Apex Payment Solutions

Serial ECR & DLL Integration
Apex SmartPOS

May 11, 2023
2.04

PROPRIETARY NOTICE AND LIABILITY DISCLAIMER

Table of Contents

Definitions

Abbreviations

ECR	Electronic Cash Register
POS	Point of Sale\SmartPOS
APDU	Application Protocol Data Unit (Request or Response Data)
CRC	Cyclic Redundancy Check
var	Variable Length
REQ	Request
RSP	Response

Special Character in Transfer Protocol:

STX	0x02	Start of Text
ETX	0x03	End of Text
ACK	0x06	Acknowledgment
NAK	0x15	Negative Acknowledgment
ENQ	0x05	Enquiry
EOT	0x04	End of transmission
DLE	0x10	Data link escape

1. Introduction

1.1. Purpose

The purpose of this document is to provide a technical (message structure and data elements) and customization based on Apex-ECR protocol required for ECR-POS machine to integrate with Apex SmartPOS financial application.

1.2. Audience

This document is primarily for the ECR-POS application developer for implementing Apex-ECR protocol to integrate with Apex SmartPOS financial application using Low-Level Serial protocol and\or using High-Level DLL.

1.3. Overview

ECR machine will be integrated with A90 SmartPOS using Apex Low-Level ECR interface, to initiate different financial and inquiry transactions. A90 SmartPOS application will handle all transactions with the authorization Host.

2. Transport Protocol

The transport protocol defines the lower-level data link protocol that moves the messages\packets between ECR and Apex SmartPOS terminals.

The supported communication protocol is based on a standard asynchronous serial communication protocol (RS232 serial communication), to transfer packets (Application Protocol Data Unit-APDU) encoded as JSON data, the message format illustrates how the data (APDU) is encapsulated within the transport protocol.

2.1. Interface Parameters

The data-transfer is carried-out according to RS232C with the following parameters (115200,8,N,1):

- 115200 Baud Rate, asynchronous.
- No Handshake.
- 8 data-bits.
- No parity-bit.
- 1 Stop-bit.

2.2. Asynchronous Packet Framing

Character-based framing, uses special communication control characters for idle fill and to indicate the beginning and ending of frames. Before transmission, sender should encapsulate the message inside framing control characters **STX** & **ETX** and should append **CRC** checksum at the end of the packet. The information to be transferred (hereafter referred to as APDU) is always packed in the following block structure:

STX	APDU			ETX	CRC
0x02	**Protocol**	**Length**	**Data Block**	0x03	0xXX
	0x01	0xLLLL	JSON data		

Data messages contain checksums to verify the integrity of the APDU data. Checksums work in such a way that if a single bit of the data is corrupted, the checksum would have a different value, so they can provide an inexpensive way to check for (probable) signal integrity. If a message is received with an invalid checksum, the receiver can know that some information was corrupted.

2.3. Retry & Timers

The receiver immediately sends a positive or negative acknowledgement to each received message (before processing the information contained in the APDU). The acknowledgement only confirms that the message was received error-free or not. Error-free means: the message is formally correct constructed and the CRC is also correct.

Positive Acknowledgement: The receiver sends an ACK (0x06)
Negative Acknowledgement: The receiver sends a NAK (0x15)

Sender should wait for **ACK** for each transmitted message to ensure packet delivery by the receiver. In case of no **ACK** received within "**Timer T1**", or in case of receiving **NAK**, sender should retransmit the message again. Retry should not exceed "**Retry Counter**":

Timer (T1):	**Timeout used to acknowledge receiving packets (wait-time between blocks and acknowledgement with ACK/NAK)** The receiver must immediately acknowledge a received message to transmitter with ACK or NAK. The wait-time between reception of the message and transmission of the acknowledgement must not exceed T1 seconds. If this time-out is reached, the message is repeated by the transmitter.	5 seconds
Timer (T2):	**Timeout used to wait for the response (wait-time between ECR Request & Terminal Response)** After sending the request to the terminal, the sender must wait for the final response. The wait-time must not exceed T2 seconds.	60 seconds
Retry Counter:	Counter used to retry sending the message in case of failure	3 retries

