ID**	***K*** **Hold time**	***M*** **System ID**
Handshake Request	-	-	-	-	-	-	**M**
Handshake Response	-	-	-	-	-	-	-
Hold Request	-	-	-	-	-	**M**	-
Hold Response	-	-	-	-	-	-	-
Sale Request	**M**	-	-	**M**	-	-	-
Sale Response	**M**	-	**C**	-	**C**	-	-
Cashback Request	**M**	**M**	-	**M**	-	-	-
Cashback Response	**M**	**M**	**C**	-	**C**	-	-
Cash Request	**M**	-	-	**M**	-	-	-
Cash Response	**M**	-	**C**	-	**C**	-	-
Void Last Request	-	-	**M**	-	**M**	-	-
Void Last Response	-	-	-	-	-	-	-
End of Day Request	-	-	-	-	-	-	-
End of Day Response	-	-	-	-	-	-	-

.7 Message Description

Handshake Request/Response

Handshake is initiated by ECR. It is used to check connection between ECR and POS. Based on Handshake message the POS will determine the actual protocol version.

Example of ECR request:

104000999YYY<FS>M99999999

where *YYY* is the current transmission number to be set as initial value in POS, *99999999* is the System ID.

Example of POS reply:

104100ZZZYYY

where *ZZZ* is *000* in case protocol version is supported or *001* in case protocol not supported.

Hold Request/Response

Hold is always initiated by POS. It is used to inform ECR that additional time is required to process the pending ECR's request. Any errors related with ECR communication (timeout, bad reply etc.) will be ignored by POS. Hold message failure should not lead to transaction termination on POS side.

Example of POS request:

104001999123<FS>K0000XX

where *XX* is the hold time in seconds

Example of ECR reply:

104101YYY123

where *YYY* is the response code (*000* – hold accepted).

Sale transaction

The sale transaction is initiated by ECR. The amount in request must match the amount in reply. If currency code in ECR request does not match the currency code configured in terminal, the message will be rejected with code *004*. If transaction is not allowed on terminal, the message will be rejected with code 088. If transaction is approved (error code in message header equals to '*000*'), approval code will be passed to ECR in field **F**. This field, together with field **Q,** may be used later to void the transaction.

Example of ECR request:

104010999124<FS>B<Amount>*<FS>T*<Currency code>

Example of POS reply (approval):

104110000124<FS>B<Amount>*<FS>F123456<sp><sp><FS>QP0010001<sp><sp><sp><sp><sp><sp><FS>q123456*

Cashback transaction

This is a Sale transaction with cashback amount. It is initiated by ECR. The amounts from request should match the amounts in reply. If currency code in ECR request does not match the currency code configured in terminal, the message will be rejected with code *004*. If transaction is not allowed on terminal, the message will be rejected with code 088. If transaction is approved (error code in message header equals to '*000*'), approval code will be passed to ECR in field **F**. This field, together with field **Q,** may be used later to void the transaction.

Example of ECR request:

104011999125<FS>B<Amount>*<FS>C*<Cashback Amount>*<FS>T975*

Example of POS reply (approval):

104111000125<FS>B<Amount>*<FS>C*<Cashback Amount>*<FS>F123456<sp><sp>*
<FS>QP0010001<sp><sp><sp><sp><sp><sp><FS>q123456

Cash

Same as Sale transaction. The only difference is Message type.

Void of last transaction

Void is always initiated by ECR. Only last, initiated by ECR and approved transaction can be voided. ECR must provide Terminal ID and approval number of the transaction to be voided. In case last transaction is not initiated by ECR, or business day was already closed, or in case the terminal ID or approval code do not match, then the request will be rejected with code *005*.

Example of ECR request:

104020999126<FS>F123456<sp><sp><FS>QP0010001<sp><sp><sp><sp><sp><sp>

Example of POS reply:

104120000126

End of day

ECR send this command in order to force the POS to end business day and upload all transactions to authorization host. After the End of Day, the terminal can perform some automatic operations and restart few times. While these operations are in progress, the terminal will not send responses to ECR.

