SmartPay Software

SmartPay ECR Link

Protocol specification

10/18/2023

Revision

Version	Date	Author	Description
1.0	10/25/2018	Marius Constantinescu	New release
1.1	01/25/2019	Marius Constantinescu	Add preauthorization, sale completion and read data from fleet cards
1.2	03/11/2019	Marius Constantinescu	Updates
1.3	01/25/2021	Marius Constantinescu	Add cash advance, lost connection handling, USB driver installation
1.4	03/14/2022	Marius Constantinescu	Add cancel command for sale enhanced on unattended terminals
1.5	10/27/2022	Marius Constantinescu	Add new command for card token
1.6	7/12/2023	Tiberius Staicu	Add Bluetooth/Ble interface specification
1.7	26/07/2023	Ghilinta Daniel	Add void preauthorization and update preauthorization response tags
1.8	18/10/2023	Ghilinta Daniel	Void Preauthorization TAG fixed.

Contents

1. Overview

1.1. Scope

This document describes ECR protocol between EFT-POS and master devices.

1.2. Audience

This document is for technical departments.

1.3. Organization

This document is organized as follows:

Chapter	Description
Chapter 1 – Overview	Provides an overview of this document and elements used in this document
Chapter 2 - Integration overview	Provides an overview of integration process
Chapter 3 – Protocol description	Provides an overview of the ECR protocol
Chapter 4 – Protocol commands	Provides details about the commands
Chapter 5 – Message flow	Provides details about the message flow
Appendix 1 – Communication parameters	Provide serial communication parameters
Appendix 2 – CRC algorithm	Provides CRC algorithm
Appendix 3 - Lost connection handling	Provides algorithm to handle desynchronization during payment operations
Appendix 4 - Install Windows Driver for Ingenico terminals	Provides some examples for different payment transactions
Appendix 5 - Examples	Provides some examples for different payment transactions

1.4. References Document / Location

Reference	Document/Location

1.5. Abbreviations and acronyms

This section provides reference to conventions, abbreviations and acronyms used in this manual.

The various abbreviations used throughout this manual are listed in the table below.

Abbreviation	Description
ECR	Electronic Cash Register
POS	Point of Sale
AT	Attended POS
CAT	Unattended POS
CRC	Cyclic Redundancy Check

Abbreviation	Description
STAN	System Trace Audit Number
RRN	Return Reference Number
PAN	Personal Account Number

The various acronyms used throughout this manual are listed in the table below.

Acronym	Description

2. Integration overview

The protocol commands are divided into payment operations and management operations. Payment operations are used for financial transactions and management operations are used for non-financial transactions.

2.1. Payment transaction

The cash register interface offers the following commands as part of payment operations.

Type	Description	AT	CAT
Sale	Transfer an amount from the cardholder's account to the terminal owner's account (merchant) as payment for goods and services.	Yes	Yes
Cash advance	Transfer an amount from the cardholder's account to the terminal owner's account (bank) as a cash withdrawal for goods and services.	Yes	No
Preauthorization	Locks an amount from the cardholder's account for a limited time. The time interval depends on the card issuer. Completion of the sale is required to transfer the amount from the cardholder's account to the terminal owner account (merchant).	Yes	Yes
Sale Completion	Completes the pre-authorization transaction. It means that the initially blocked amount is transferred from the cardholder's account to the account of the terminal owner (merchant). The completion can be done on a partial or total amount of pre-authorization.	Yes	Yes
Void	Void is used to cancel a previously approved sale or sale completion transaction. The operation is allowed until the first settlement.	Yes	Yes
Void preauthorization	Void preauthorization is used to cancel a previously approved pre-authorization. The operation is allowed until the first settlement.	Yes	Yes
Send offline transactions	Sends offline transactions locally stored in terminal to authorization host. This is used mainly by unattended terminals.	Yes	Yes
Settlement	Settlement is used to conclude the accounting period and check if the balance transaction state between terminal and bank. The command also clean transaction from terminal journal.	Yes	Yes
Get cash back service allowed and limit	Checks if card support sale with cash back and return maximum amount accepted as cash back.	Yes	No

2.2 Management transaction

The cash register interface offers the following commands as part of management operations.

Type	Description	AT	CAT
Get Information	Checks if terminal is available and retrieve POS hardware version and software module version.	Yes	Yes
Report Initialization	Retrieves number and total amount of financial transactions from terminal internally journals.	Yes	Yes
Report Record	Retrieves financial transactions from terminal internal journals. This is used mainly for accounting re-synchronization.	Yes	Yes
Remove Card	Check if card is inserted into terminal.	Yes	Yes
Read Fleet Card	Reads the non-banking chip cards. This is used mainly for petrol fleet cards.	Yes	Yes

2.3. Attended requirements (AT)

Segment	Mandatory	Optional
Retail store	Sale Void Settlement	Get information. Get cash back service allowed and limit Send offline transactions Report Initialization Report Record
Hotel or car rental	Sale Preauthorization Sale completion Void Settlement	Get information Send offline transactions Report Initialization Report Record
Bank branch	Cash advance Void Settlement	Get information Report Initialization Report Record

The supported terminals for the attended environment are: Ingenico iCT series, Desk series, iPP series and Lane series.

2.4. Unattended requirements (CAT)

Segment	Mandatory	Optional
Vending machine	Sale Void Settlement	Get information Send offline transactions Report Initialization Report Record

Segment	Mandatory	Optional
Ticket vending	Sale Void Settlement	Get information Send offline transactions Report Initialization Report Record
Parking station	Sale Preauthorization Sale completion Void Settlement	Get information Send offline transactions Report Initialization Report Record Read Fleet Card
Fuel station	Preauthorization Sale completion Void Settlement	Get information Send offline transactions Report Initialization Report Record Read Fleet Card

The supported terminals for the un-attended environment are: Ingenico iUN series and Self series.

2.5. Benefits

The integration benefits are:

- Fast transactions processing.
- Avoid human errors.

2.6. Risks

The integration risks are:

- Possibility to have different accounting balances between ECR and POS systems.
- Possibility to hold operator or customer due ECR and POS system de-synchronization.

To avoid the risks, we recommend to you to implement the handling of lost connection errors described in Appendix 3.

3. Protocol description

The structure of all messages is described below.

Data							
STX	Length	Tag	Len	Data	...	ETX	CRC
1	2	2	1	Variable	...	1	2
1	2	Variable				1	2

Packet description:

Field	Length	Format	Description
STX	1	HEX	Start character.
Length	2	HEX	Length in hexadecimal (..0xFFFF) (MSB, LSB) of the entire data.
Tag	2	HEX	Tag identifier.
Len	1	HEX	Tag length (..0xFF).
Data	..255	HEX	Tag data.
ETX	1	HEX	Stop character.
CRC	2	HEX	CRC16 IBM (MSB, LSB) of the entire data for request messages. CRC16 IBM (LSB, MSB) of the entire data for response messages.

Following tags are currently implemented:

Tag	Length	Format	Description
0xA000	1	HEX	Command tag from ECR to terminal: 01 – Get POS information. 02 – Perform sale by card. 03 – Perform settlement. 04 – Get report initialization. 05 – Get report record. 06 – Perform void. 07 – Get cash back service allowed and limit. 08 – Send offline transactions. 09 – Remove card. 10 – Perform void preauthorization. 0C – Perform preauthorization. 0D – Perform sale completion. 0E – Read card fleet data. 17 – Perform cash advance. 18 – Perform get card token

Tag	Length	Format	Description
0xA100	1	HEX	Response tag (Error code from terminal): 00 – Success. 01 – Generic error. 02 – Communication to Cards Host error. 03 – Out of range (transaction missing in current batch). 04 – Invalid input parameters. 05 – Invalid POS terminal parameters. 06 – Missing PIN encryption key. 07 – Cash back service allowed. 08 – Cash back service not allowed. 09 – Cancel key pressed (cancel by user).
0xA001	12	ASCII	Amount.
0xA002	3	ASCII	Currency name in ISO 4217 format (see more).
0xA003	3	ASCII	Currency code in ISO 4217 format (see more).
0xA004	3	ASCII	Transaction index.
0xA005	..255	ASCII	Additional receipt information (will be printed on receipt).
0xA006	6	ASCII	STAN (System Trace Audit Number) / Receipt Number used to identify transaction from batch.
0xA007	12	ASCII	Cash back amount.
0xA008	..25	ASCII	Unique transaction ID (received from ECR).
0xA009	1	HEX	Command flags: 0x01 – Printer status needed.
0xA012	10	ASCII	Unique transaction code (on request).
0xA101	8	ASCII	Software version in format `N.NN.NNN`.
0xA102	..255	ASCII	Hardware version / serial number.
0xA103	8	ASCII	Terminal ID.
0xA104	15	ASCII	Merchant ID.
0xA105	14	ASCII	Payment date in format `YYYYMMDDHHMISS`.
0xA106	12	ASCII	Approved amount.
0xA107	3	ASCII	Response code returned by card host.
0xA108	..255	ASCII	Response text returned by card host.
0xA109	6	ASCII	STAN (System Trace Audit Number) / Receipt Number.
0xA10A	12	ASCII	RRN (Return Reference Number).
0xA10B	6	ASCII	Authorization code.
0xA10C	..19	ASCII	PAN (Primary Account Number).
0xA10D	..40	ASCII	Card holder name.
0xA10E	3	ASCII	Transaction count.
0xA10F	6	ASCII	Batch number.
0xA110	12	ASCII	Transactions total amount.
0xA111	4	HEX	Printer status (printer is connected on serial port).

Tag	Length	Format	Description
0xA112	1	HEX	Transaction type: 0 – Sale. 1 – Voided sale. 2 – Void.
0xA113	..16	ASCII	EMV application label.
0xA114	..32	ASCII	EMV application ID.
0xA115	12	ASCII	Cash back limit amount.
0xA116	1	HEX	Transaction flags: 0x01 – Chip card used. 0x02 – Contactless card used. 0x04 – Magnetic Stripe card used. 0x08 – Signature requested. 0x10 – PIN was introduced. 0x20 – Offline transaction.
0xA117	..25	ASCII	Unique transaction ID (received from ECR and sent back by POS).
0xA119	12	ASCII	Available offline spending amount.
0xA11A	..256	HEX	Card Fleet Data
0xA11B	3	ASCII	Currency name in ISO 4217 format (see more).
0xA11C	3	ASCII	Currency code in ISO 4217 format (see more).
0xA128	10	ASCII	Unique transaction code (on response).
0xA150	64	ASCII	Card token

3.1. Bluetooth protocol (Android devices only)

The Bluetooth protocol is designed as a wrapper over the wired ECR protocol. The Bluetooth protocol can function in 2 ways

- Using Classical Bluetooth RFCOMM protocol which emulates a RS-232 serial port.
- Using Bluetooth Low Energy, in which case the communication is split between 3 GATT characteristics (Status, Downlink and Uplink).

3.1.2 Classical Bluetooth (Android devices only)

In order to use Classical Bluetooth the Android device will act as a DTE (Data Terminal Equipment) and the ECR device will act as a DCE (Data Communications Equipment).
The ECR device must provide an incoming COM port via Bluetooth RFCOMM that is opened and waiting for data before the Android device can connect.

In order to enable the Classical Bluetooth the devices must first be paired.

After pairing the ECR device must open a COM port with the parameters found in Appendix 1.

The Android device can now enable the classical BLE by going into the launcher>settings >Bluetooth ECR and selecting the according device from the list. If the connection is successful the device turns green and the status changes to "Connected".

The device uses the standard protocol for communication described for wired devices without any other modifications.

3.1.3 Bluetooth Low Energy (Android devices only)

The Bluetooth Low Energy ECR mode starts the Bluetooth GATT Server on the Android Device. The device can be connected to by using any BLE capable device.

In order to fit all the required ECR data into the GATT characteristic the MTU must be set to 512 bytes.

For communication purposes the following GATT characteristics are used:

ECR Service UUID:	00000000-8766-4d24-b978-99bed57a530d
ECR Status UUID:	00000001-0000-1000-8000-00805f9b34fb
ECR Uplink UUID:	00000002-0000-1000-8000-00805f9b34fb
ECR Downlink UUID:	00000005-0000-1000-8000-00805f9b34fb

The ECR Status UUID GATT has the following permissions: READ; NOTIFY

The ECR Uplink UUID GATT has the following permissions: WRITE

The ECR Downlink UUID GATT has the following permissions: READ; NOTIFY.

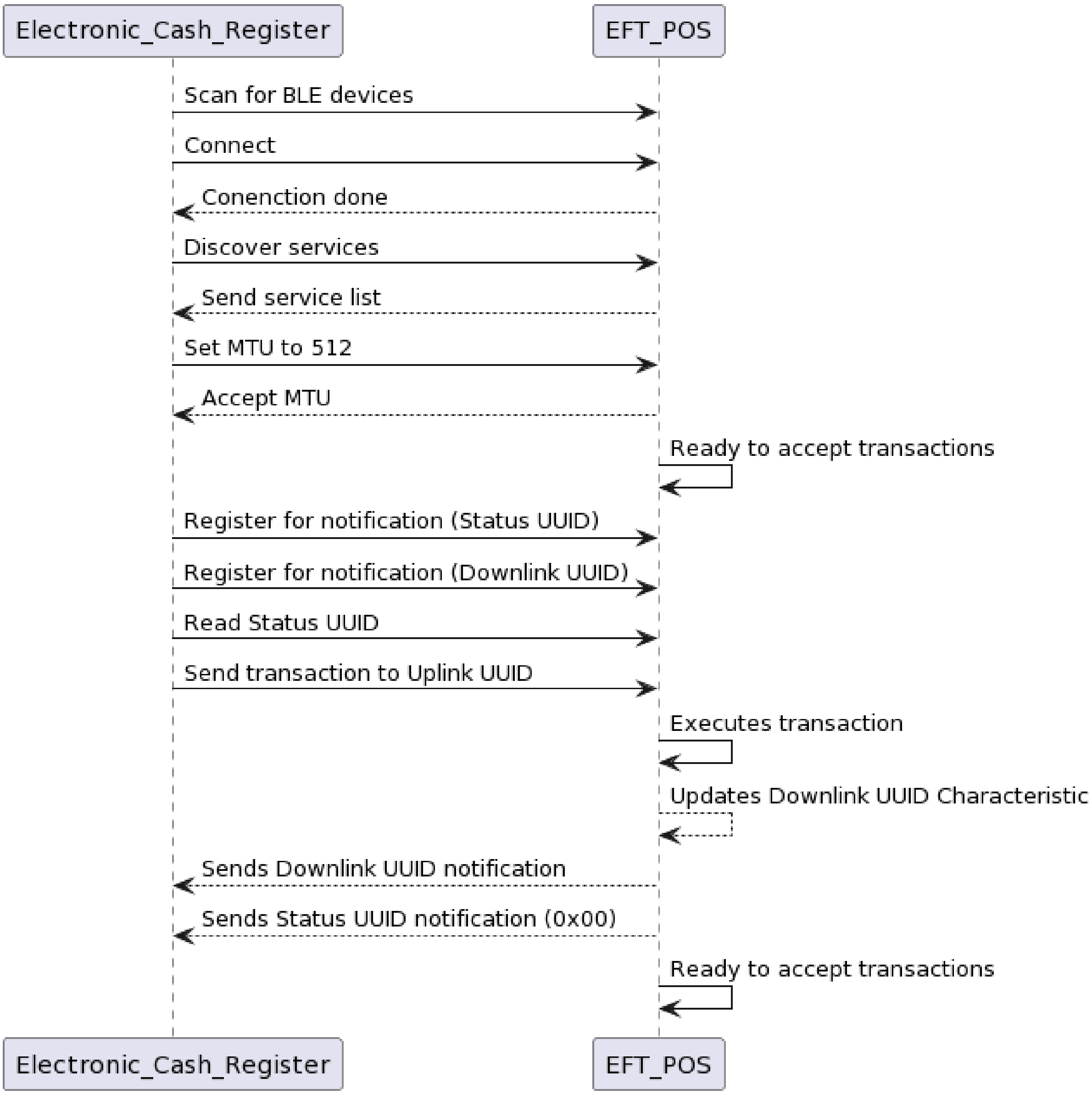
The standard (Wired) ECR protocol described in this document is extended as following:

In order to indicate that the POS device is busy with another operation the ECR Status can be 0 (indicating that the POS device is free to accept requests) or 1 (indicating that the POS device is busy with other requests).

The Uplink GATT will receive requests from the ECR device, in the same format as described in the document.

The Downlink GATT will respond to the requests sent to the Uplink GATT using the same format as described in the document, with no modifications.

The following flow represents the suggested application flow:


Electronic_Cash_Register
EFT_POS
Scan for BLE devices
Connect
Conenction done
Discover services
Send service list
Set MTU to 512
Accept MTU
Ready to accept transactions
Register for notification (Status UUID)
Register for notification (Downlink UUID)
Read Status UUID
Send transaction to Uplink UUID
Executes transaction
Updates Downlink UUID Characteristic
Sends Downlink UUID notification
Sends Status UUID notification (0x00)
Ready to accept transactions
Electronic_Cash_Register
EFT_POS

4. Operations

4.1. POS Get Information

This function should be used to retrieve POS hardware version and software module version.

Input parameters:

Tag	Length	Format	Required	Description
0xA000	1	HEX	M	Command tag: 01 – Get POS information.
0xA009	1	HEX	O	Command flags: 0x01 – Printer status needed.

Output parameters:

Tag	Length	Format	Required	Description
0xA100	1	HEX	M	Response tag: 00 – Success. 01 – Generic error.
0xA101	8	ASCII	C1	Software version in format `N.NN.NNN`.
0xA102	..255	ASCII	C1	Hardware version / serial number.
0xA111	4	HEX	C1	Printer status.

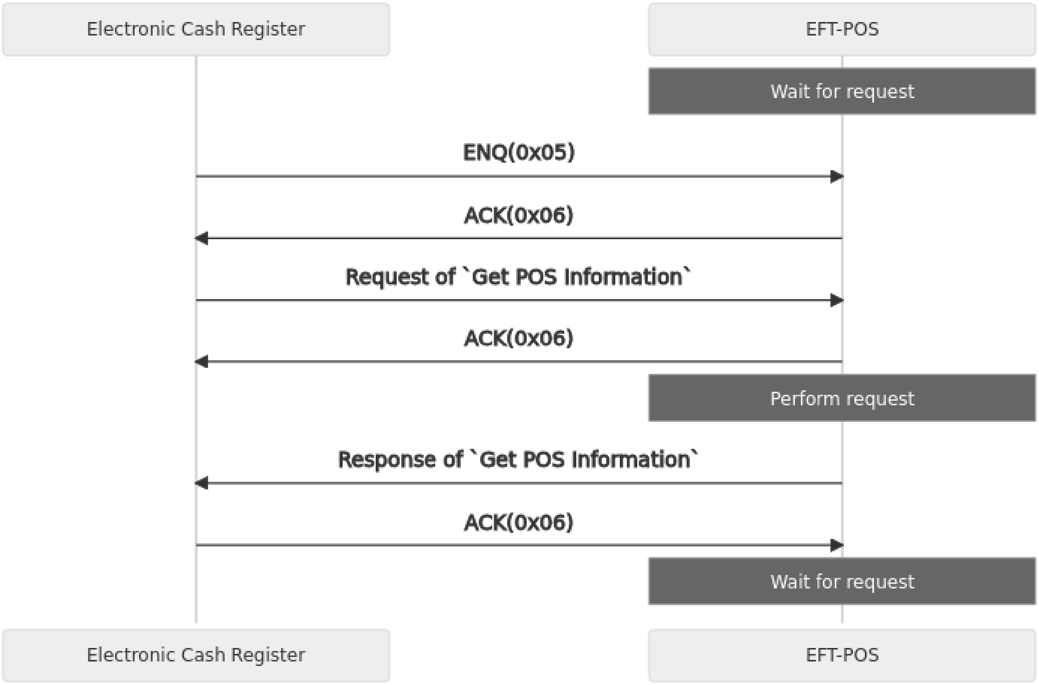
*__Require description:__

- M – mandatory
- O – optional
- C1 – mandatory when tag `0xA100` contains `00` – Success value.