2.4. CRC Calculation

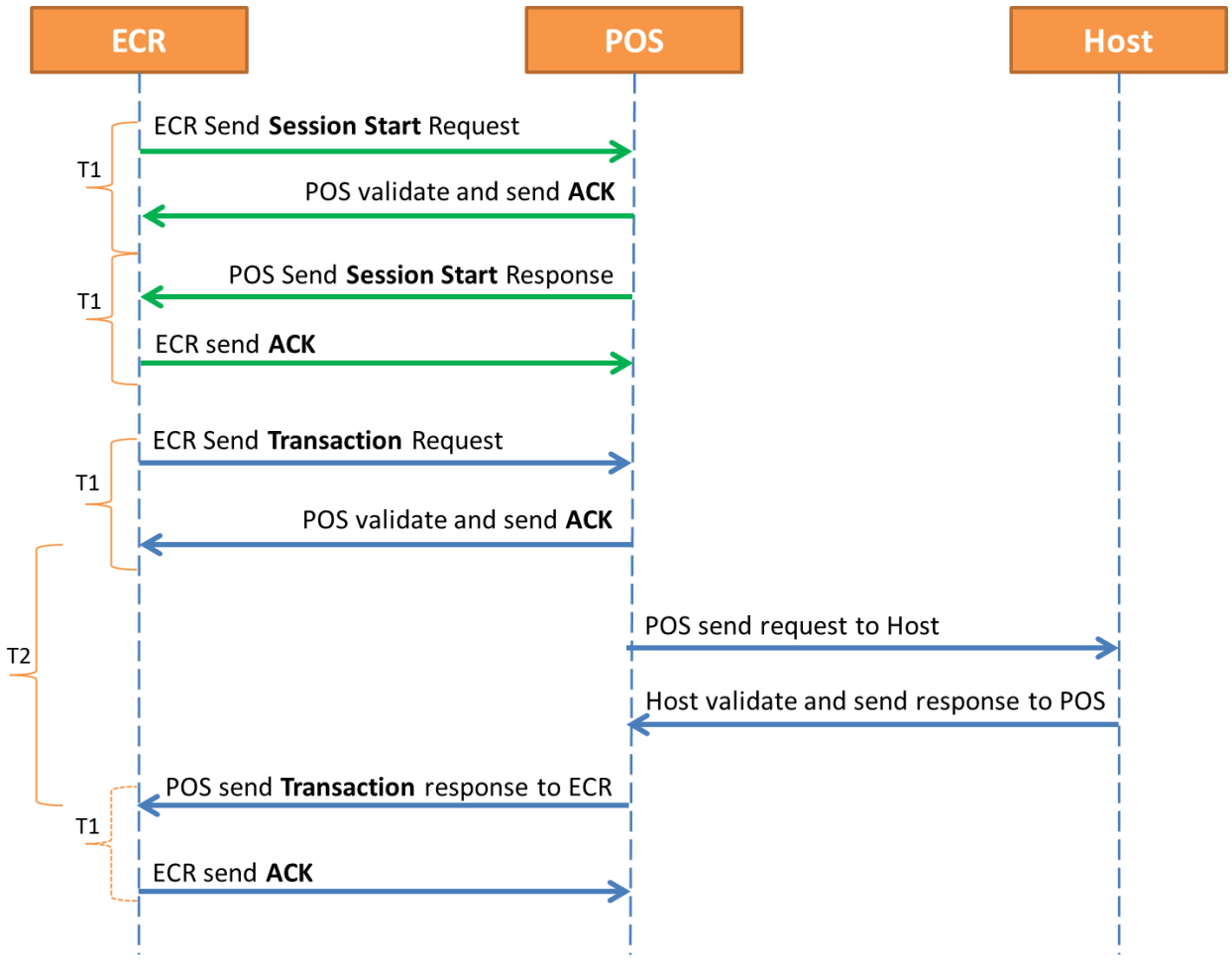
The Cyclic Redundancy Check (CRC-checksum) is calculated by XORing each byte of the APDU and including **ETX**. All characters from the **APDU** plus **ETX** are used in the checksum calculation. **STX** is not used in the CRC-calculation