Example of ECR request:

104021999126

Example of POS reply:

104121000126

.8 Transaction flow

Message exchange during the business day is illustrated below:

		<Start of day>		
		Handshake request	←	ECR
POS	→	Handshake reply		
		<Start of transaction>		
		Sale transaction request	←	ECR
POS	→	Hold request (optional)		
		Hold reply	←	ECR
POS	→	Hold request (optional)		
		Hold reply	←	ECR
		. . .		
POS	→	Sale transaction reply		
		<End of transaction>		
		<Start of void>		
		Void request	←	ECR
POS	→	Hold request (optional)		
		Hold reply	←	ECR
		. . .		
POS	→	Void reply		
		<End of void>		

Notes:

- ECR sends a handshake request before first transaction request (i.e on power up).
- More than one "hold" message can be send, depending on communication retries to host system.

Timeout

This is a ‘no reply’ timer which should be pre-set to 150 seconds. It is used to detect the lack of an application level response and to detect irrecoverable errors at link level. It is started when an application level request message has been sent and an application level response is expected, and is reset when this is received. This timer is extended by Hold requests.

Error handling

This protocol specification does not support reversal and confirmation messages between POS and ECR.
In case of lost communication between POS and ECR, and after a transaction is approved on POS side, but no reply message was sent to ECR, then manual intervention is required. The operator has followed options to resolve the issue:

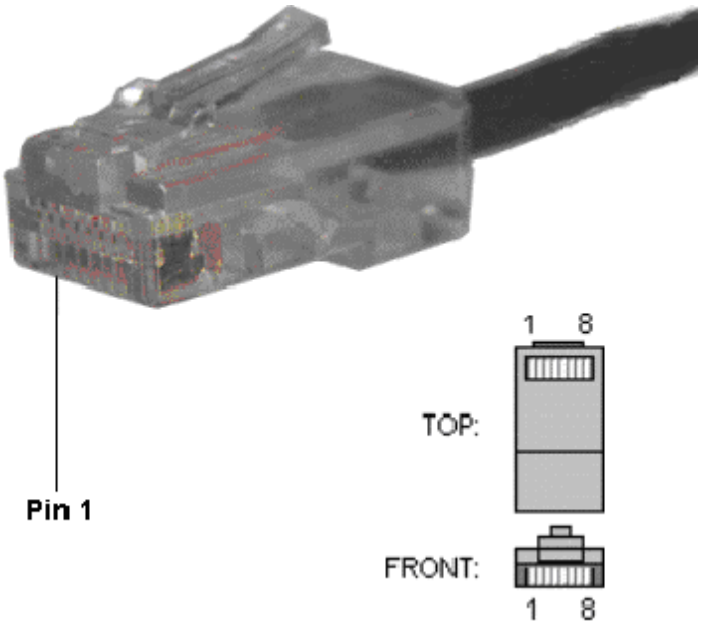
A) The approved POS transaction to be voided on POS terminal by the operator.
B) The ECR to provide an option to the operator for manual acceptance of the amount, as authorized by POS.

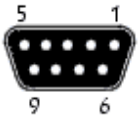
.9 Recommended Parameters

It is recommended ECR to support the parameters listed below in a way that allows update of these parameters at the time of installation:

- RS-232 Baudrate (default 2400 bps)
- RS-232 Number of bits (default 8)
- RS-232 Parity (default NO)
- RS-232 Stop bits (default 1)
- RS-232 Flow control (default none)
- Data Link Level Retransmission Timer (default 3 sec.)
- Data Link Level Max Retransmissions (default 3 times)
- Intercharacter timeout (default 50 ms)
- Application Level Timeout (default 150 sec.)
- System ID – ask bank for it’s ID

RJ 45





D-sub 9 Female

RJ 45	D-sub 9 Female
PIN 6 Transmit Data	PIN 2 Receive Data
PIN 5 Receive Data	PIN 3 Transmit Data
PIN 4 Signal Ground	PIN 5 Signal Ground