Example:

Flow

The operation flow is described in the following diagram:



Data

The data exchanged is described below:

```
SEND 1 Byte
05
RECEIVE 1 Byte
06
SEND 10 Bytes
02 00 04 a0 00 01 01 03 06 35
RECEIVE 1 Byte
06
RECEIVE 63 Bytes
02 00 39 a1 00 01 00 a1 01 08 31 2e 30 33 2e 30
38 31 a1 02 20 30 30 30 30 31 38 36 39 30 30 30
30 3b 32 30 30 30 31 34 35 35 20 20 20 20 20 20
20 20 20 20 20 a1 11 04 00 00 00 00 03 60 eb
SEND 1 Byte
06
```

4.2. POS Sale by Card

This function should be used to perform sale by card transaction.

Input parameters:

Tag	Length	Format	Required	Description
0xA000	1	HEX	M	Command tag: 02 – Perform sale by card.
0xA001	12	ASCII	M	Amount.
0xA002	3	ASCII	M	Currency name.
0xA003	3	ASCII	M	Currency code.
0xA007	12	ASCII	O	Cash back amount.
0xA008	..25	ASCII	O	Unique transaction ID (if is the same ID like for POS Get Cash Back Service Allowed the terminal should not ask the card again).

Output parameters:

Tag	Length	Format	Required	Description
0xA100	1	HEX	M	Response tag: 00 – Success. 01 – Generic error. 02 – Connection to Cards Host error. 04 – Invalid input parameters. 05 – Invalid POS terminal parameters. 06 – Missing PIN encryption key. 09 – Cancel key pressed (cancel by user).
0xA103	8	ASCII	M	Terminal ID.
0xA104	15	ASCII	M	Merchant ID.
0xA105	14	ASCII	M	Payment date in format *YYYYMMDDHHMISS*.
0xA106	12	ASCII	M	Approved amount.
0xA107	2	ASCII	C1	Response code returned by card host.
0xA108	..255	ASCII	C1	Response text returned by card host.
0xA109	6	ASCII	C1	STAN (System Trace Audit Number) / Receipt Number.
0xA10A	12	ASCII	C1	RRN (Return Reference Number).
0xA10B	6	ASCII	C2	Authorization code.
0xA10C	..19	ASCII	C1	PAN (Primary Account Number).
0xA10D	..40	ASCII	C1	Card holder name.
0xA10F	6	ASCII	C1	Batch number.
0xA113	..16	ASCII	C3	EMV application label.
0xA114	..32	ASCII	C3	EMV application ID.

Tag	Length	Format	Required	Description
0xA116	1	HEX	O	Transaction flags: 0x01 – Chip card used. 0x02 – Contactless card used. 0x04 – Magnetic Stripe card used. 0x08 – Signature requested. 0x10 – PIN was introduced. 0x20 – Offline transaction.
0xA117	..25	ASCII	C4	Unique transaction ID (from ECR).
0xA119	12	ASCII	O	Available offline spending amount.

***Require description:**

- M – mandatory.
- O – optional.
- C1 – mandatory when tag `0xA100` contains `00` – Success value.
- C2 – mandatory for approved sale by card transaction.
- C3 – mandatory when EFT-POS perform chip-based transaction.
- C4 – mandatory when is send in request.

Notes:

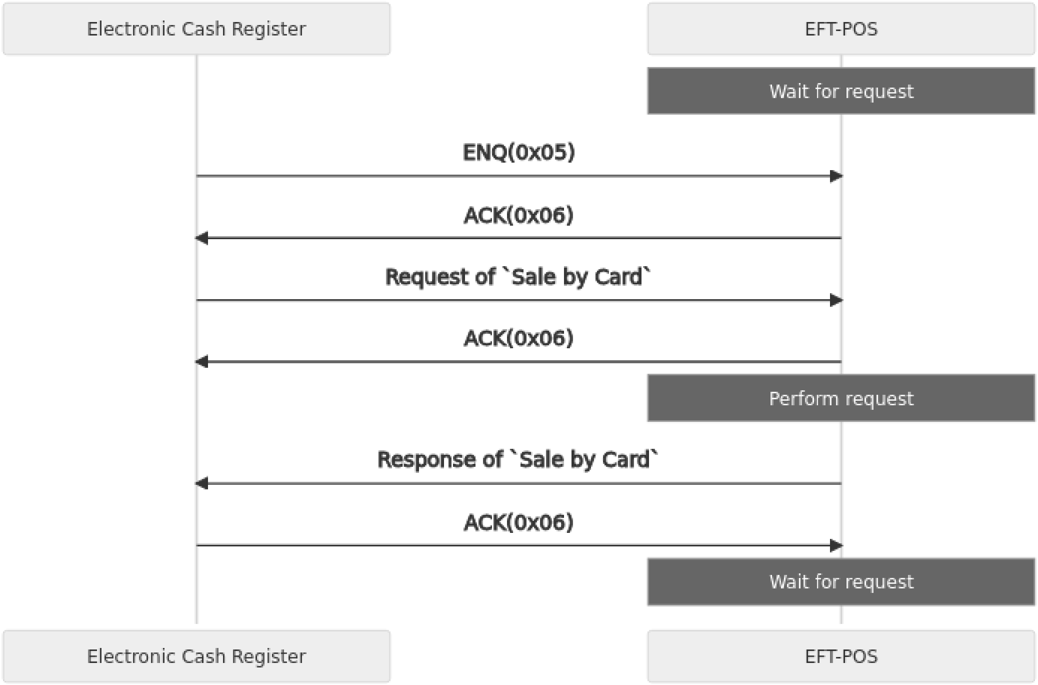
The sale by card transaction is approved (products or services are paid) only if tag `0xA100` contains value `00` and tag `0xA107` contains following values:

- `00` – approved by bank.
- `Y1` – approved offline.
- `Y3` – approved offline when cannot connect to the bank.

Example:

Flow

The operation flow is described in the following diagram:



Data

The data exchanged is described below:

```
SEND 1 Byte
05
RECEIVE 1 Byte
06
SEND 37 Bytes
02 00 1f a0 00 01 02 a0 01 0c 30 30 30 30 30 30
30 30 37 30 30 30 a0 02 03 52 4f 4e a0 03 03 39
34 36 03 31 08
RECEIVE 1 Byte
06
RECEIVE 193 Bytes
02 00 bb a1 00 01 00 a1 03 08 48 37 34 32 37 38
39 39 a1 04 0c 33 48 37 34 32 30 37 38 30 30 30
30 a1 05 0e 32 30 31 38 30 34 30 31 31 38 35 36
35 33 a1 06 0c 30 30 30 30 30 30 30 30 37 30 30
30 a1 07 02 30 30 a1 08 10 54 52 41 4e 5a 2e 20
41 43 43 45 50 54 41 54 41 a1 09 06 30 30 36 35
```

```
34 30 a1 0f 06 30 30 30 36 33 38 a1 0a 0c 30 31
39 34 36 38 39 36 31 33 39 36 a1 0b 06 35 30 33
36 30 37 a1 0c 06 2a 2a 37 30 34 37 a1 0d 0a 43
4c 49 45 4e 54 20 49 4e 47 a1 13 0d 56 69 73 61
20 45 6c 65 63 74 72 6f 6e a1 14 0e 41 30 30 30
30 30 30 30 30 33 31 30 31 30 a1 16 01 11 03 46
c5
SEND 1 Byte
06
```

4.3. POS Cash advance

This function should be used to perform cash advance transactions. This transaction is used only by bank branches.

Input parameters:

Tag	Length	Format	Required	Description
0xA000	1	HEX	M	Command tag: 17 – Perform cash advance.
0xA001	12	ASCII	M	Amount.
0xA002	3	ASCII	M	Currency name.
0xA003	3	ASCII	M	Currency code.
0xA008	..25	ASCII	O	Unique transaction ID.

Output parameters:

Tag	Length	Format	Required	Description
0xA100	1	HEX	M	Response tag: 00 – Success. 01 – Generic error. 02 – Communication to Cards Host error. 04 – Invalid input parameters. 05 – Invalid POS terminal parameters. 06 – Missing PIN encryption key. 09 – Cancel key pressed (cancel by user).
0xA103	8	ASCII	M	Terminal ID.
0xA104	15	ASCII	M	Merchant ID.
0xA105	14	ASCII	M	Payment date in format YYYYMMDDHHMISS.
0xA106	12	ASCII	M	Approved amount.
0xA107	2	ASCII	C1	Response code returned by card host.
0xA108	..255	ASCII	C1	Response text returned by card host.
0xA109	6	ASCII	C1	STAN (System Trace Audit Number) / Receipt Number.
0xA10A	12	ASCII	C1	RRN (Return Reference Number).

Tag	Length	Format	Required	Description
0xA10B	6	ASCII	C2	Authorization code.
0xA10C	..19	ASCII	C1	PAN (Primary Account Number).
0xA10D	..40	ASCII	C1	Card holder name.
0xA10F	6	ASCII	C1	Batch number.
0xA113	..16	ASCII	C3	EMV application label.
0xA114	..32	ASCII	C3	EMV application ID.
0xA116	1	HEX	O	Transaction flags: 0x01 – Chip card used. 0x02 – Contactless card used. 0x04 – Magnetic Stripe card used. 0x08 – Signature requested. 0x10 – PIN was introduced. 0x20 – Offline transaction.
0xA117	..25	ASCII	C4	Unique transaction ID (from ECR).
0xA119	12	ASCII	O	Available offline spending amount.

***Require description:**

- M – mandatory.
- O – optional.
- C1 – mandatory when tag `0xA100` contains `00` – Success value.
- C2 – mandatory for approved sale by card transaction.
- C3 – mandatory when EFT-POS perform chip-based transaction.
- C4 – mandatory when is send in request.

Notes:

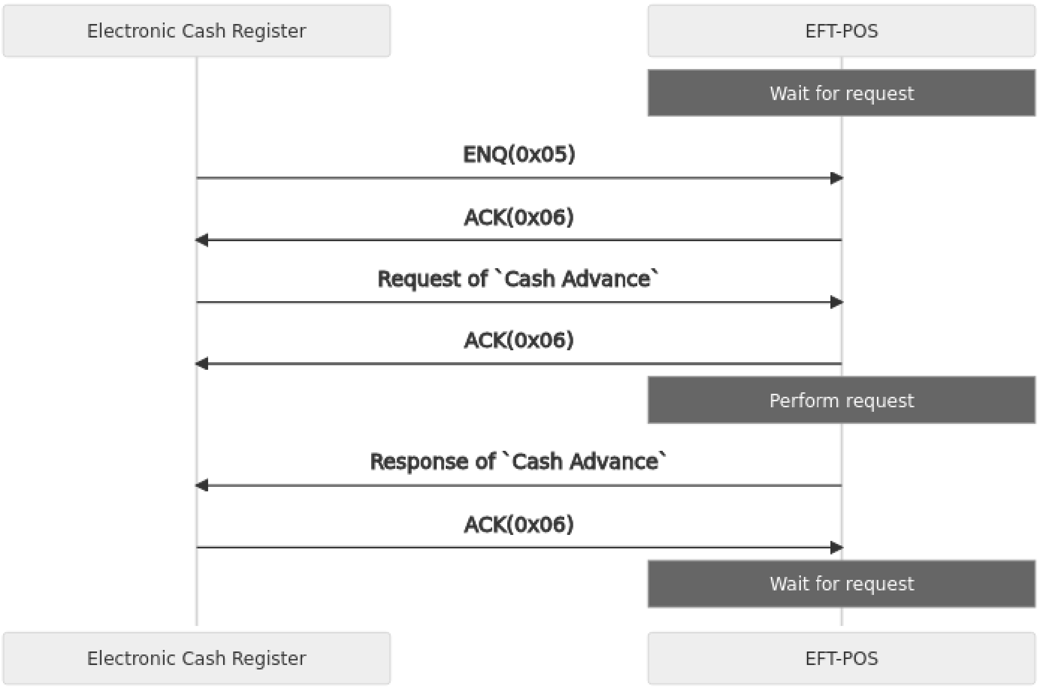
The cash advance transaction is approved (products or services are paid) only if tag `0xA100` contains value `00` and tag `0xA107` contains following values:

- `00` – approved by bank.

Example:

Flow

The operation flow is described in the following diagram:



Data

The data exchanged is described below:

```
SEND 1 Byte
05
RECEIVE 1 Byte
06
SEND 37 Bytes
02 00 1f a0 00 01 17 a0 01 0c 30 30 30 30 30 30
30 30 37 30 30 30 a0 02 03 52 4f 4e a0 03 03 39
34 36 03 dd 8a
RECEIVE 1 Byte
06
RECEIVE 193 Bytes
02 00 bb a1 00 01 00 a1 03 08 48 37 34 32 37 38
39 39 a1 04 0c 33 48 37 34 32 30 37 38 30 30 30
30 a1 05 0e 32 30 31 38 30 34 30 31 31 38 35 36
35 33 a1 06 0c 30 30 30 30 30 30 30 30 37 30 30
30 a1 07 02 30 30 a1 08 10 54 52 41 4e 5a 2e 20
41 43 43 45 50 54 41 54 41 a1 09 06 30 30 36 35
```

```
34 30 a1 0f 06 30 30 30 36 33 38 a1 0a 0c 30 31
39 34 36 38 39 36 31 33 39 36 a1 0b 06 35 30 33
36 30 37 a1 0c 06 2a 2a 37 30 34 37 a1 0d 0a 43
4c 49 45 4e 54 20 49 4e 47 a1 13 0d 56 69 73 61
20 45 6c 65 63 74 72 6f 6e a1 14 0e 41 30 30 30
30 30 30 30 30 33 31 30 31 30 a1 16 01 11 03 46
c5
SEND 1 Byte
06
```

4.4. POS Preauthorization

This function should be used to perform preauthorization by card transaction.

Input parameters:

Tag	Length	Format	Required	Description
0xA000	1	HEX	M	Command tag: 0C – Perform preauthorization.
0xA001	12	ASCII	M	Amount.
0xA002	3	ASCII	M	Currency name.
0xA003	3	ASCII	M	Currency code.
0xA008	..25	ASCII	O	Unique transaction ID.

Output parameters:

Tag	Length	Format	Required	Description
0xA100	1	HEX	M	Response tag: 00 – Success. 01 – Generic error. 02 – Connection to Cards Host error. 04 – Invalid input parameters. 05 – Invalid POS terminal parameters. 06 – Missing PIN encryption key. 09 – Cancel key pressed (cancel by user).
0xA103	8	ASCII	M	Terminal ID.
0xA104	15	ASCII	M	Merchant ID.
0xA105	14	ASCII	M	Payment date in format *YYYYMMDDHHMISS*.
0xA106	12	ASCII	M	Approved amount.
0xA107	2	ASCII	C1	Response code returned by card host.
0xA108	..255	ASCII	C1	Response text returned by card host.
0xA109	6	ASCII	C1	STAN (System Trace Audit Number) / Receipt Number.
0xA10A	12	ASCII	C1	RRN (Return Reference Number).
0xA10B	6	ASCII	C2	Authorization code.
0xA10C	..19	ASCII	C1	PAN (Primary Account Number).

Tag	Length	Format	Required	Description
0xA10D	..40	ASCII	C1	Card holder name.
0xA10F	6	ASCII	C1	Batch number.
0xA113	..16	ASCII	C3	EMV application label.
0xA114	..32	ASCII	C3	EMV application ID.
0xA116	1	HEX	O	Transaction flags: 0x01 – Chip card used. 0x02 – Contactless card used. 0x04 – Magnetic Stripe card used. 0x08 – Signature requested. 0x10 – PIN was introduced. 0x20 – Offline transaction.
0xA117	..25	ASCII	C4	Unique transaction ID (from ECR).
0xA119	12	ASCII	O	Available offline spending amount.
0xA128	10	ASCII	C5	Unique transaction code.

*__Require description:__

- M – mandatory.
- O – optional.
- C1 – mandatory when tag `0xA100` contains `00` – Success value.
- C2 – mandatory for approved sale by card transaction.
- C3 – mandatory when EFT-POS perform chip-based transaction.
- C4 – mandatory when is send in request.
- C5 – mandatory when EFT-POS is attended.

Notes:

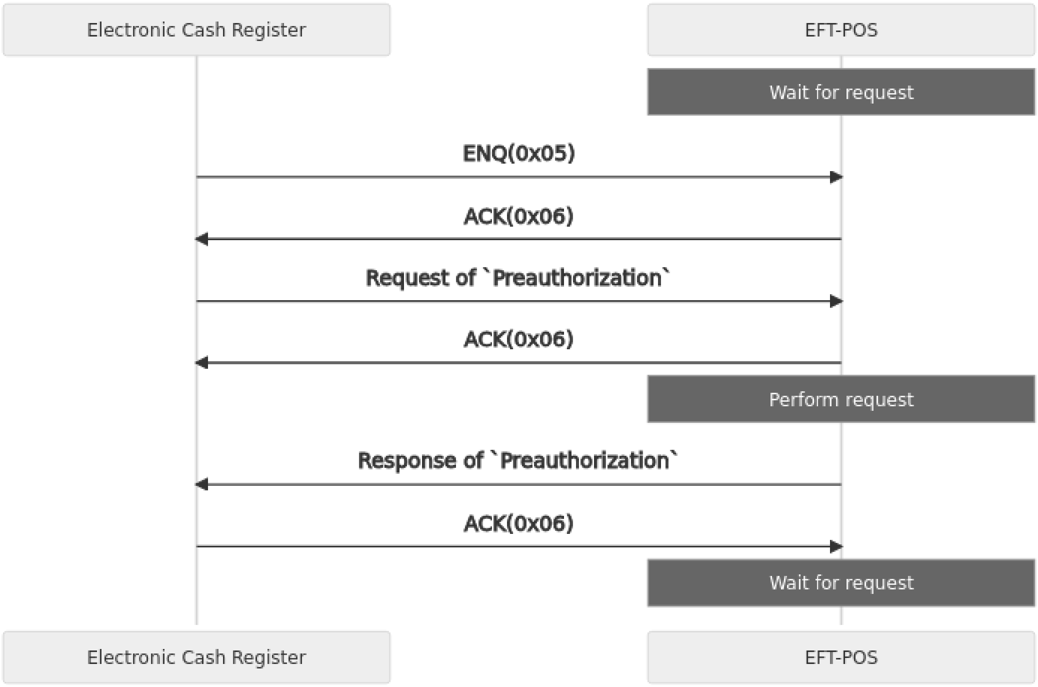
The preauthorization transaction is approved (amount is blocked) only if tag `0xA100` contains value `00` and tag `0xA107` contains following values:

- `00` – approved by bank.

Example:

Flow

The operation flow is described in the following diagram:



Data

The data exchanged is described below:

```
SEND 1 Byte
05
RECEIVE 1 Byte
06
SEND 37 Bytes
02 00 1f a0 00 01 0c a0 01 0c 30 30 30 30 30 30
30 30 30 31 32 33 a0 02 03 52 4f 4e a0 03 03 39
34 36 03 53 9c
RECEIVE 1 Byte
06
RECEIVE 176 Bytes
00 ab a1 00 01 00 a1 03 08 39 39 30 30 31 30 30
30 a1 04 0c 31 33 45 37 30 34 32 38 58 58 30 30
```

```
a1 05 0e 32 30 32 33 30 37 32 37 31 36 30 38 33
34 a1 06 0c 30 30 30 30 30 30 30 30 30 31 32 33
a1 07 02 30 30 a1 08 10 54 52 41 4e 5a 2e 20 41
43 43 45 50 54 41 54 41 a1 09 06 30 30 30 30 34
36 a1 0f 06 30 30 30 30 30 39 a1 0a 0c 30 30 37
33 39 30 35 31 30 31 38 31 a1 0b 06 35 31 30 31
38 31 a1 0c 06 2a 2a 37 39 36 32 A1 13 0a 4d 61
73 74 65 72 63 61 72 64 a1 14 0e 41 30 30 30 30
30 30 30 30 34 31 30 31 30 a1 16 01 03 03 26 67
SEND 1 Byte
06
```

4.5. POS Sale Completion

This function should be used to perform sale completion by card transaction.