2.5. Error Handling

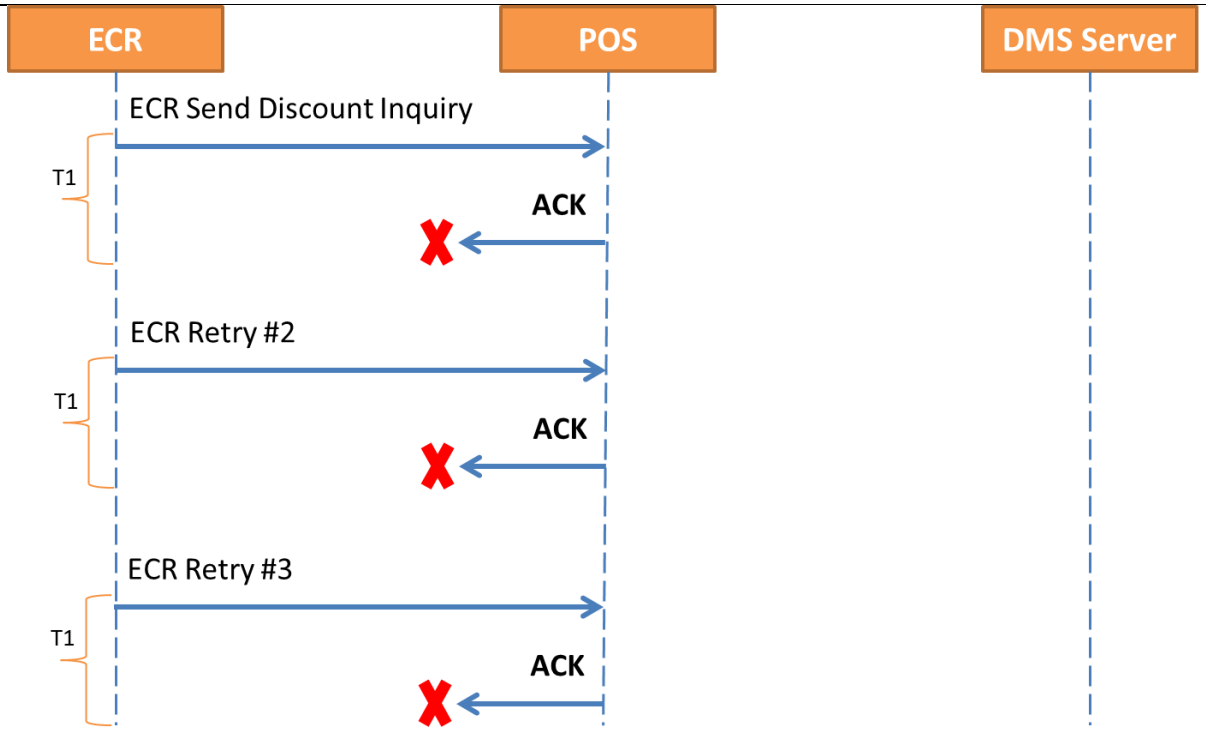
If the receiver responds to a message with NAK, or if time-outs T1 or T2 apply, the transmitter repeats the message up to **Retry-Counter** times. Following this, if still no valid message has been received (NAK, Time-out T1 or Time-out T2), then both communications-partners report a transmission-error to the application level.

2.6. ECR Transaction Flow

Following illustrates the transaction flow between ECR & POS terminal:



Following illustrates the retry after T1 expiry with no ACK received:



3. Application Protocol

The data\information (APDU) transmitted between POS & ECR is constructed as:

Protocol	Length	Data Block
0x01	0xLLLL	JSON data
1 Byte	*2 Bytes*	*data with a length of 0 to 65535 bytes*

3.1. Protocol Field

The protocol value indicates the subsequent information to follow, as:

Protocol	Description
0x01	The data is formatted as JSON object.
Others	Reserved for future use (RFU).