Input parameters:

Tag	Length	Format	Required	Description
0xA000	1	HEX	M	Command tag: 0D – Perform sale completion.
0xA001	12	ASCII	M	Amount.
0xA002	3	ASCII	M	Currency name.
0xA003	3	ASCII	M	Currency code.
0xA008	..25	ASCII	O	Unique transaction ID.
0xA00A	12	ASCII	M	RRN (Return Reference Number) from preauthorization.
0xA00B	6	ASCII	M	Authorization Code from preauthorization.

Output parameters:

Tag	Length	Format	Required	Description
0xA100	1	HEX	M	Response tag: 00 – Success. 01 – Generic error. 02 – Connection to Cards Host error. 04 – Invalid input parameters. 05 – Invalid POS terminal parameters. 06 – Missing PIN encryption key. 09 – Cancel key pressed (cancel by user).
0xA103	8	ASCII	M	Terminal ID.
0xA104	15	ASCII	M	Merchant ID.
0xA105	14	ASCII	M	Payment date in format `YYYYMMDDHHMISS`.
0xA106	12	ASCII	M	Approved amount.
0xA107	2	ASCII	C1	Response code returned by card host.
0xA108	..255	ASCII	C1	Response text returned by card host.

Tag	Length	Format	Required	Description
0xA109	6	ASCII	C1	STAN (System Trace Audit Number) / Receipt Number.
0xA10A	12	ASCII	C1	RRN (Return Reference Number).
0xA10B	6	ASCII	C2	Authorization code.
0xA10C	..19	ASCII	C1	PAN (Primary Account Number).
0xA10D	..40	ASCII	C1	Card holder name.
0xA10F	6	ASCII	C1	Batch number.
0xA113	..16	ASCII	C3	EMV application label.
0xA114	..32	ASCII	C3	EMV application ID.
0xA116	1	HEX	O	Transaction flags: 0x01 – Chip card used. 0x02 – Contactless card used. 0x04 – Magnetic Stripe card used. 0x08 – Signature requested. 0x10 – PIN was introduced. 0x20 – Offline transaction.
0xA117	..25	ASCII	C4	Unique transaction ID (from ECR).
0xA119	12	ASCII	O	Available offline spending amount.

***Require description:**

- M – mandatory.
- O – optional.
- C1 – mandatory when tag `0xA100` contains `00` – Success value.
- C2 – mandatory for approved sale by card transaction.
- C3 – mandatory when EFT-POS perform chip-based transaction.
- C4 – mandatory when is send in request.

Notes:

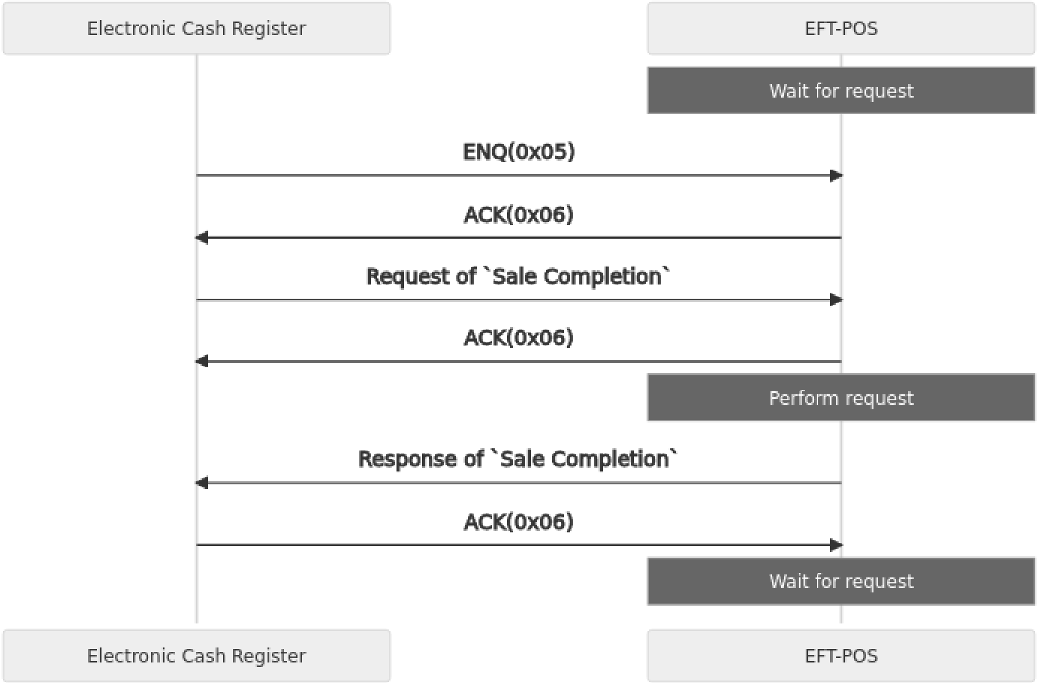
The sale completion transaction is approved (products or services are paid) only if tag *`0xA100`* contains value *`00`* and tag *`0xA107`* contains following values:

- *`00`* – approved by bank.
- *`Y1`* – approved offline.
- *`Y3`* – approved offline when cannot connect to the bank.

Example:

Flow

The operation flow is described in the following diagram:



Data

The data exchanged is described below:

```
SEND 1 Byte
05
RECEIVE 1 Byte
06
SEND 61 Bytes
02 00 37 a0 00 01 0d a0 01 0c 30 30 30 30 30 30
30 30 30 31 30 30 a0 02 03 52 4f 4e a0 03 03 39
34 36 a0 0a 0c 31 35 38 38 34 35 36 35 36 38 34
38 a0 0b 06 38 33 38 37 35 32 03 74 dd
RECEIVE 1 Byte
06
RECEIVE 138 Bytes
02 00 84 a1 00 01 00 a1 03 08 50 4f 53 30 30 30
```

```
30 31 a1 05 0e 32 30 32 31 30 31 32 36 31 39 34
35 35 39 a1 06 0c 30 30 30 30 30 30 30 30 30 31
30 30 a1 07 02 30 30 a1 08 10 54 52 41 4e 5a 2e
20 41 43 43 45 50 54 41 54 41 a1 09 06 30 30 30
30 30 36 a1 0f 06 30 30 30 30 30 31 a1 0a 0c 34
35 30 35 37 33 32 37 32 31 33 34 a1 0b 06 39 34
31 39 35 37 a1 0c 0c 35 31 37 30 34 35 2a 2a 34
35 31 30 a1 16 01 04 03 10 1e
SEND 1 Byte
06
```

4.6. POS Settlement

This function should be used to perform settlement or close day procedure with bank.

Input parameters:

Tag	Length	Format	Required	Description
0xA000	1	HEX	M	Command tag: 03 – Perform settlement.
0xA002	3	ASCII	M	Currency name.
0xA003	3	ASCII	M	Currency code.

Output parameters:

Tag	Length	Format	Required	Description
0xA100	1	HEX	M	Response tag: 00 – Success. 01 – Generic error. 02 – Connection to Card Host error. 04 – Invalid input parameters. 05 – Invalid POS terminal parameters.
0xA103	8	ASCII	M	Terminal ID.
0xA104	15	ASCII	M	Merchant ID.
0xA105	14	ASCII	M	Payment date in format `YYYYMMDDHHMISS`.
0xA107	2	ASCII	C1	Response code returned by card host.
0xA108	..255	ASCII	C1	Response text returned by card host.
0xA10E	3	ASCII	C1	Transactions count (number of transactions in batch).
0xA10F	6	ASCII	C1	Batch number.
0xA110	12	ASCII	C1	Transactions total amount.

***Require description:**

- M – mandatory.
- O – optional.
- C1 – mandatory when tag `0xA100` contains `00` – Success value.

Notes:

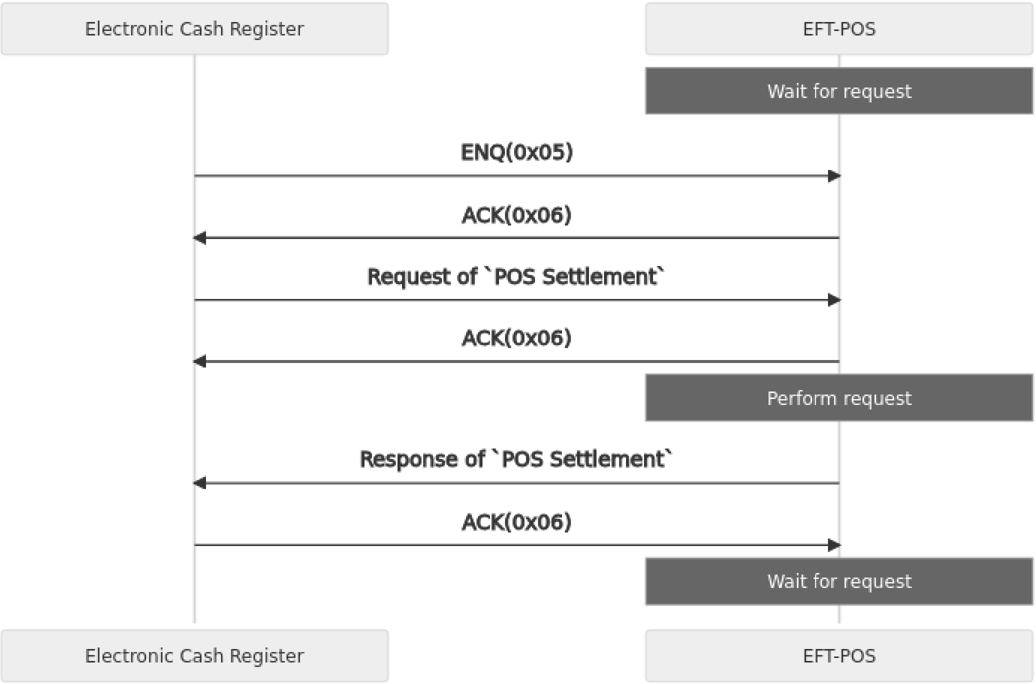
The settlement transaction is approved only if tag `0xA107` contains the following values:

- `00` – approved by bank.

Example:

Flow

The operation flow is described in the following diagram:



Data

The data exchanged is described below:

```
SEND 1 Byte
05
RECEIVE 1 Byte
06
SEND 22 Bytes
02 00 10 a0 00 01 03 a0 02 03 52 4f 4e a0 03 03
39 34 36 03 36 b9
RECEIVE 1 Byte
06
```

```
RECEIVE 107 Bytes
02 00 65 a1 00 01 00 a1 03 08 30 30 30 36 30 31
30 36 a1 04 0c 33 30 30 30 36 30 30 31 30 30 30
30 a1 05 0e 32 30 31 38 30 38 31 30 31 30 35 37
31 39 a1 07 02 30 30 a1 08 10 54 52 41 4e 5a 2e
20 41 43 43 45 50 54 41 54 41 a1 0e 03 30 30 30
a1 0f 06 30 30 30 31 36 37 a1 10 0c 30 30 30 30
30 30 30 30 30 30 30 30 03 62 ff
SEND 1 Byte
06
```

4.7. POS Report Initialization

This function should be used to retrieve current batch details.

Input parameters:

Tag	Length	Format	Required	Description
0xA000	1	HEX	M	Command tag: 04 – Perform report initialization.
0xA002	3	ASCII	M	Currency name.
0xA003	3	ASCII	M	Currency code.

Output parameters:

Tag	Length	Format	Required	Description
0xA100	1	HEX	M	Response tag: 00 – Success. 01 – Generic error. 04 – Invalid input parameters. 05 – Invalid POS terminal parameters.
0xA10F	6	ASCII	M	Batch number.
0xA10E	3	ASCII	C1	Transactions count.
0xA10D	12	ASCII	C1	Transaction total amount.

*__Require description:__

- M – mandatory.
- O – optional.
- C1 – mandatory when tag `0xA100` contains `00` – Success value.

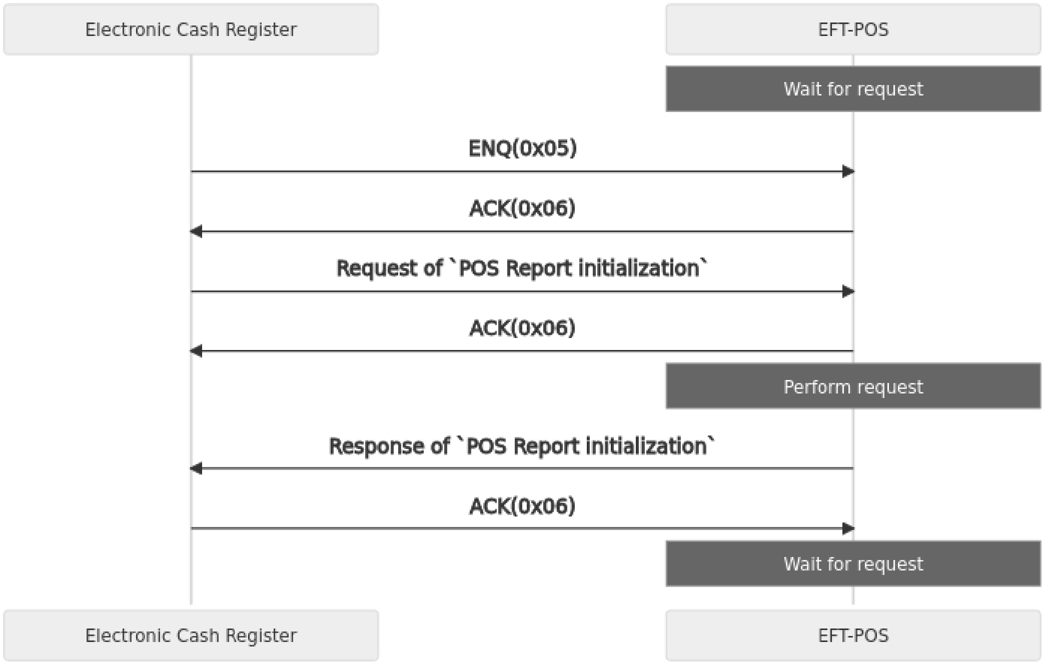
Notes:

This is local command (no message exchanged with bank host).

Example:

Flow

The operation flow is described in the following diagram:



Data

The data exchanged is described below:

```
SEND 1 Byte
05
RECEIVE 1 Byte
06
SEND 22 Bytes
02 00 10 a0 00 01 04 a0 02 03 52 4f 4e a0 03 03
39 34 36 03 48 3f
RECEIVE 1 Byte
06
RECEIVE 40 Bytes
02 00 22 a1 00 01 00 a1 0e 03 30 30 30 a1 0f 06
30 30 30 30 30 31 a1 10 0c 30 30 30 30 30 30 30
30 30 30 30 30 03 04 b5
SEND 1 Bytes
06
```

4.8. POS Report Record

This function should be used to retrieve transaction details from terminal journal/batch.

Input parameters:

Tag	Length	Format	Required	Description
0xA000	1	HEX	M	Command tag: 05 – Get report record.
0xA004	3	ASCII	M	Transaction index (zero based index).

Output parameters:

Tag	Length	Format	Required	Description
0xA100	1	HEX	M	Response tag: 00 – Success. 01 – Generic error. 03 – Out of range (transaction missing in batch). 04 – Invalid input parameters. 05 – Invalid POS terminal parameters.
0xA103	8	ASCII	M	Terminal ID.
0xA104	15	ASCII	M	Merchant ID.
0xA105	14	ASCII	M	Payment date in format *YYYYMMDDHHMISS*.
0xA106	12	ASCII	M	Approved amount.
0xA107	2	ASCII	C1	Response code returned by card host.
0xA108	..255	ASCII	C1	Response text returned by card host.
0xA109	6	ASCII	C1	STAN (System Trace Audit Number) / Receipt Number.
0xA10A	12	ASCII	C1	RRN (Return Reference Number).
0xA10B	6	ASCII	C2	Authorization code.
0xA10C	..19	ASCII	C1	PAN (Primary Account Number).
0xA10D	..40	ASCII	C1	Card holder name.
0xA10F	6	ASCII	C1	Batch number.
0xA112	1	HEX	M	Transaction type: 0 – Sale. 1 – Voided sale. 2 – Void.
0xA113	..16	ASCII	C3	EMV application label.
0xA114	..32	ASCII	C3	EMV application ID.

Tag	Length	Format	Required	Description
0xA116	1	HEX	M	Transaction flags: 0x01 – Chip card used. 0x02 – Contactless card used. 0x04 – Magnetic Stripe card used. 0x08 – Signature requested. 0x10 – PIN was introduced. 0x20 – Offline transaction.
0xA117	..25	ASCII	C4	Unique transaction ID (from ECR).

*__Require description:__

- M – mandatory.
- O – optional.
- C1 – mandatory when is used when transaction was performed.

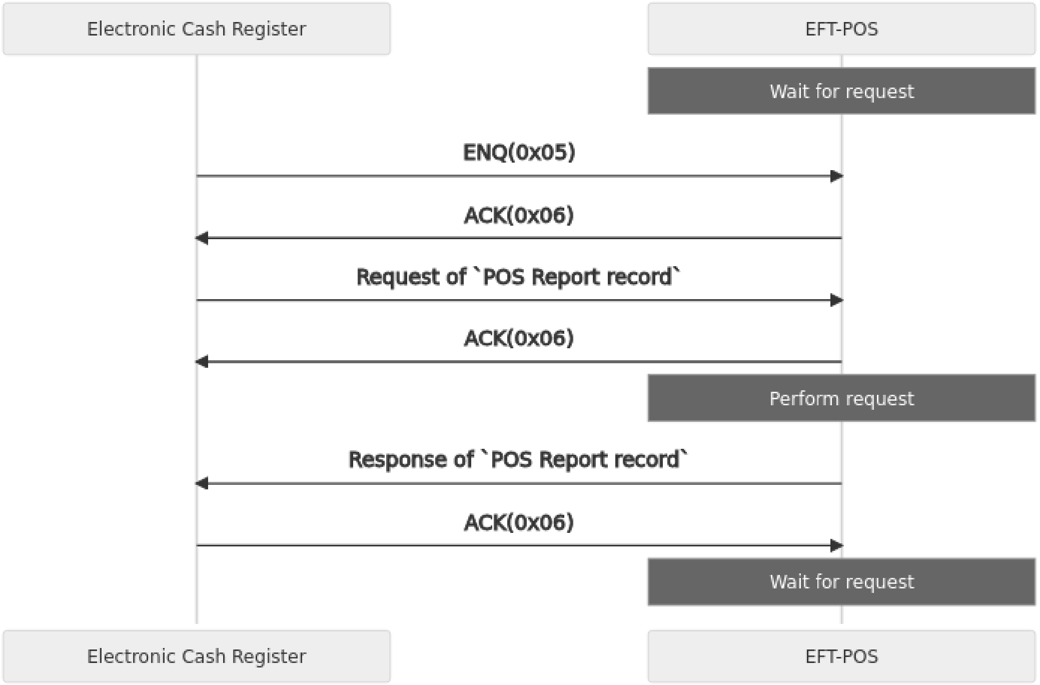
Notes:

This is local command (no message exchanged with bank host).

Example:

Flow:

The operation flow is described in the following diagram:



Data

The data exchanged is described below:

```
SEND 1 Byte
05
RECEIVE 1 Byte
06
SEND 65 Bytes
02 00 3b a0 00 01 02 a0 01 0c 30 30 30 30 30 30
30 30 30 32 30 30 a0 02 03 52 4f 4e a0 03 03 39
34 36 a0 08 0a 31 31 32 32 33 33 34 34 35 35 a0
07 0c 30 30 30 30 30 30 30 30 30 30 30 30 03 7d
66
RECEIVE 1 Byte
06
RECEIVE 193 Bytes
02 00 bb a1 00 01 00 a1 03 08 50 4f 53 30 30 30
30 31 a1 05 0e 32 30 32 31 30 31 32 36 32 32 32
32 32 30 a1 06 0c 30 30 30 30 30 30 30 30 30 32
30 30 a1 07 02 30 30 a1 08 10 54 52 41 4e 5a 2e
```

```
20 41 43 43 45 50 54 41 54 41 a1 09 06 30 30 30
30 30 37 a1 0f 06 30 30 30 30 30 31 a1 0a 0c 34
31 32 32 31 36 32 30 32 32 39 37 a1 0b 06 34 33
35 36 31 34 a1 0c 0c 35 31 37 30 34 35 2a 2a 34
35 31 30 a1 13 0a 4d 61 73 74 65 72 43 61 72 64
a1 14 0e 41 30 30 30 30 30 30 30 30 34 31 30 31
30 a1 17 0a 31 31 32 32 33 33 34 34 35 35 a1 1b
03 52 4f 4e a1 1c 03 39 34 36 a1 16 01 11 03 1d
94
SEND 1 Byte
06
```

4.9. POS Void

This function should be used to perform cancelation of sale and sale completion transaction from batch.

Input parameters:

Tag	Length	Format	Required	Description
0xA000	1	HEX	M	Command tag: 06 – Perform void.
0xA006	6	ASCII	M	STAN (System Trace Audit Number) / Receipt Number used to identify transaction from batch.
0xA001	12	ASCII	M	Amount.

Output parameters:

Tag	Length	Format	Required	Description
0xA100	1	HEX	M	Response tag: 00 – Success. 01 – Generic Error. 02 – Connection to bank host error. 04 – Invalid Input Parameters. 05 – Invalid POS terminal parameters.
0xA103	8	ASCII	M	Terminal ID.
0xA104	15	ASCII	M	Merchant ID.
0xA105	14	ASCII	M	Payment date in format *YYYYMMDDHHMISS*.
0xA106	12	ASCII	M	Approved amount.
0xA107	2	ASCII	C1	Response code returned by card host.
0xA108	..255	ASCII	C1	Response text returned by card host.
0xA109	6	ASCII	C1	STAN (System Trace Audit Number) / Receipt Number.
0xA10A	12	ASCII	C1	RRN.
0xA10B	6	ASCII	C2	Authorization code.
0xA10C	..19	ASCII	C1	PAN.

Tag	Length	Format	Required	Description
0xA10D	..40	ASCII	C1	Card holder name.
0xA117	..25	ASCII	C3	Unique transaction ID (from ECR).

*__Require description:__

- M – mandatory.
- O – optional.
- C1 – mandatory when tag 0xA100 contains 00 – Success value.
- C2 – mandatory for approved void transaction.
- C3 – mandatory when is send in request.

Notes:

The void transaction is approved only if tag `0xA107` contains the following values:

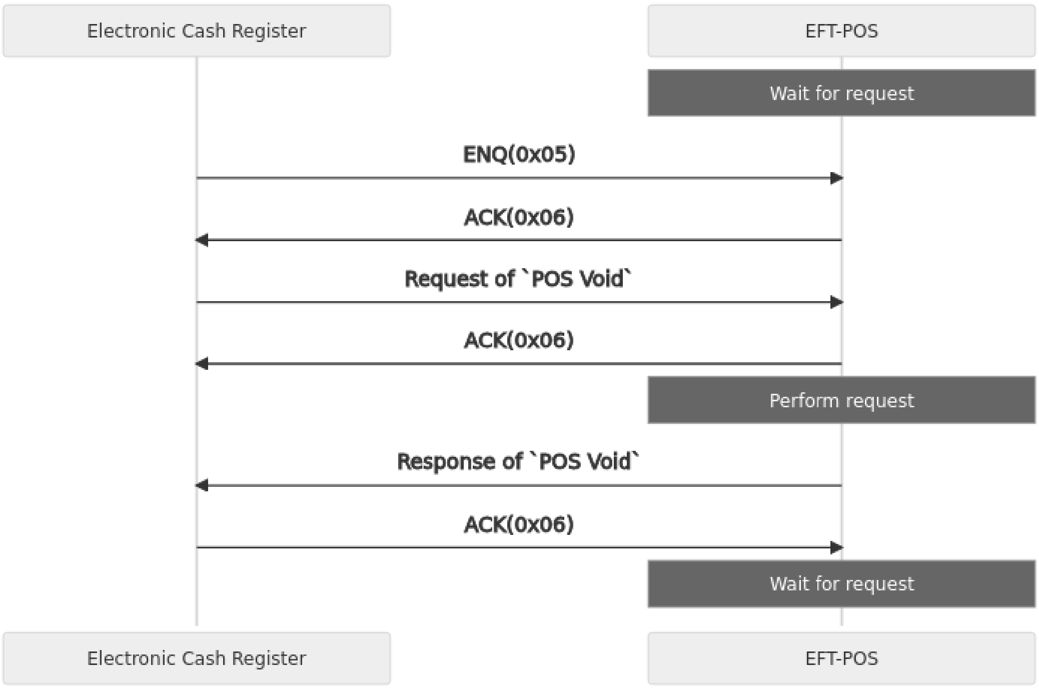
- `00` – approved by bank.

If command return error, the ECR must repeat the command later to void transaction.

Example:

Flux

The operation flow is described in the following diagram:



Data

The data exchanged is described below:

```
SEND 1 Byte
05
RECEIVE 1 Byte
06
SEND 34 Bytes
02 00 1c a0 00 01 06 a0 01 0c 30 30 30 30 30 30
30 30 30 30 31 30 a0 06 06 30 30 32 32 32 33 03
d8 77
RECEIVE 1 Byte
06
RECEIVE 153 Bytes
02 00 93 a1 00 01 00 a1 03 08 30 30 30 36 30 31
30 36 a1 04 0c 33 30 30 30 36 30 30 31 30 30 30
30 a1 05 0e 32 30 31 38 30 38 31 30 31 30 35 36
34 38 a1 06 0c 30 30 30 30 30 30 30 30 30 30 31
30 a1 07 02 30 30 a1 08 10 54 52 41 4e 5a 2e 20
41 43 43 45 50 54 41 54 41 a1 09 06 30 30 32 32
32 34 a1 0f 06 30 30 30 31 36 37 a1 0a 0c 30 31
39 30 33 34 38 30 38 36 31 37 a1 0b 06 35 36 30
31 37 33 a1 0c 0c 35 31 30 31 36 30 2a 2a 39 31
38 32 a1 16 01 00 03 e5 91
SEND 1 Byte
06
```

4.10. POS Get Cash Back Service Allowed and Limit

This function should be used to retrieve if Cash Back service is allowed for client card and Cash Back limit amount. If the card is magnetic stripe the terminal (EFT POS) should temporarily save the track2 data to perform sale transaction without asking the card again. If the card is contactless the terminal (EFT POS) should directly perform sale transaction without cashback amount. If the client wants cashback should insert or swipe the card.

Input parameters:

Tag	Length	Format	Required	Description
0xA000	1	HEX	M	Command tag: 07 – Get Cash Back Service Allowed and Limit.
0xA001	12	ASCII	M	Amount.
0xA002	3	ASCII	O	Currency name.
0xA003	3	ASCII	O	Currency code.
0xA008	..25	ASCII	O	Unique transaction ID (if is the same ID like for Get POS Cash Back Service Allowed the terminal should not ask the card again).

Output parameters:

Tag	Length	Format	Required	Description
0xA100	1	HEX	M	Response tag: 00 – Success. 01 – Generic Error. 04 – Invalid input parameters. 05 – Invalid POS terminal parameters. 07 – Cash Back Service Allowed. 08 – Cash Back Service Not Allowed. 09 – Cancel key pressed (cancel by user).
0xA115	12	ASCII	C1	Cash Back Limit Amount.
0xA103	8	ASCII	M	Terminal ID.
0xA104	15	ASCII	M	Merchant ID.
0xA105	14	ASCII	M	Payment date in format `YYYYMMDDHHMISS`.
0xA117	..25	ASCII	C2	Unique transaction ID (from ECR).

*__Require description:__

- M – mandatory.
- O – optional.
- C1 – mandatory when tag `0xA100` contains `07` – Cash Back Service Allowed.
- C2 – mandatory when is send in request.

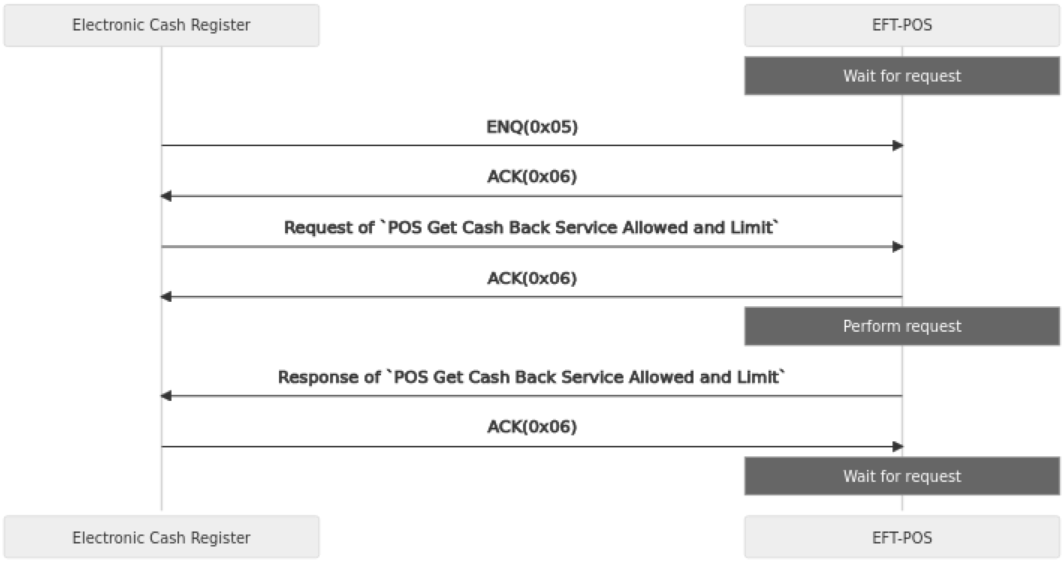
Notes:

This is local command (no message exchanged with bank host).

Example:

Flow

The operation flow is described in the following diagram:



Data

The data exchanged is described below:

```
SEND 1 Byte
05
RECEIVE 1 Byte
06
SEND 37 Bytes
02 00 1f a0 00 01 07 a0 01 0c 30 30 30 30 30 30
30 30 31 30 30 30 a0 02 03 52 4f 4e a0 03 03 39
34 36 03 21 9c
RECEIVE 1 Byte
06
RECEIVE 14 Bytes:
02 00 08 a1 00 01 08 a1 16 01 00 03 32 fb
SEND 1 Byte
06
```

4.11. POS Send Offline Transactions

This function should be used to operate settlement or close day procedure with bank.

Input parameters:

Tag	Length	Format	Required	Description
0xA000	1	HEX	M	Command tag: 08 – Send offline transactions
0xA002	3	ASCII	M	Currency name
0xA003	3	ASCII	M	Currency code

Output parameters:

Tag	Length	Format	Required	Description
0xA100	1	HEX	M	Response tag: 00 – Success. 01 – Generic error. 02 – Connection to Card Host error. 04 – Invalid input parameters. 05 – Invalid POS terminal parameters.
0xA103	8	ASCII	M	Terminal ID.
0xA104	15	ASCII	M	Merchant ID.
0xA105	14	ASCII	M	Payment date in format `YYYYMMDDHHMISS`.
0xA10E	3	ASCII	C1	Transactions count (number of transactions sent to host).
0xA10F	6	ASCII	C1	Batch number.
0xA110	12	ASCII	C1	Uploaded transactions total amount.

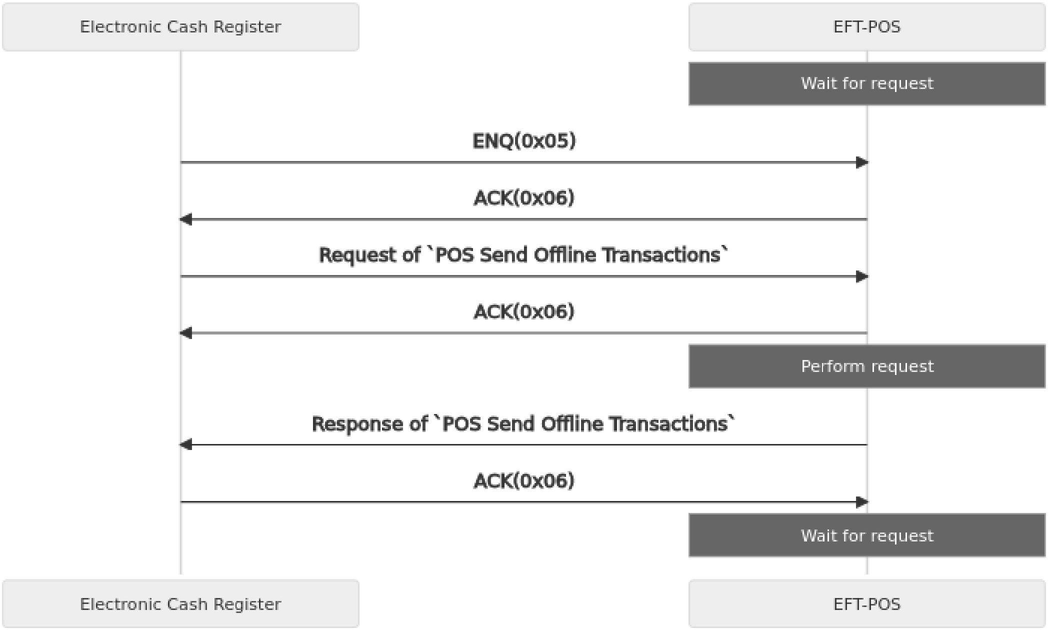
Required:

- M – mandatory.
- C1 – mandatory when tag `0xA100` contains `00` – Success value and found offline transactions.

Example:

Flow

The operation flow is described in the following diagram:



Data

The data exchanged is described below:

```
SEND 1 Byte
05
RECEIVE 1 Byte
06
SEND 22 Bytes
02 00 10 a0 00 01 08 a0 02 03 52 4f 4e a0 03 03
39 34 36 03 ca 30
RECEIVE 1 Byte
06
RECEIVE 79 Bytes
00 4a a1 00 01 00 a1 03 08 50 4f 53 30 30 30 30
31 a1 05 0e 32 30 32 31 30 31 32 36 32 32 32 37
33 35 a1 0e 03 30 30 30 a1 0f 06 30 30 30 30 30
31 a1 10 0c 30 30 30 30 30 30 30 30 30 30 30 30
a1 1b 03 52 4f 4e a1 1c 03 39 34 36 03 67 c0
SEND 1 Byte
06
```

4.12. POS Remove card

This function should be used to ask the client to remove the card. For example, if the client has contact chip card, the ECR sends to terminal *Get POS Cash Back Service Allowed and Limit* receive the answer and then the client wants to terminate the transaction. The terminal does not ask the client to remove the card after the command *Get POS Cash Back Service Allowed and Limit* is finished.

Input parameters:

Tag	Length	Format	Required	Description
0xA000	1	HEX	M	Command tag: 09 – Remove Card

Output parameters:

Tag	Length	Format	Required	Description
0xA100	1	HEX	M	Response tag: 00 – Success. 01 – Generic Error.

Required:

- M – mandatory.

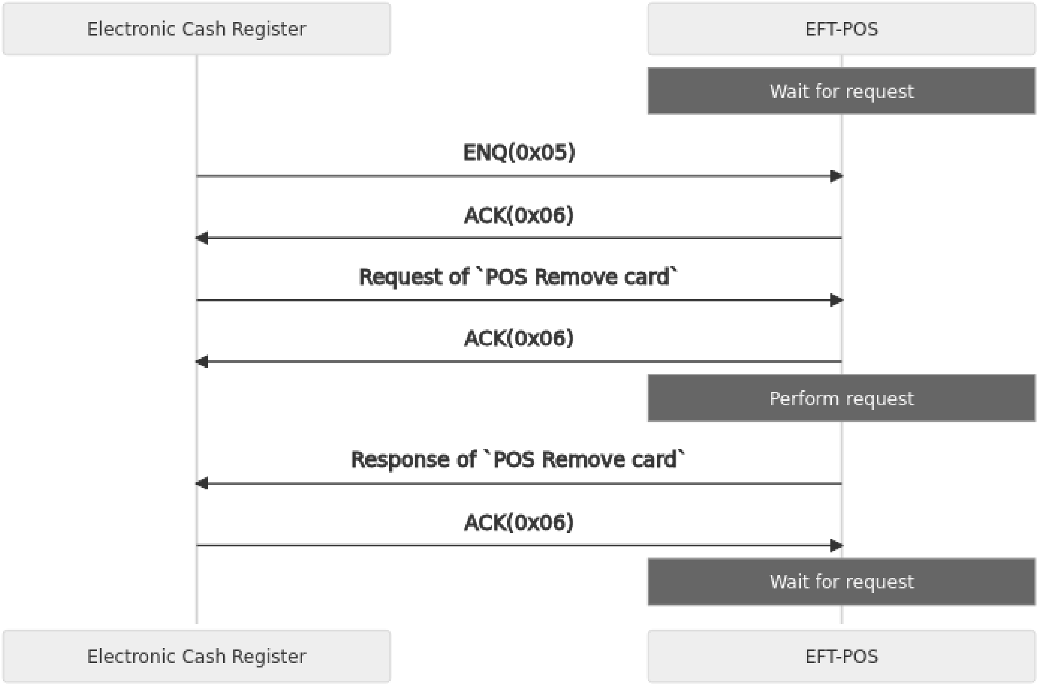
Notes:

This is local command (no message exchanged with bank host).

Example:

Flow

The operation flow is described in the following diagram:



Data

The data exchanged is described below:

```
SEND 1 Byte
05
RECEIVE 1 Byte
06
SEND 10 Bytes
02 00 04 a0 00 01 09 03 86 06
RECEIVE 1 Byte
06
RECEIVE 10 Bytes
02 00 04 a1 00 01 00 03 33 12
SEND 1 Byte
06
```

4.13. POS Read Card Fleet Data

This function should be used to read card fleet data.

Input parameters:

Tag	Length	Format	Required	Description
0xA000	1	HEX	M	Command tag: 0E – Read card fleet data

Output parameters:

Tag	Length	Format	Required	Description
0xA100	1	HEX	M	Response tag: 00 – Success. 01 – Generic Error.
0xA11A	..256	HEX	Optional	Card Fleet Data

Required:

- M – mandatory.
- O – optional.

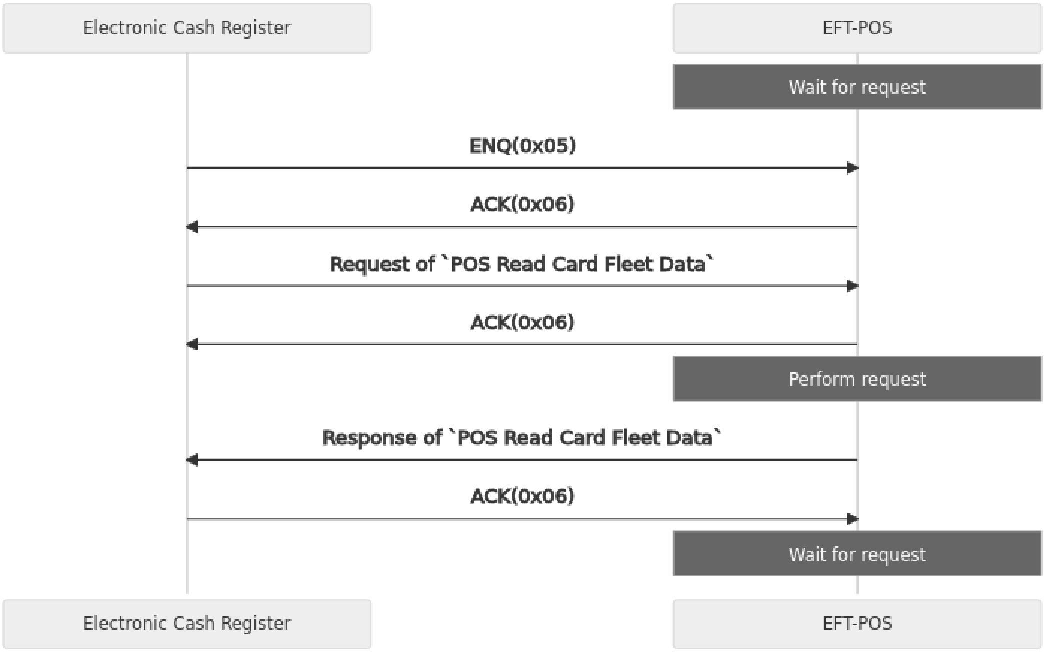
Notes:

This is local command (no message exchanged with bank host).