3.2. Length Field

The length field contains the actual length of the directly adjacent data-block, represented in Binary format (0x0000 – 0xFFFF), i.e. (0 - 65535) bytes.

3.3. Data Block Field

The data block field contains the information to be transmitted between ECR and POS terminal. The format of the data depends on the value of the Protocol field.

3.3.1. JSON Data

The data format of the Data Block is formatted using the **JavaScript Object Notation (JSON)** standard.

JSON (JavaScript Object Notation) is an open-standard file format, lightweight data-interchange format, that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value). It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

4. Messages Specifications

4.1. Start Session

The Start Session transaction is used by ECR before any other transactions to check the communication availability with POS terminal.

Tag	Value				
Field Name	**Format**	**Length**	**REQ**	**RSP**	**Description**
"msg"	String	var	M	M	Message Type: "**START**": Start Session
sendMirrorMsg	String	1	O		"Y": POS to send a mirroring messages. Else: disabled.
"tid"	String	8	M	M	Terminal ID
"mid"	String	15	O	O	Merchant ID
"key"	String	var	O		Secure Key
"datetime"	String	14	M		Date & Time "yyyyMMddHHmmss"
"curr"	String	3	M		Currency Code "400"
"username"	String	var	O		Tiller Username
"Fullname"	String	var	O		Tiller fullname
"ref"	String	var	O		Ecr unique reference Number
"ecrRef"	String	Var	O		Echo ECR Reference Number "ref"
"code"	String	2		M	Response code
"description"	String	var		O	Processing errors description
M: *Mandatory* ***O:*** *Optional*					

4.2. Sale

The sale transaction is used to obtain authorization for a financial transaction.

Tag	Value				
Field Name	**Format**	**Length**	**REQ**	**RSP**	**Description**
msg	String	var	M	M	Message Type: "**SALE**": Sale Transaction
sendMirrorMsg	String	1	O		"Y": POS to send a mirroring messages. Else: disabled.
tid	String	8	M	M	Terminal ID
mid	String	15	O	M	Merchant ID
key	String	var	O		Secure Key
printPosReceipt	Integer	1	O		0: None 1: Merchant Copy 2: Customer Copy 3: Both
datetime	String	14	M		Date & Time "yyyyMMddHHmmss"
username	String	50	M		Tiller Username
fullname	String	var	O		Tiller Full Name
amt	Decimal	13	M	O	REQ: Transaction Amount (Sale Amount) RSP: Final authorized Amount, absent in case of error.
curr	String	3	M		Currency Code (400: JOD)
ref	String	var	M	O	REQ: ECR Reference Number RSP: POS Reference Number
"ecrRef"	String	var	O		Echo ECR Reference Number "ref"
cardSlot	Integer			O	Card Entry Slot
pan	String	var		O	Card PAN, absent in case of error. *Sample: 539526******9063*
issuerName	String	var		O	Issuer Name3
cardtype	String	1		O	L: local F: foreign
clientName	String	Var		O	Client Name
isDCC	String	Var		O	True False
hostTid	String	8		O	Host Terminal ID
dcc	object			O	DCC data, see below description
emv	Object			O	EMV data, see below description
hostMid	String	15		O	Host Merchant ID
Serial	String	Var		O	POS Serial Number
inv	String	var		O	POS Receipt Invoice Number
batchNum	String	var		O	Batch Number
rrn	String	var		O	Host Retravel Reference Number
authCode	String	var		O	Host Authorization Number
isApproved	Boolean			O	Transaction approval result (true, false)
isReversal	Boolean			O	Reversal Transaction (true, false)
isVoided	Boolean			O	Voided Transaction (true, false)
isOffline	Boolean			O	Offline transaction (true, false)
code	String	2		M	Response code

description	`String`	`var`		O	Processing errors description
extra	`String`	`Var`		O	Extra data
***M:** Mandatory* ***O:** Optional*					