Example:

Flow

The operation flow is described in the following diagram:



Data

The data exchanged is described below:

```
TODO: fill with sample after implement command
```

4.14. POS Get card token

This command should be used to activate the function of reading the card in the form of a token on the EFTPOS. This command is not financial.

Input parameters:

Tag	Length	Format	Required	Description
0xA000	1	HEX	M	Command tag: 18 – Perform get card token.
0xA008	..25	ASCII	O	Unique operation ID (received from ECR).

Output parameters:

Tag	Length	Format	Required	Description
0xA100	1	HEX	M	Response tag: 00 – Success. 01 – Generic error. 09 – Cancel key pressed (cancel by user).
0xA150	64	ASCII	C1	Card token
0xA117	..25	ASCII	C2	Unique operation ID (received from ECR).

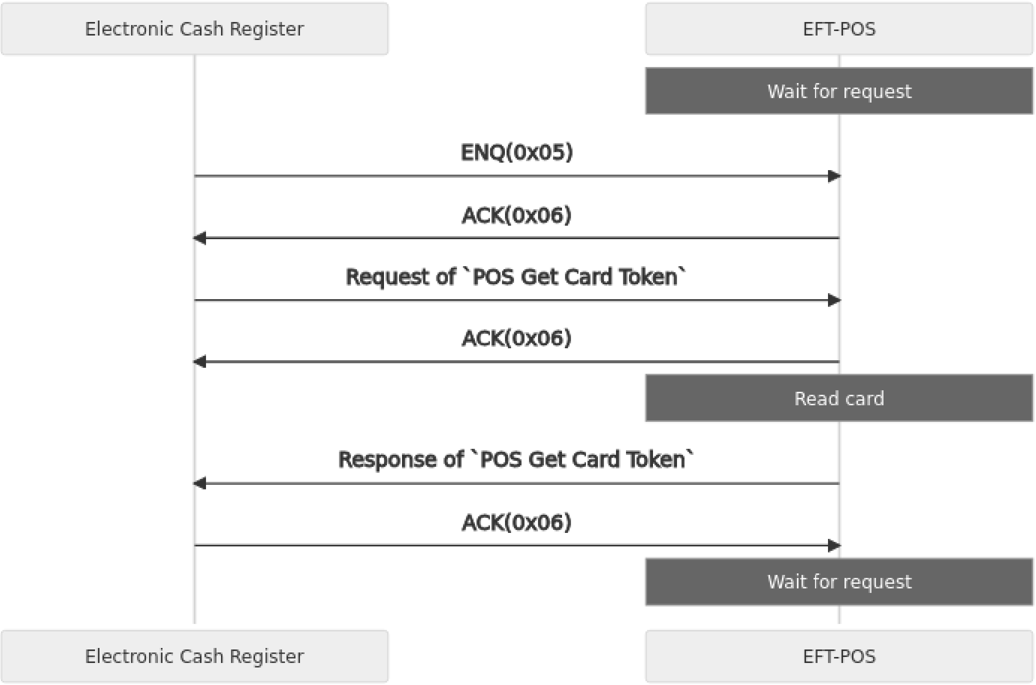
Required:

- M – mandatory
- O – optional
- C1 – mandatory when tag `0xA100` contains `00` – Success value.
- C2 – mandatory when is send in request.

Example:

Flow

The following diagram describes the successful flow to remotely cancel ongoing sale by card operation:



Data

The data exchanged is described below:

```
SEND 1 Byte
05
RECEIVE 1 Byte
06
SEND 23 Bytes
02 00 11 A0 00 01 18 A0 08 0A 30 31 32 33 34 35
36 37 38 39 03 1F 5F
RECEIVE 1 Byte
06
RECEIVE 90 Bytes
02 00 54 A1 00 01 00 A1 50 40 42 37 35 41 38 46
33 43 42 36 37 45 41 46 33 34 45 35 35 33 45 30
43 34 34 43 31 44 30 33 44 34 35 31 43 33 30 30
37 42 42 44 38 39 41 46 32 42 43 32 45 36 43 31
36 38 44 33 37 32 31 36 33 35 A1 17 0A 30 31 32
33 34 35 36 37 38 39 03 29 18
```

```
SEND 1 Byte
06
```

4.15. Cancel transaction / operation

This function can be sent between request and response of sale by card or get card token. This function is available only on Self terminals.

Input parameters:

Tag	Length	Format	Required	Description
0xA000	1	HEX	M	Command tag: 20 – Perform remote cancel.

Output parameters:

Tag	Length	Format	Required	Description
0xA100	1	HEX	M	Response tag: 0A – Transaction/operation has been canceled. 0B – Transaction/operation has not been canceled.

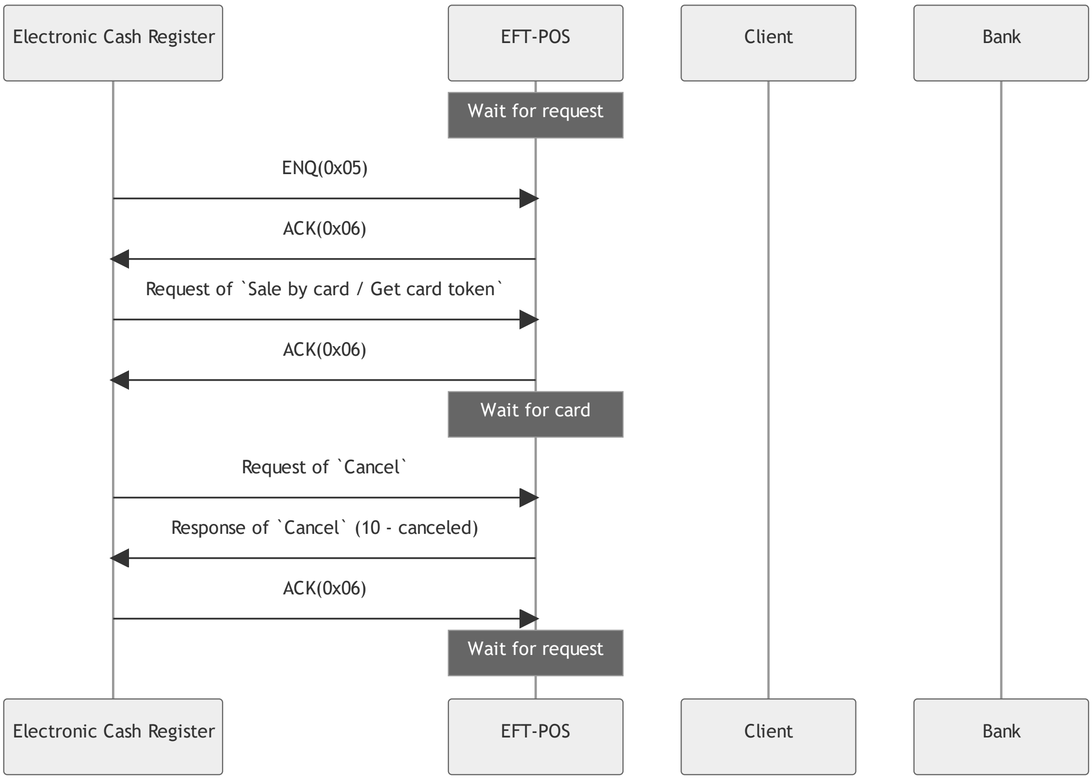
Required:

- M – mandatory.

Example:

Flow

The following diagram describes the successful flow to remotely cancel ongoing sale by card operation:



Data

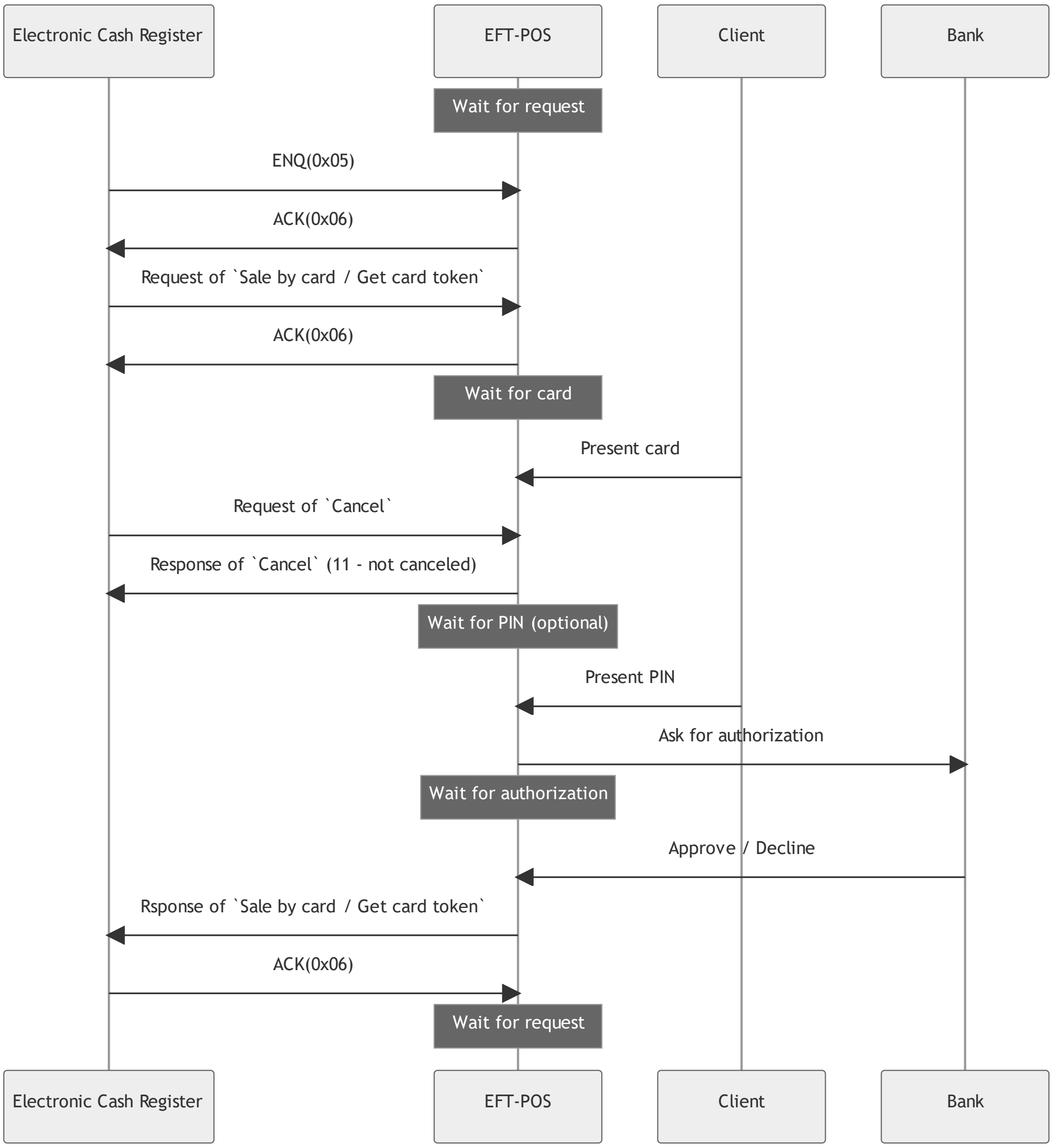
The data exchanged is described below:

```
SEND 1 Byte
05
RECEIVE 1 Byte
06
SEND 52 Bytes
02 00 2E A0 00 01 02 A0 01 0C 30 30 30 30 30 30
30 30 30 31 30 30 A0 02 03 52 4F 4E A0 03 03 39
34 36 A0 07 0C 30 30 30 30 30 30 30 30 30 30 30
30 03 9D 96
RECEIVE 1 Byte
06
SEND 10 Bytes
02 00 04 A0 00 01 20 03 06 F3
RECEIVE 1 Byte
06
RECEIVE 10 Bytes
```

```
02 00 04 A1 00 01 0A 03 0F 12
SEND 1 Byte
06
```

Flow

The following diagram describes the unsuccessful flow to remotely cancel ongoing sales by card operation:



Data

The data exchanged is described below:

```
SEND 1 Byte
05
RECEIVE 1 Byte
06
SEND 52 Bytes
02 00 2E A0 00 01 02 A0 01 0C 30 30 30 30 30 30
30 30 30 31 30 30 A0 02 03 52 4F 4E A0 03 03 39
34 36 A0 07 0C 30 30 30 30 30 30 30 30 30 30 30
30 03 9D 96
RECEIVE 1 Byte
06
SEND 10 Bytes
02 00 04 A0 00 01 20 03 06 F3
RECEIVE 1 Byte
06
RECEIVE 10 Bytes
02 00 04 A1 00 01 0B 03 0A 92
SEND 1 Byte
06
RECEIVE 174 Bytes
02 00 A8 A1 00 01 00 A1 03 08 50 4F 53 55 30 30
30 32 A1 04 08 50 4F 53 55 30 30 30 32 A1 05 0E
32 30 32 32 30 33 31 34 31 34 33 39 32 37 A1 06
0C 30 30 30 30 30 30 30 30 30 31 30 30 A1 07 02
30 30 A1 08 10 54 52 41 4E 5A 2E 20 41 43 43 45
50 54 41 54 41 A1 09 06 30 30 30 32 37 32 A1 0A
0C 37 38 32 35 33 34 39 30 32 30 31 39 A1 0B 06
45 4B 48 49 43 39 A1 0C 06 2A 2A 32 35 36 30 A1
0F 06 30 30 30 30 30 33 A1 13 0B 56 69 73 61 20
43 72 65 64 69 74 A1 14 0E 41 30 30 30 30 30 30
30 30 33 31 30 31 30 A1 16 01 11 03 50 3F
SEND 1 Byte
06
```

4.16. POS Void Preauthorization

This function should be used to perform cancelation of any preauthorization transaction.

Input parameters:

Tag	Length	Format	Required	Description
0xA000	1	HEX	M	Command tag: 10 – Perform void preauthorization.
0xA006	6	ASCII	M	STAN (System Trace Audit Number) / Receipt Number used to identify transaction.
0xA001	12	ASCII	M	Amount.
0xA012	10	ASCII	C4	Unique transaction code.

Output parameters:

Tag	Length	Format	Required	Description
0xA100	1	HEX	M	Response tag: 00 – Success. 01 – Generic Error. 02 – Connection to bank host error. 04 – Invalid Input Parameters. 05 – Invalid POS terminal parameters.
0xA103	8	ASCII	M	Terminal ID.
0xA104	15	ASCII	M	Merchant ID.
0xA105	14	ASCII	M	Payment date in format `YYYYMMDDHHMISS`.
0xA106	12	ASCII	M	Approved amount.
0xA107	2	ASCII	C1	Response code returned by card host.
0xA108	..255	ASCII	C1	Response text returned by card host.
0xA109	6	ASCII	C1	STAN (System Trace Audit Number) / Receipt Number.
0xA10A	12	ASCII	C1	RRN.
0xA10B	6	ASCII	C2	Authorization code.
0xA10C	..19	ASCII	C1	PAN.
0xA10D	..40	ASCII	C1	Card holder name.
0xA117	..25	ASCII	C3	Unique transaction ID (from ECR).

*__Require description:__

- M – mandatory.
- O – optional.
- C1 – mandatory when tag 0xA100 contains 00 – Success value.
- C2 – mandatory for approved void transaction.
- C3 – mandatory when is send in request.
- C4 – mandatory for attended terminals.

Notes:

The void preauthorization transaction is approved only if tag `0xA107` contains the following values:

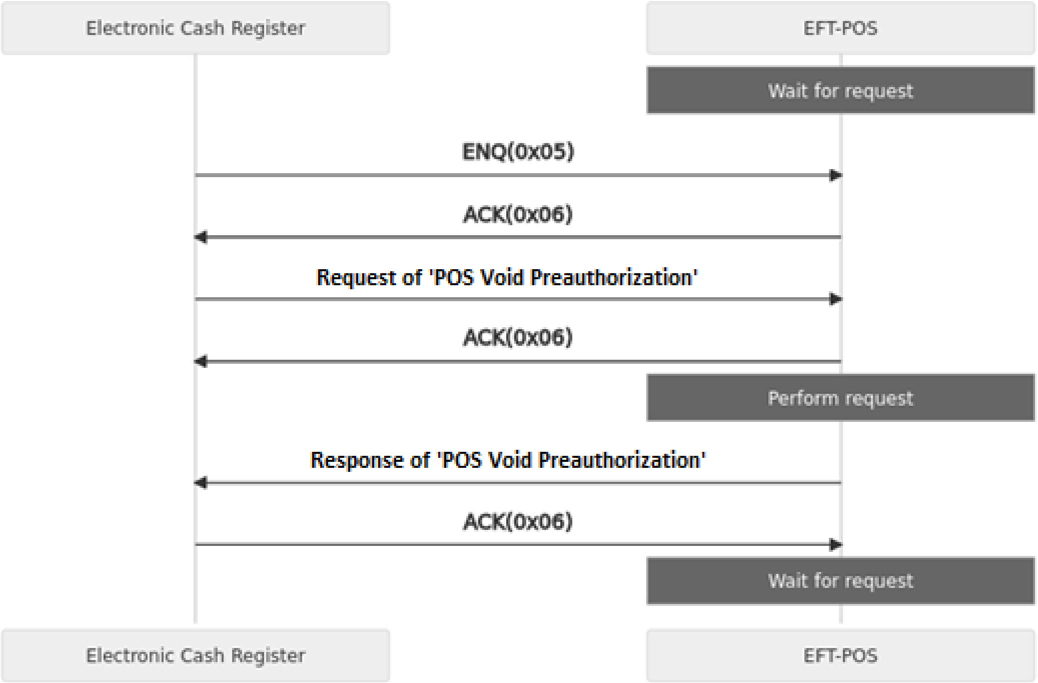
- `00` – approved by bank.

If command return error, the ECR must repeat the command later to void preauthorization transaction.

Example:

Flux

The operation flow is described in the following diagram:



Data

The data exchanged is described below:

```
SEND 1 Byte
05
RECEIVE 1 Byte
06
SEND 49 Bytes
02 00 2B A0 00 01 10 A0 01 0C 30 30 30 30 30 30
30 30 30 31 31 31 A0 02 03 52 4F 4E A0 03 03 39
34 36 A0 06 06 30 30 30 30 32 38 A0 12 00 03 58
B6
RECEIVE 1 Byte
06
RECEIVE 143 Bytes
02 00 89 A1 00 01 00 A1 03 08 54 50 41 59 55 30
30 31 A1 04 08 54 50 41 59 55 30 30 31 A1 05 0E
32 30 32 33 30 37 32 38 31 35 33 36 34 32 A1 06
0C 30 30 30 30 30 30 30 30 30 31 31 31 A1 07 02
30 30 A1 08 10 54 52 41 4E 5A 2E 20 41 43 43 45
50 54 41 54 41 A1 09 06 30 30 30 30 33 31 A1 0A
0C 31 32 32 36 39 31 30 34 32 37 34 30 A1 0B 06
33 32 38 30 33 37 A1 0C 06 2A 2A 37 39 36 32 A1
```

```
0F 06 30 30 30 30 30 35 A1 16 01 04 03 B2 2F
SEND 1 Byte
06
```

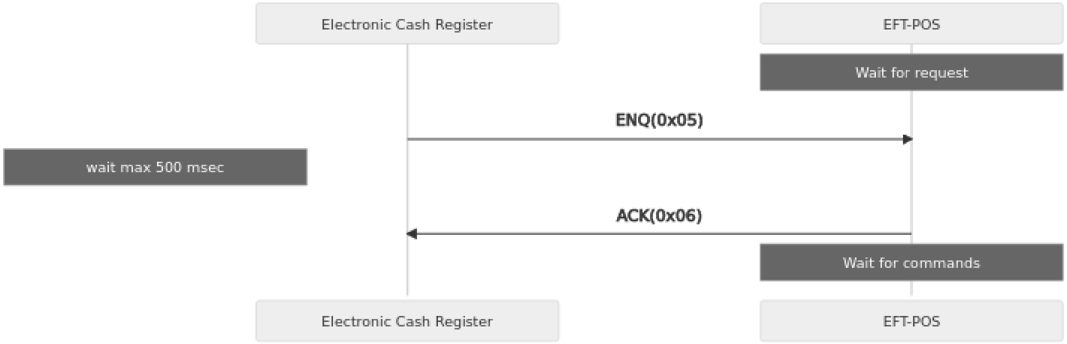
5. Message flow

This protocol is a question - answer protocol. The PC/ECR is always master, sends commands and receives responses. POS is always slave, receives commands and sends responses.

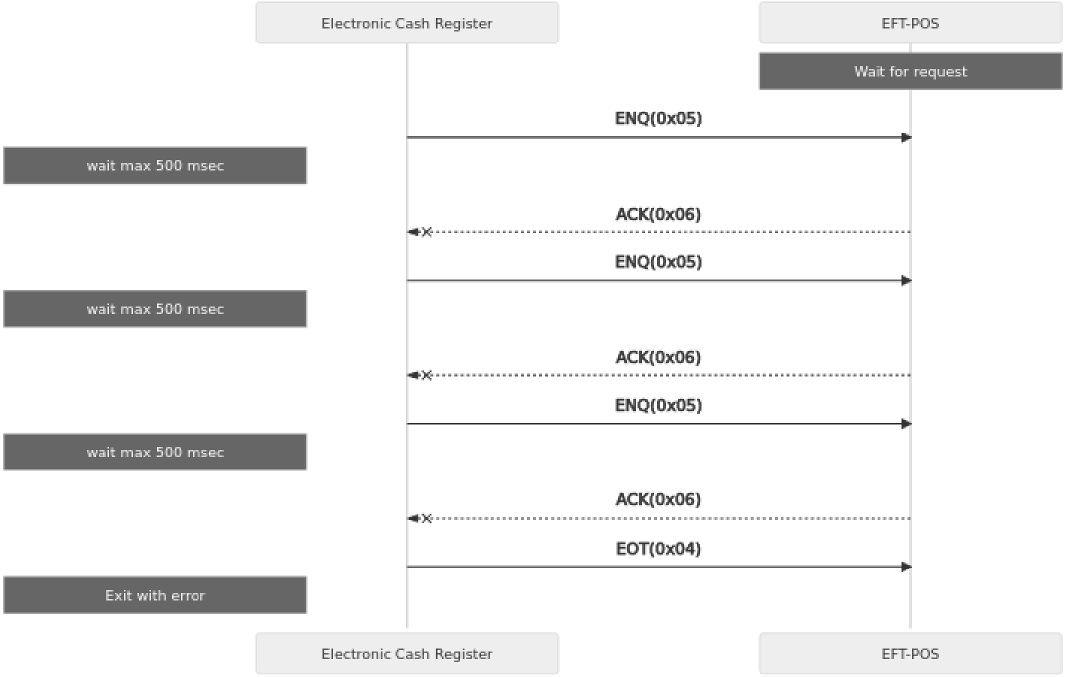
5.1 Login

Communication is started by ECR when it sends ENQ character.

In best scenario, POS responds with ACK character when receives ENQ character from PC/ECR. After this synchronization, POS enters in wait for command state.



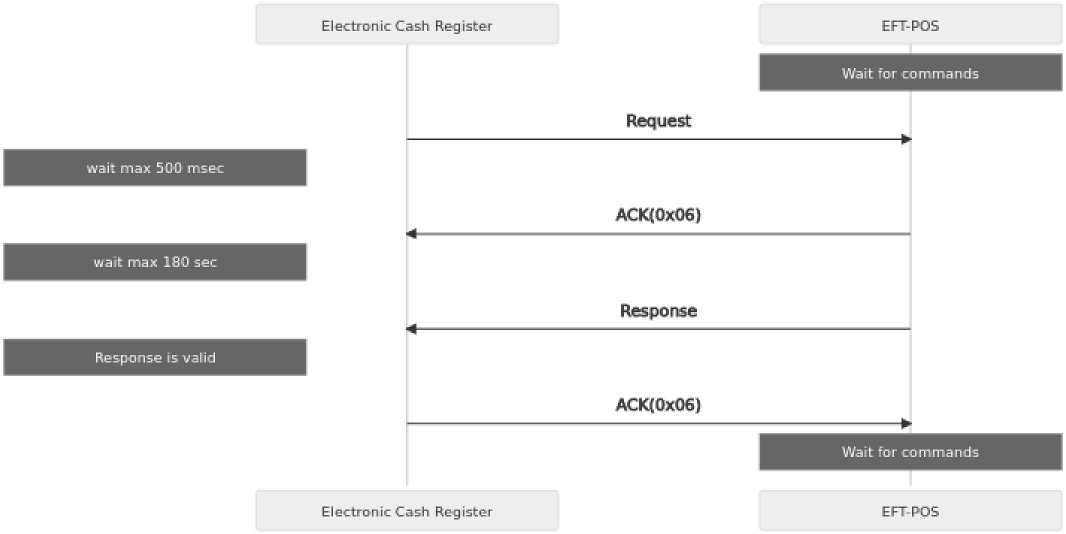
In the worst scenario, POS responds with NAK character when receives ENQ character from PC/ECR. The PC/ECR must send ENQ character again. After sending three times, PC/ECR must abort and sends EOT character.



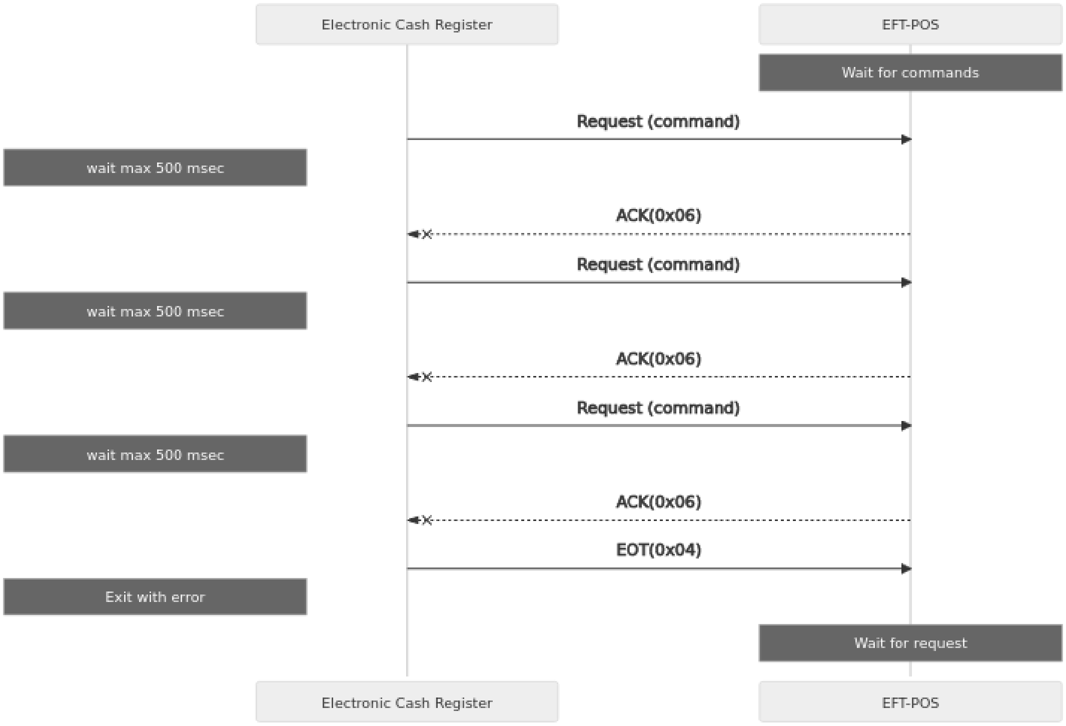
5.2 Command

In command state, POS accepts commands from PC/ECR and replies with responses to PC/ECR.

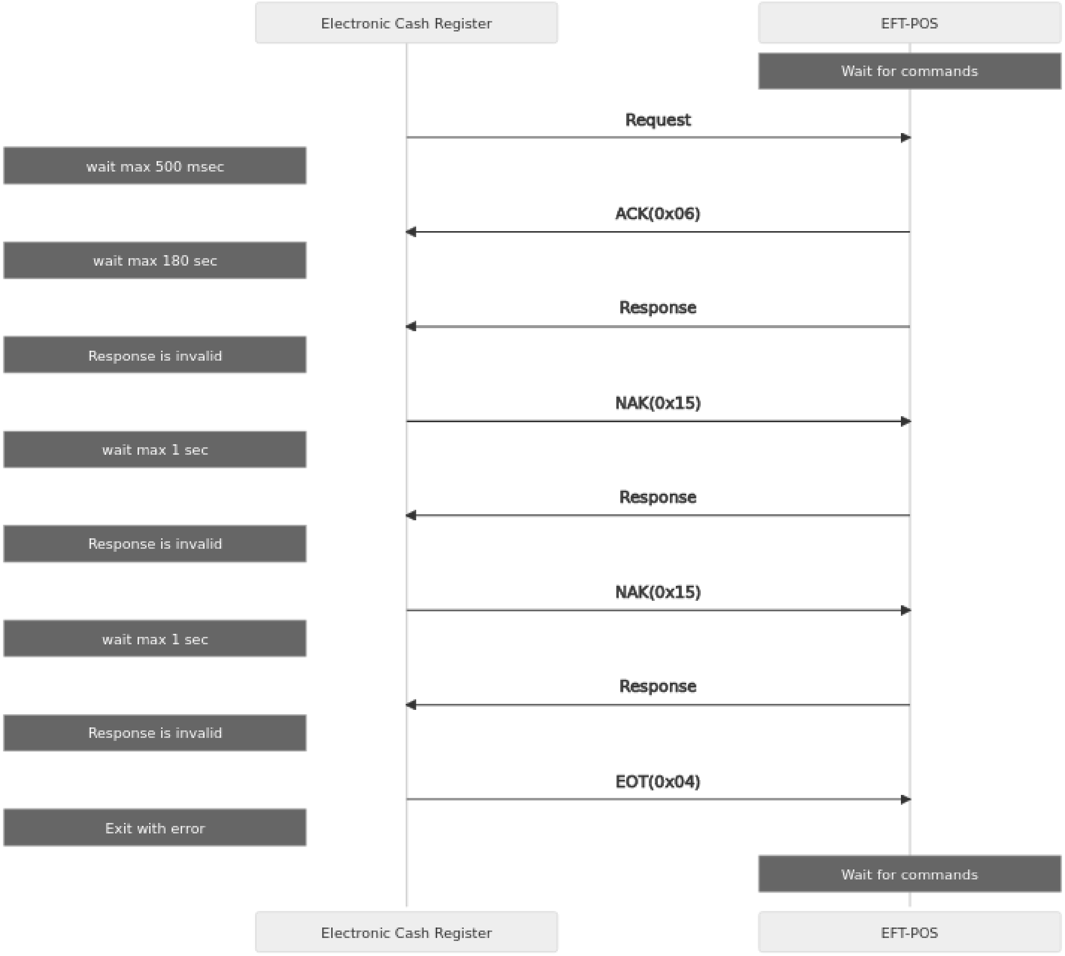
In the best scenario, PC/ECR sends command and POS replies with ACK character and after maximum 180 seconds POS replies with valid response and PC/ECR respond with ACK character.



In the first worst scenario, when PC/ECR send command and POS respond with NAK character or POS do not respond. The PC/ECR must send command again. After sending three times, PC/ECR must abort and sends EOT character.

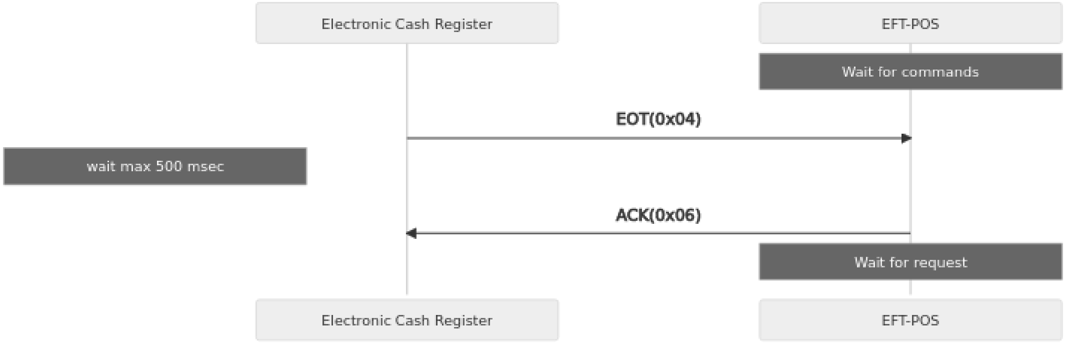


In the second worst scenario, when PC/ECR sends command and POS respond with ACK character and after maximum 180 seconds POS replies with invalid response and PC/ECR respond with NAK character. The POS must reply with response again. After replies three times, PC/ECR must abort and sends EOT character.



5.3 Logout

The communication is closed by ECR when it sends EOT character. POS responds with ACK/NAK when POS receives EOT character from ECR and exits from wait for command state. This sequence is not mandatory.



Appendix 1 – Communication parameter

The EFT-POS can be connected using a native RS232 serial port or virtual serial port through USB cable.

Parameters of serial port connection are:

Parameter	Value
Baud Rate	115200 bps
Data bits	8 bits
Parity	None
Stop bits	1 bit
Flow control	No flow control

Appendix 2 – CRC description

The ECR protocol use CRC-16/BUYPASS (see https://crccalc.com)

C/C++ implementation

```
unsigned short CalculateCRC(unsigned char* a_szBufuer, short a_sBufferLen)
{
    unsigned short usCRC = 0;
    for (short j = 0; j < a_sBufferLen; j++)
    {
        unsigned char* pucPtr = (unsigned char *)&usCRC;
        *(pucPtr + 1) = *(pucPtr + 1) ^ *a_szBufuer++;
        for (short i = 0; i <= 7; i++)
        {
            if (usCRC & ((unsigned short) 0x8000))
            {
                usCRC = usCRC << 1;
                usCRC = usCRC ^ ((unsigned short) 0x8005);
            }
            else
            {
                usCRC = usCRC << 1;
            }
        }
    }
    return (usCRC);
}
```

Java implementation

```
public class CRC {

    private static final int CRC_POLYNOM = (int) 0x00008005;

    private static final int CRC_MASK = (int) 0x00008000;

    public static int CalculateCRC(byte[] buffer, int length) {
        int crc = 0x00000000;
        for (int i = 0; i < length; i++) {
            crc ^= (int) ((buffer[i] << 8) & 0x0000FFFF);
            for (int j = 0; j < 8; j++) {
                if ((crc & CRC_MASK) == CRC_MASK) {
                    crc <<= 1;
                    crc = crc ^ CRC_POLYNOM;
                } else {
                    crc <<= 1;
                }
                crc &= 0x0000FFFF;
            }
        }
        return crc;
    }
}
```

VB implementation

```
Public Function CRCCalculator(data() As Byte) As UShort
    Dim crc As UShort = &H0&
    Dim tmp As UShort = &H0&

    Dim i As Long
    Dim j As Long
    For i = 0 To data.Length() – 1
        tmp = data(i) * &H100&
        crc = crc Xor tmp
        For j = 0 To 7
            If (crc And &H8000&) Then
                crc = ((crc * 2) Xor &H8005&) And &HFFFF&
            Else
                crc = (crc * 2) And &HFFFF&
            End If
        Next j
    Next i
    CRCCalculator = crc And &HFFFF
End Function
```

C# implementation

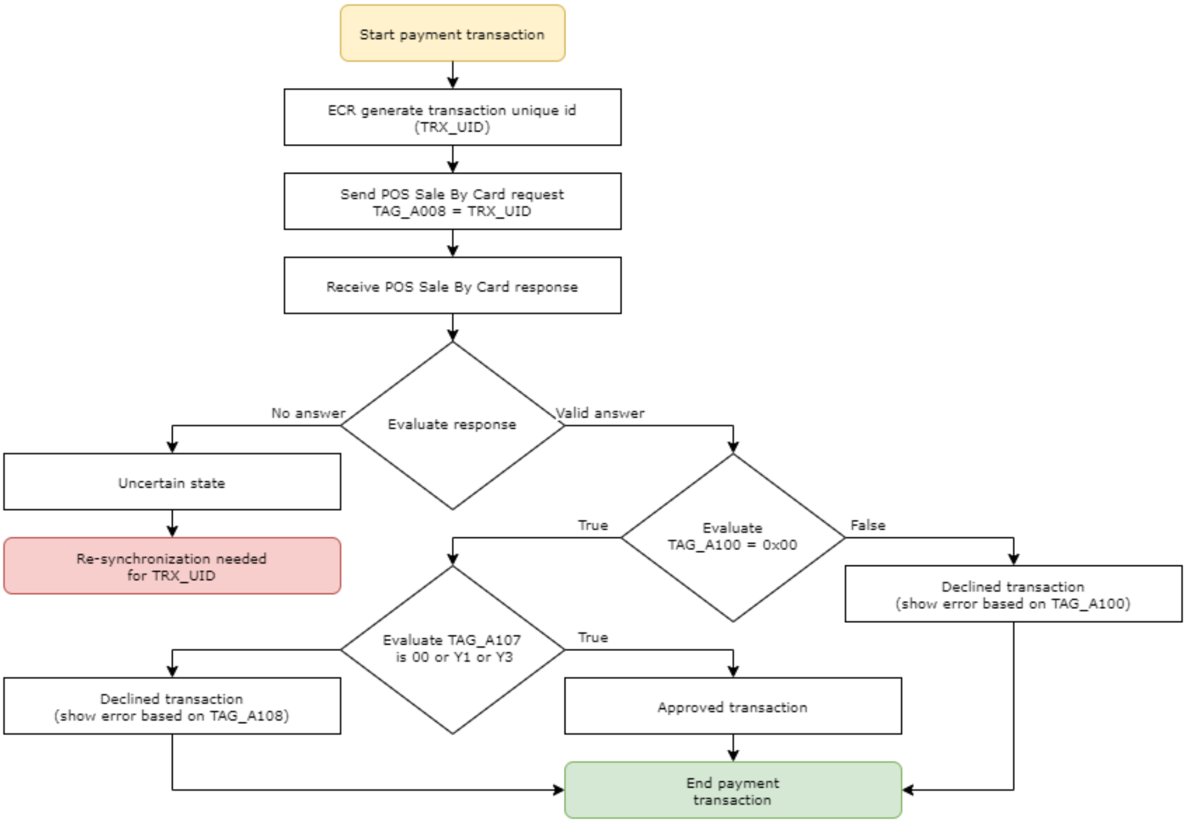
```
public static ushort CRCCalculator(byte[] data)
{
    ushort crc = 0x0000;
    for (var offset = 0; offset < data.Length; offset++)
    {
        crc ^= (ushort)((data[offset] << 8) & 0xFFFF);
        for (var i = 0; i < 8; i++)
        {
            if ((crc & 0x8000) != 0)
            {
                crc <<= 1;
                crc ^= 0x8005;
            }
            else
            {
                crc <<= 1;
            }
        }
    }
    return crc;
}
```

PHP implementation

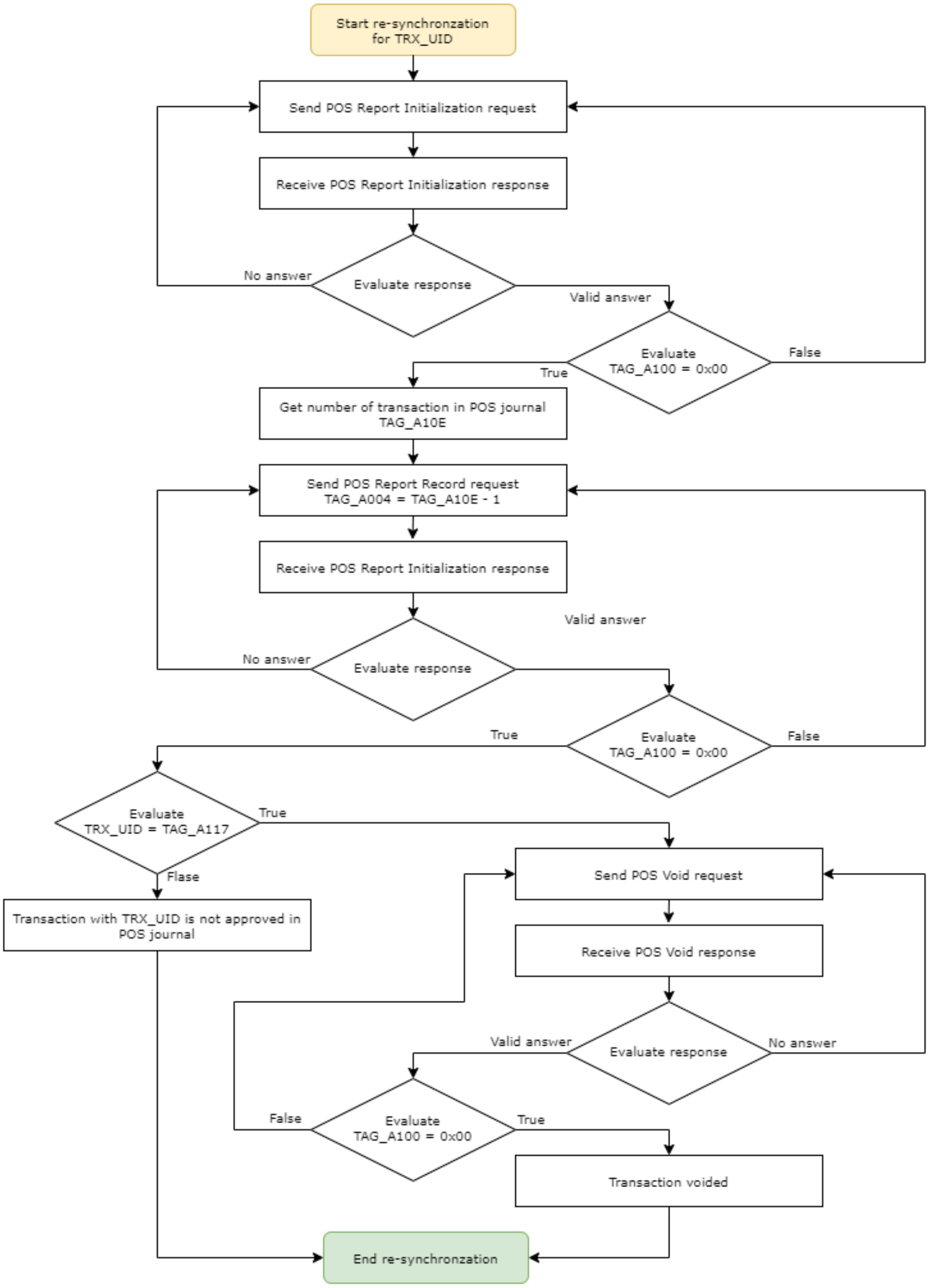
```
function CRC16($buffer) {
    $crc = 0x0000;
    for ($offset = 0; $offset < strlen($buffer); $offset++) {
        $crc ^= (ord($buffer[$offset]) << 8);
        for ($bitwise = 0; $bitwise < 8; $bitwise++) {
            if (($crc <<= 1) & 0x10000) {
                $crc ^= 0x8005;
            }
            $crc &= 0xFFFF;
        }
    }
    return $crc;
}
```

Appendix 3 - Lost connection handling

The following diagram describes payment transaction flow (sale, preauthorization, sale completion, void) and when re-synchronization is needed between ECR and payment terminal.


Start payment transaction
ECR generate transaction unique id (TRX_UID)
Send POS Sale By Card request TAG_A008 = TRX_UID
Receive POS Sale By Card response
Evaluate response
No answer
Valid answer
Uncertain state
Re-synchronization needed for TRX_UID
Evaluate TAG_A100 = 0x00
True
False
Declined transaction (show error based on TAG_A100)
Evaluate TAG_A107 is 00 or Y1 or Y3
True
Declined transaction (show error based on TAG_A108)
Approved transaction
End payment transaction

The following diagram describes how to handle re-synchronization between ECR and payment terminal.


Start re-synchronzation for TRX_UID
Send POS Report Initialization request
Receive POS Report Initialization response
No answer
Evaluate response
Valid answer
Evaluate TAG_A100 = 0x00
False
True
Get number of transaction in POS journal TAG_A10E
Send POS Report Record request TAG_A004 = TAG_A10E - 1
Receive POS Report Initialization response
Valid answer
No answer
Evaluate response
True
Evaluate TAG_A100 = 0x00
False
Evaluate TRX_UID = TAG_A117
True
Flase
Send POS Void request
Transaction with TRX_UID is not approved in POS journal
Receive POS Void response
Valid answer
Evaluate response
No answer
False
Evaluate TAG_A100 = 0x00
True
Transaction voided
End re-synchronzation

Appendix 4 – Install Windows Driver for Ingenico terminals

The driver is compatible with the following versions of Microsoft Windows operating system:

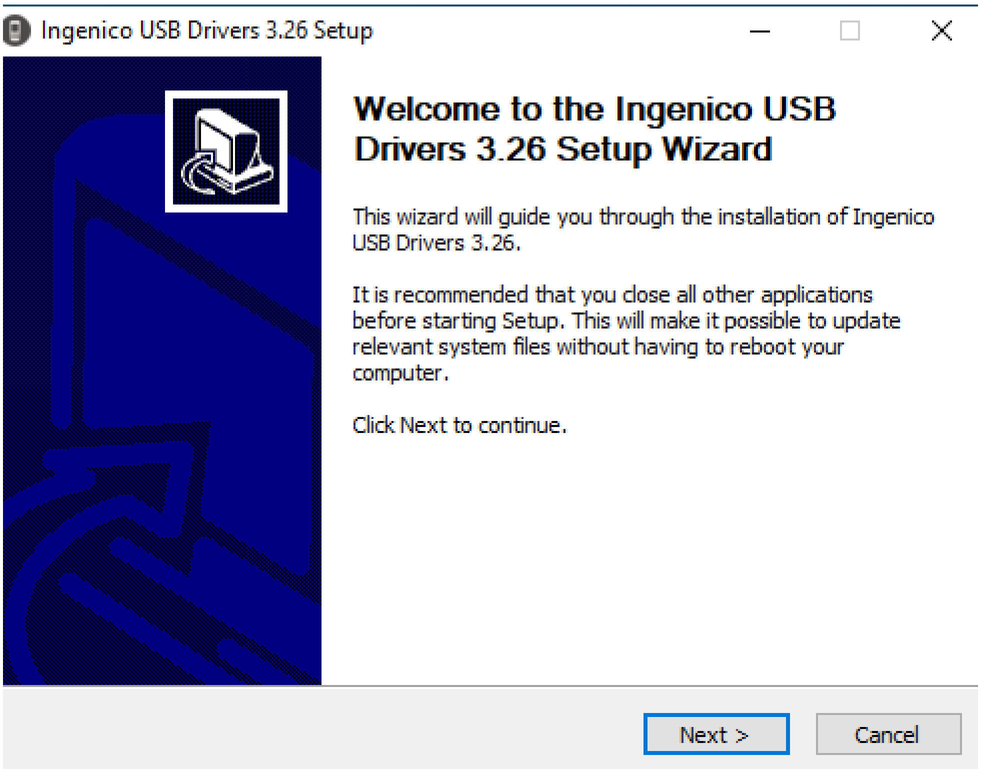
- Windows Vista 32-bits or 64-bits
- Windows 7 32-bits or 64-bits
- Windows 8/8.1 32-bits or 64-bits
- Windows 10 32-bits or 64-bits
- Windows Server 2012
- Windows Server 2016

Notes: The user must have administrator level access to install or uninstall the driver correctly. The installer will uninstall all previous versions of the drivers. This operation may require a reboot of the operating system. Do not connect Ingenico terminal during driver installation.

Notes: The driver can be downloaded from the address below (archive password is **driver**).

https://www.smartpay-software.ro/IngenicoUSBDrivers_3.32_setup_signed.7z

This installer can be launched with the graphical interface by double-clicking on the executable file. The initial screen will confirm which version of the driver will be installed.


Ingenico USB Drivers 3.26 Setup

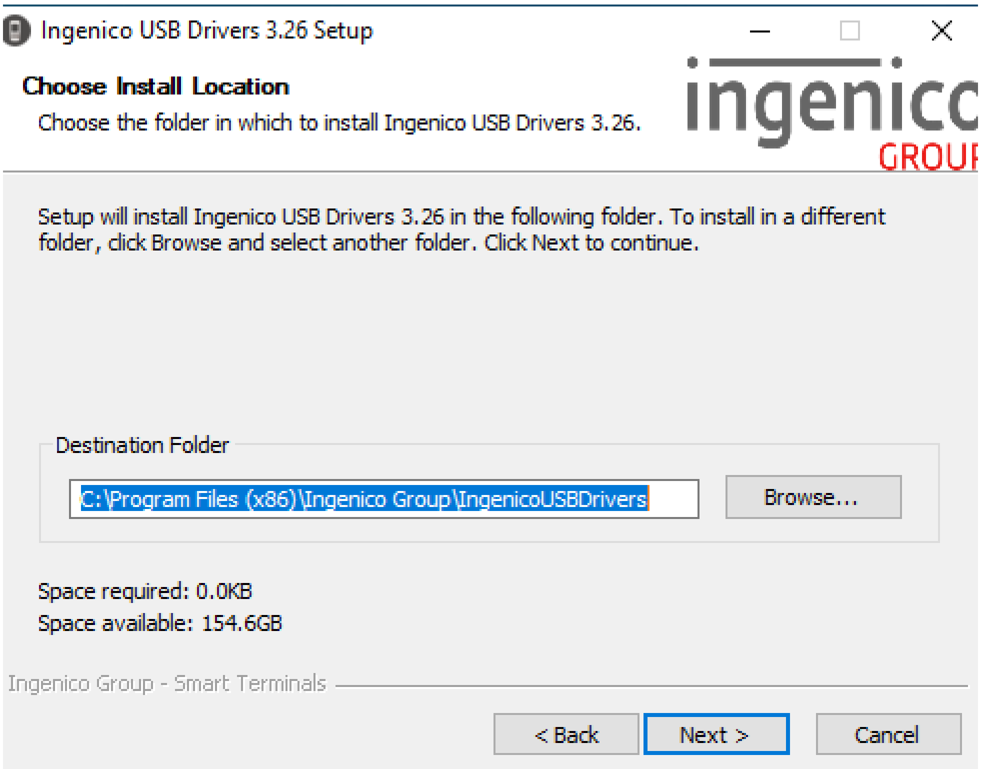
Welcome to the Ingenico USB Drivers 3.26 Setup Wizard

This wizard will guide you through the installation of Ingenico USB Drivers 3.26.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Click **Next >** to start the installation process.



Ingenico USB Drivers 3.26 Setup

Choose Install Location

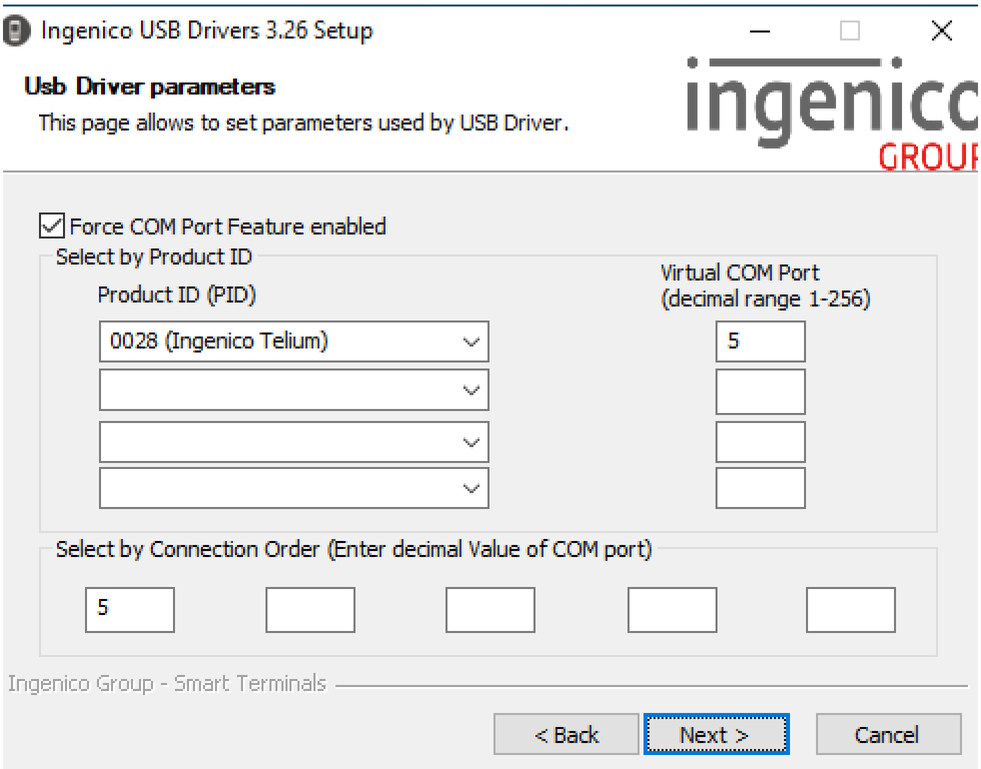
Choose the folder in which to install Ingenico USB Drivers 3.26.

Setup will install Ingenico USB Drivers 3.26 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: C:\Program Files (x86)\Ingenico Group\IngenicoUSBDrivers

Space required: 0.0KB
Space available: 154.6GB

To install the driver to a specific location click **Browse** to search for the desired location, otherwise just click **Next >** to accept the default location specified.



Ingenico USB Drivers 3.26 Setup

Usb Driver parameters

This page allows to set parameters used by USB Driver.

☑ Force COM Port Feature enabled

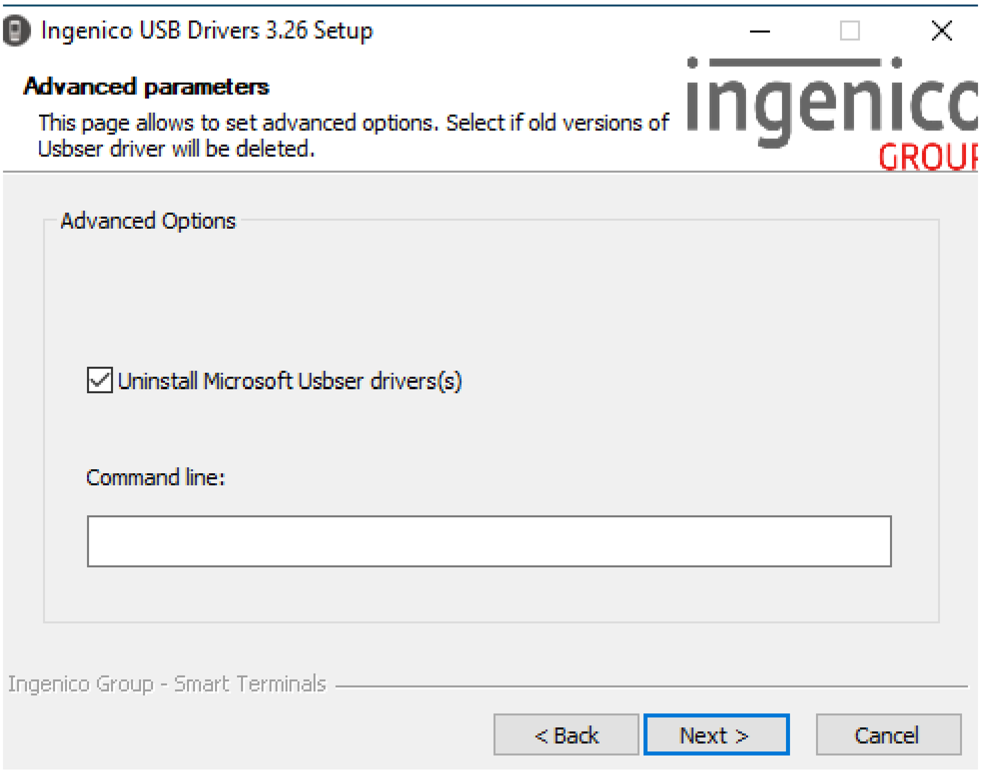
Select by Product ID

Product ID (PID)	Virtual COM Port (decimal range 1-256)
0028 (Ingenico Telium)	5

Select by Connection Order (Enter decimal Value of COM port)

5				

To enable the Force COM Port feature, check the Force COM Port Feature enabled option and complete the configuration as per your requirements (terminal type/family and desired serial port number). Once completed click **Next >**.



Ingenico USB Drivers 3.26 Setup

Advanced parameters

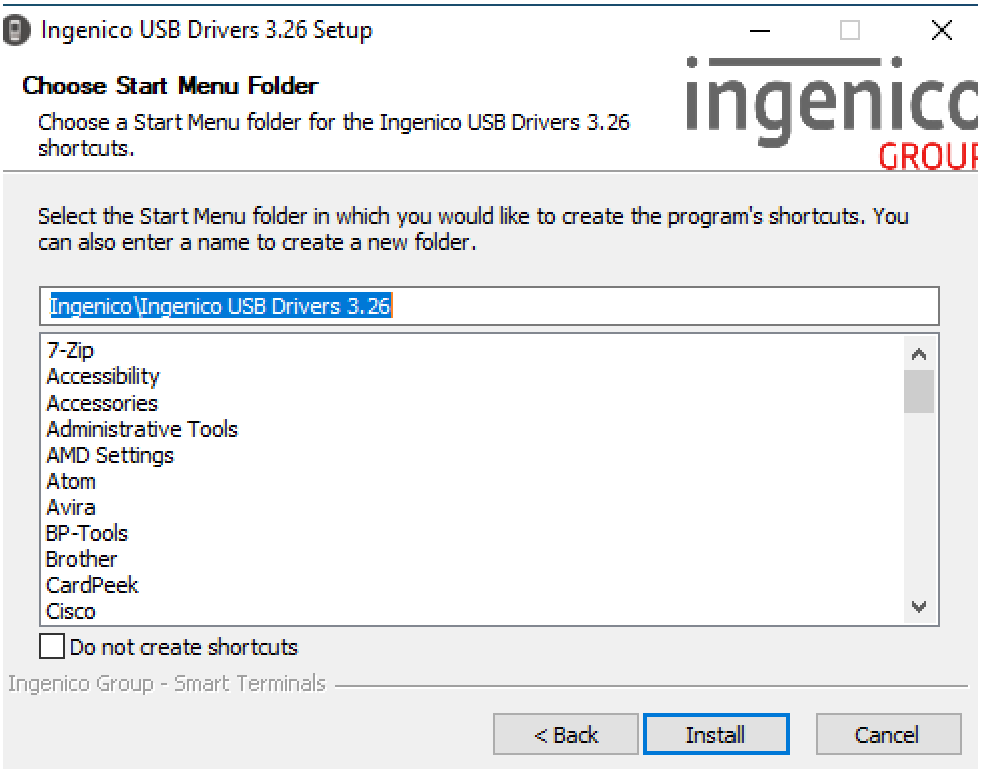
This page allows to set advanced options. Select if old versions of Usbser driver will be deleted.

Advanced Options

☑ Uninstall Microsoft Usbser drivers(s)

Command line:

To ensure that use the Ingenico driver check the option **Uninstall Microsoft Usbser driver(s)** to uninstall the **Microsoft USB driver (usbser)**. If you have other devices that require the Microsoft USB driver uncheck the **Uninstall Microsoft Usbser driver(s)** option before clicking **Next >**.



Ingenico USB Drivers 3.26 Setup

Choose Start Menu Folder

Choose a Start Menu folder for the Ingenico USB Drivers 3.26 shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

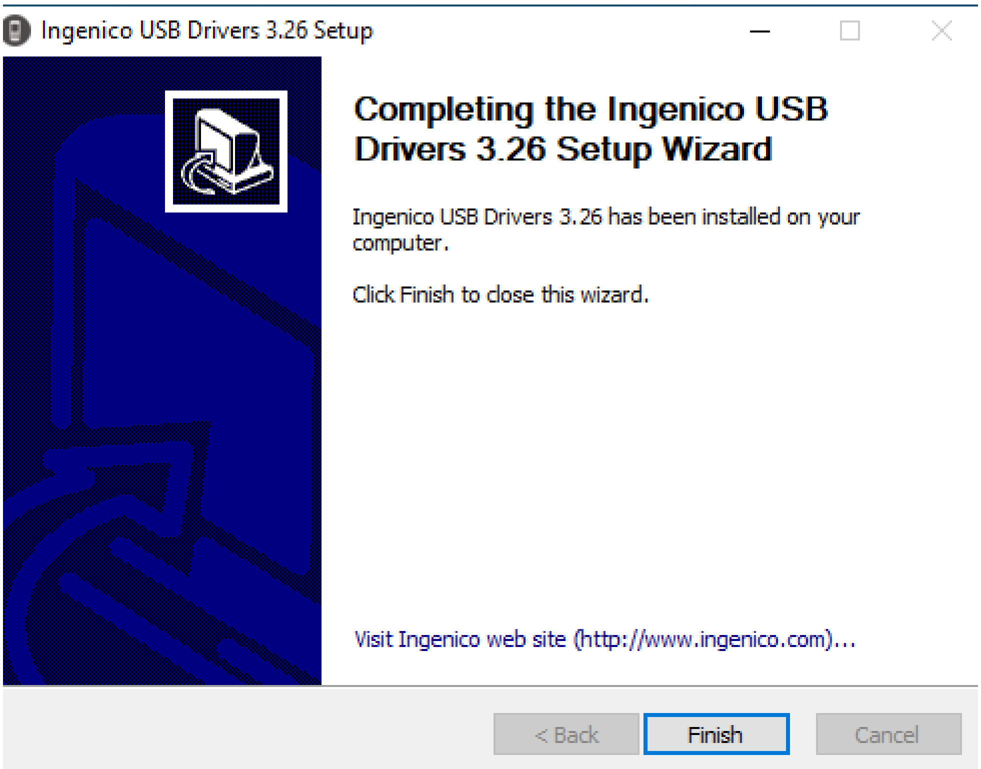
Ingenico\Ingenico USB Drivers 3.26

- 7-Zip
- Accessibility
- Accessories
- Administrative Tools
- AMD Settings
- Atom
- Avira
- BP-Tools
- Brother
- CardPeek
- Cisco

□ Do not create shortcuts

Ingenico Group - Smart Terminals

Choose the start menu folder into which shortcuts will be placed or check the **Do not create shortcuts** option before clicking **Install** to start the installation.



Ingenico USB Drivers 3.26 Setup

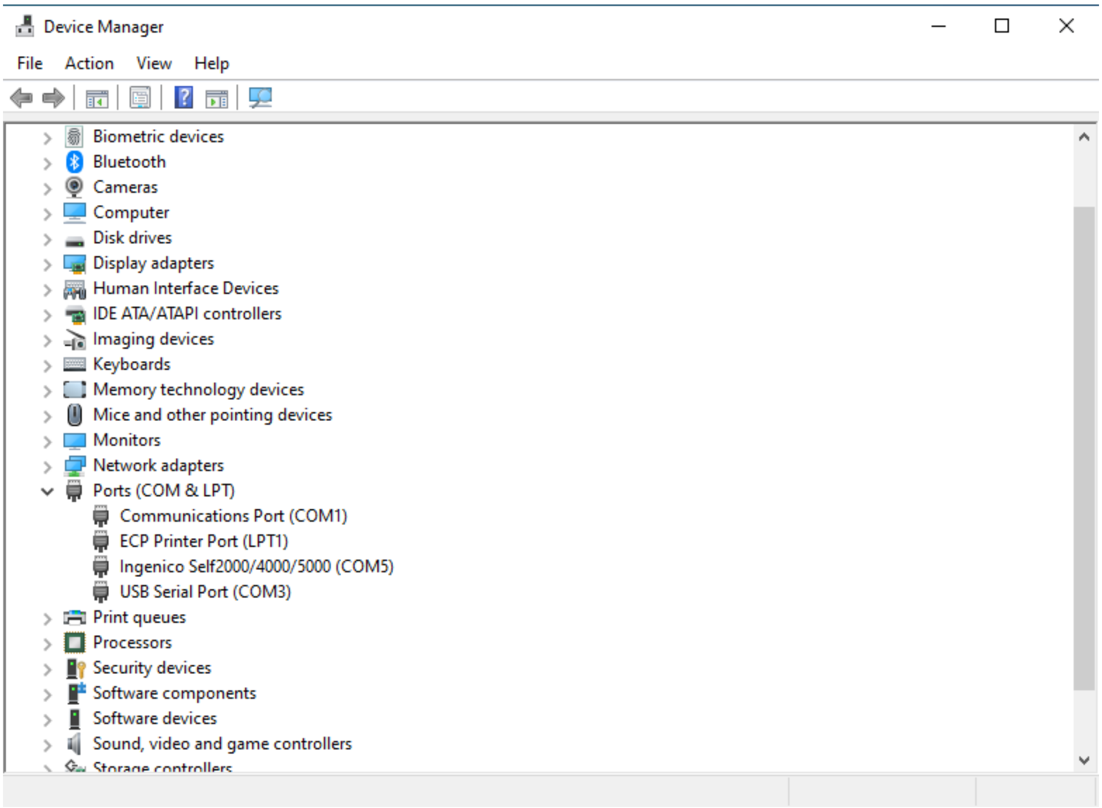
Completing the Ingenico USB Drivers 3.26 Setup Wizard

Ingenico USB Drivers 3.26 has been installed on your computer.

Click Finish to close this wizard.

Visit Ingenico web site (http://www.ingenico.com)...

Close the installation wizard by clicking **Finish**.



Connect terminal using USB cable. Open **Device Manager** and verify if Ingenico terminal appear as **Ports (COM & LPT)** device.

Appendix 5 – Examples

Approved sale transaction

```
SEND 1 Byte
05 (ENQ)

RECEIVE 1 Byte
06 (ACK)

SEND 67 Bytes
02 00 3d a0 00 01 02 a0 | 01 0c 30 30 30 30 30 30
30 30 32 34 35 35 a0 02 | 03 52 4f 4e a0 03 03 39
34 36 a0 08 0c 30 30 30 | 30 30 30 30 30 30 30 30
31 a0 07 0c 30 30 30 30 | 30 30 30 30 30 30 30 30
03 8d 4b

-------------------------------------------------------------------------------
02 (STX)
00 3d (data length)
a0 00 01 02
    (TAG = A000, LEN = 1, VALUE = 02) // Sale operation
a0 01 0c 30 30 30 30 30 30 30 30 32 34 35 35
    (TAG = A001, LEN = 12, VALUE = 000000002455) // Amount 24,55
a0 02 03 52 4f 4e
    (TAG = A002, LEN = 3, VALUE = RON) // Currency name
a0 03 03 39 34 36
    (TAG = A003, LEN = 3, VALUE = 946) // Currency code
a0 08 0c 30 30 30 30 30 30 30 30 30 30 30 31
    (TAG = A008, LEN = 12, VALUE = 000000000001) // Unique ID optional
a0 07 0c 30 30 30 30 30 30 30 30 30 30 30 30
    (TAG = A007, LEN = 12, VALUE = 000000000000) // Cash back amount 0,00 optional
03 (ETX)
8d 4b (CRC16)
-------------------------------------------------------------------------------

RECEIVE 1 Byte
06 (ACK)

RECEIVE 230 Bytes
02 00 e0 a1 00 01 00 a1 | 03 08 54 45 54 52 41 30
30 31 a1 04 08 54 45 54 | 52 41 30 30 31 a1 05 0e
32 30 31 38 30 37 30 35 | 31 36 31 30 35 32 a1 06
0c 30 30 30 30 30 30 30 | 30 32 34 35 35 a1 07 02
30 30 a1 08 10 54 52 41 | 4e 5a 2e 20 41 43 43 45
50 54 41 54 41 a1 09 06 | 30 30 30 30 35 36 a1 0f
06 30 30 30 30 30 31 a1 | 0a 0c 39 36 33 35 35 37
37 31 31 30 34 39 a1 0b | 06 31 32 38 34 36 30 a1
0c 06 2a 2a 33 33 33 37 | a1 0d 1a 43 41 52 44 20
30 32 20 56 49 53 41 20 | 41 43 51 55 49 52 45 52
20 54 45 53 54 a1 13 0b | 56 49 53 41 20 43 52 45
44 49 54 a1 14 0e 41 30 | 30 30 30 30 30 30 30 33
```

```
31 30 31 30 a1 17 0c 30 | 30 30 30 30 30 30 30 30
30 30 31 a1 1b 03 52 4f | 4e a1 1c 03 39 34 36 a1
16 01 09 03 db f2

--------------------------------------------------------------------------------
02 (STX)
00 e0 (data length)
a1 00 01 00
    (TAG = A100, LEN = 1, VALUE = 00) // ECR Success
a1 03 08 54 45 54 52 41 30 30 31
    (TAG = A103, LEN = 8, VALUE = TETRA001)
a1 04 08 54 45 54 52 41 30 30 31
    (TAG = A104, LEN = 8, VALUE = TETRA001)
a1 05 0e 32 30 31 38 30 37 30 35 31 36 31 30 35 32
    (TAG = A105, LEN = 14, VALUE = 20180705161052)
a1 06 0c 30 30 30 30 30 30 30 30 32 34 35 35
    (TAG = A106, LEN = 12, VALUE = 000000002455)
a1 07 02 30 30
    (TAG = A107, LEN = 2, VALUE = 00) // Approved by host
a1 08 10 54 52 41 4e 5a 2e 20 41 43 43 45 50 54 41 54 41
    (TAG = A108, LEN = 16, VALUE = TRANZ. ACCEPTATA)
a1 09 06 30 30 30 30 35 36
    (TAG = A109, LEN = 6, VALUE = 000056)
a1 0f 06 30 30 30 30 30 31
    (TAG = A10F, LEN = 6, VALUE = 000001)
a1 0a 0c 39 36 33 35 35 37 37 31 31 30 34 39
    (TAG = A10A, LEN = 12, VALUE = 963557711049)
a1 0b 06 31 32 38 34 36 30
    (TAG = A10B, LEN = 6, VALUE = 128460)
a1 0c 06 2a 2a 33 33 33 37
    (TAG = A10C, LEN = 6, VALUE = **3337)
a1 0d 1a 43 41 52 44 20 30 32 20 56 49 53 41 20 41 43 51 55 49 52 45 52 20 54 45 5
3 54
    (TAG = A10D, LEN = 26, VALUE = CARD 02 VISA ACQUIRER TEST)
a1 13 0b 56 49 53 41 20 43 52 45 44 49 54
    (TAG = A113, LEN = 11, VALUE = VISA CREDIT)
a1 14 0e 41 30 30 30 30 30 30 30 30 33 31 30 31 30
    (TAG = A114, LEN = 14, VALUE = A0000000031010)
a1 17 0c 30 30 30 30 30 30 30 30 30 30 30 31
    (TAG = A117, LEN = 12, VALUE = 00000000001)
a1 1b 03 52 4f 4e
    (TAG = A11B, LEN = 3, VALUE = RON)
a1 1c 03 39 34 36
    (TAG = A11C, LEN = 3, VALUE = 946)
a1 16 01 09
    (TAG = A116, LEN = 1, VALUE = 0x09)
03 (ETX)
db f2 (CRC16)

SEND 1 Byte
06 (ACK)
```

Declined sale transaction

```
SEND 1 Byte
05 (ENQ)

RECEIVE 1 Byte
06 (ACK)

SEND 67 Bytes
02 00 3d a0 00 01 02 a0 | 01 0c 30 30 30 30 30 30
30 30 31 35 34 30 a0 02 | 03 52 4f 4e a0 03 03 39
34 36 a0 08 0c 30 30 30 | 30 30 30 30 30 30 30 30
32 a0 07 0c 30 30 30 30 | 30 30 30 30 30 30 30 30
03 35 fa

--------------------------------------------------------------------------------
02 (STX)
00 3d (data length)
a0 00 01 02
    (TAG = A000, LEN = 1, VALUE = 02) // Sale operation
a0 01 0c 30 30 30 30 30 30 30 30 31 35 34 30
    (TAG = A001, LEN = 12, VALUE = 000000001540) // Amount 15,40
a0 02 03 52 4f 4e
    (TAG = A002, LEN = 3, VALUE = RON) // Currency name
a0 03 03 39 34 36
    (TAG = A003, LEN = 3, VALUE = 946) // Currency code
a0 08 0c 30 30 30 30 30 30 30 30 30 30 30 32
    (TAG = A008, LEN = 12, VALUE = 000000000002) // Unique ID optional
a0 07 0c 30 30 30 30 30 30 30 30 30 30 30 30
    (TAG = A007, LEN = 12, VALUE = 000000000000) // Cash back amount 0,00 optional
03 (ETX)
35 fa (CRC)
--------------------------------------------------------------------------------

RECEIVE 1 Byte
06 (ACK)

RECEIVE 209 Bytes
02 00 cb a1 00 01 01 a1 | 03 08 54 45 54 52 41 30
30 31 a1 04 08 54 45 54 | 52 41 30 30 31 a1 05 0e
32 30 31 38 30 37 30 35 | 31 36 31 34 34 36 a1 06
0c 30 30 30 30 30 30 30 | 30 31 35 34 30 a1 07 02
5a 33 a1 08 13 54 52 41 | 4e 5a 41 43 54 49 45 20
52 45 53 50 49 4e 53 41 | a1 09 06 30 30 30 30 35
37 a1 0f 06 30 30 30 30 | 30 31 a1 0c 06 2a 2a 33
33 33 37 a1 0d 1a 43 41 | 52 44 20 30 32 20 56 49
53 41 20 41 43 51 55 49 | 52 45 52 20 54 45 53 54
a1 13 0b 56 49 53 41 20 | 43 52 45 44 49 54 a1 14
0e 41 30 30 30 30 30 30 | 30 30 33 31 30 31 30 a1
17 0c 30 30 30 30 30 30 | 30 30 30 30 32 a1 1b
03 52 4f 4e a1 1c 03 39 | 34 36 a1 16 01 09 03 b0
5e
```

```

--------------------------------------------------------------------------------
02 (STX)
00 cb (data length)
a1 00 01 01
    (TAG = A100, LEN = 1, VALUE = 01) // ECR Error
a1 03 08 54 45 54 52 41 30 30 31
    (TAG = A103, LEN = 8, VALUE = TETRA001)
a1 04 08 54 45 54 52 41 30 30 31
    (TAG = A104, LEN = 8, VALUE = TETRA001)
a1 05 0e 32 30 31 38 30 37 30 35 31 36 31 34 34 36
    (TAG = A105, LEN = 14, VALUE = 20180705161446)
a1 06 0c 30 30 30 30 30 30 30 30 31 35 34 30
    (TAG = A106, LEN = 12, VALUE = 000000001540)
a1 07 02 5a 33
    (TAG = A107, LEN = 2, VALUE = Z3) // Declined by chip card
a1 08 13 54 52 41 4e 5a 41 43 54 49 45 20 52 45 53 50 49 4e 53 41
    (TAG = A108, LEN = 16, VALUE = TRANZACTIE RESPINSA)
a1 09 06 30 30 30 30 35 37
    (TAG = A109, LEN = 6, VALUE = 000057)
a1 0f 06 30 30 30 30 30 31
    (TAG = A10F, LEN = 6, VALUE = 000001)
a1 0c 06 2a 2a 33 33 33 37
    (TAG = A10C, LEN = 6, VALUE = **3337)
a1 0d 1a 43 41 52 44 20 30 32 20 56 49 53 41 20 41 43 51 55 49 52 45 52 20 54 45 5
3 54
    (TAG = A10D, LEN = 26, VALUE = CARD 02 VISA ACQUIRER TEST)
a1 13 0b 56 49 53 41 20 43 52 45 44 49 54
    (TAG = A113, LEN = 11, VALUE = VISA CREDIT)
a1 14 0e 41 30 30 30 30 30 30 30 30 33 31 30 31 30
    (TAG = A114, LEN = 14, VALUE = A0000000031010)
a1 17 0c 30 30 30 30 30 30 30 30 30 30 30 32
    (TAG = A117, LEN = 12, VALUE = 00000000001)
a1 1b 03 52 4f 4e
    (TAG = A11B, LEN = 3, VALUE = RON)
a1 1c 03 39 34 36
    (TAG = A11C, LEN = 3, VALUE = 946)
a1 16 01 09
    (TAG = A116, LEN = 1, VALUE = 0x09)
03 (ETX)
b0 5e (CRC)
--------------------------------------------------------------------------------

SEND 1 Byte
06 (ACK)
```

Sale transaction aborted by user

```
SEND 1 Byte
05 (ENQ)

RECEIVE 1 Byte
```

```
06 (ACK)

SEND 67 Bytes
02 00 3d a0 00 01 02 a0 | 01 0c 30 30 30 30 30 30
30 30 34 35 36 37 a0 02 | 03 52 4f 4e a0 03 03 39
34 36 a0 08 0c 30 30 30 | 30 30 30 30 30 30 30 30
33 a0 07 0c 30 30 30 30 | 30 30 30 30 30 30 30 30
03 9e 3f

--------------------------------------------------------------------------------
02 (STX)
00 3d (data length)
a0 00 01 02
    (TAG = A000, LEN = 1, VALUE = 02) // Sale operation
a0 01 0c 30 30 30 30 30 30 30 30 34 35 36 37
    (TAG = A001, LEN = 12, VALUE = 000000004567) // Amount 45,67
a0 02 03 52 4f 4e
    (TAG = A002, LEN = 3, VALUE = RON) // Currency name
a0 03 03 39 34 36
    (TAG = A003, LEN = 3, VALUE = 946) // Currency code
a0 08 0c 30 30 30 30 30 30 30 30 30 30 30 33
    (TAG = A008, LEN = 12, VALUE = 000000000003) // Unique ID optional
a0 07 0c 30 30 30 30 30 30 30 30 30 30 30 30
    (TAG = A007, LEN = 12, VALUE = 000000000000) // Cash back amount 0,00 optional
03 (ETX)
9e 3f (CRC16)
--------------------------------------------------------------------------------

RECEIVE 1 Byte
06 (ACK)

RECEIVE 118 Bytes
02 00 70 a1 00 01 09 a1 | 03 08 54 45 54 52 41 30
30 31 a1 04 08 54 45 54 | 52 41 30 30 31 a1 05 0e
32 30 31 38 30 37 30 35 | 31 36 31 36 32 38 a1 06
0c 30 30 30 30 30 30 30 | 30 34 35 36 37 a1 09 06
30 30 30 30 35 37 a1 0f | 06 30 30 30 30 30 31 a1
0c 02 2a 2a a1 17 0c 30 | 30 30 30 30 30 30 30 30
30 30 33 a1 1b 03 52 4f | 4e a1 1c 03 39 34 36 a1
16 01 24 03 86 fb

--------------------------------------------------------------------------------
02 (STX)
00 70 (data length)
a1 00 01 09
    (TAG = A100, LEN = 1, VALUE = 09) // Canceled by user
a1 03 08 54 45 54 52 41 30 30 31
    (TAG = A103, LEN = 8, VALUE = TETRA001)
a1 04 08 54 45 54 52 41 30 30 31
    (TAG = A104, LEN = 8, VALUE = TETRA001)
a1 05 0e 32 30 31 38 30 37 30 35 31 36 31 36 32 38
    (TAG = A105, LEN = 14, VALUE = 20180705161628)
a1 06 0c 30 30 30 30 30 30 30 30 34 35 36 37
    (TAG = A106, LEN = 12, VALUE = 000000004567)
a1 09 06 30 30 30 30 35 37
```

```
    (TAG = A109, LEN = 6, VALUE = 000057)
a1 0f 06 30 30 30 30 30 31
    (TAG = A10F, LEN = 6, VALUE = 000001)
a1 0c 02 2a 2a
    (TAG = A10C, LEN = 2, VALUE = **) // No card inserted
a1 17 0c 30 30 30 30 30 30 30 30 30 30 30 33
    (TAG = A117, LEN = 12, VALUE = 000000000003)
a1 1b 03 52 4f 4e
    (TAG = A11B, LEN = 3, VALUE = RON)
a1 1c 03 39 34 36
    (TAG = A11C, LEN = 3, VALUE = 946)
a1 16 01 24
    (TAG = A116, LEN = 1, VALUE = 24)
03 (ETX)
86 fb (CRC16)
--------------------------------------------------------------------------------

SEND 1 Byte
06 (ACK)
```