Tag: DCC	**Value**				
Field Name	**Format**	**Length**	**REQ**	**RSP**	**Description**
indicator	`String`	`var`		O	DCC indicator
chca	`String`	`var`		O	Cardholder currency alphanumeric code
chcn	`String`	`var`		O	Cardholder currency numeric code
comm	`String`	`var`		O	commission
markup	`String`	`var`		O	Markup rate
chta	`String`	`var`		O	Cardholder total amount
exch	`String`	`var`		O	Exchange rate
***M:** Mandatory* ***O:** Optional*					

Tag: EMV	**Value**				
Field Name	**Format**	**Length**	**REQ**	**RSP**	**Description**
aid	`String`	`var`		O	EMV App AID
tvr	`String`	`var`		O	EMV TVR
appName	`String`	`var`		O	EMV App Name
***M:** Mandatory* ***O:** Optional*					

4.3. Refund

The Refund transaction is used to perform refund transaction on POS.

Tag	Value				
Field Name	**Format**	**Length**	**REQ**	**RSP**	**Description**
msg	String	var	M	M	Message Type: "**REFUND**": Sale Transaction
sendMirrorMsg	String	1	O		"Y": POS to send a mirroring messages. Else: disabled.
tid	String	8	M	M	Terminal ID
mid	String	15	O	M	Merchant ID
key	String	var	O		Secure Key
printPosReceipt	Integer	1	O		0: None 1: Merchant Copy 2: Customer Copy 3: Both
datetime	String	14	M		Date & Time "yyyyMMddHHmmss"
username	String	50	M		Tiller Username
fullname	String	var	O		Tiller Full Name
amt	Decimal	13	M	O	REQ: Transaction Amount (Sale Amount) RSP: Final authorized Amount, absent in case of error.
curr	String	3	M		Currency Code (400: JOD)
ref	String	var	M	O	REQ: ECR Reference Number RSP: POS Reference Number
"ecrRef"	String	var	O		Echo ECR Reference Number "ref"
cardSlot	Integer			O	Card Entry Slot
pan	String	var		O	Card PAN, absent in case of error. *Sample: 539526******9063*
issuerName	String	var		O	Issuer Name3
cardtype	String	1		O	L: local F: foreign
clientName	String	Var		O	Client Name
isDCC	String	Var		O	True False
hostTid	String	8		O	Host Terminal ID
dcc	object			O	DCC data, see below description
emv	Object			O	EMV data, see below description
hostMid	String	15		O	Host Merchant ID
Serial	String	Var		O	POS Serial Number
inv	String	var		O	POS Receipt Invoice Number
batchNum	String	var		O	Batch Number
rrn	String	var		O	Host Retravel Reference Number
authCode	String	var		O	Host Authorization Number
isApproved	Boolean			O	Transaction approval result (true, false)
isReversal	Boolean			O	Reversal Transaction (true, false)
isVoided	Boolean			O	Voided Transaction (true, false)
isOffline	Boolean			O	Offline transaction (true, false)
code	String	2		M	Response code

description	`String`	`var`		O	Processing errors description
extra	`String`	`Var`		O	Extra data
***M:** Mandatory* ***O:** Optional*					

Tag: DCC	**Value**				
Field Name	**Format**	**Length**	**REQ**	**RSP**	**Description**
indicator	`String`	`var`		O	DCC indicator
chca	`String`	`var`		O	Cardholder currency alphanumeric code
chcn	`String`	`var`		O	Cardholder currency numeric code
comm	`String`	`var`		O	commission
markup	`String`	`var`		O	Markup rate
chta	`String`	`var`		O	Cardholder total amount
exch	`String`	`var`		O	Exchange rate
***M:** Mandatory* ***O:** Optional*					

Tag: EMV	**Value**				
Field Name	**Format**	**Length**	**REQ**	**RSP**	**Description**
aid	`String`	`var`		O	EMV App AID
tvr	`String`	`var`		O	EMV TVR
appName	`String`	`var`		O	EMV App Name
***M:** Mandatory* ***O:** Optional*					

4.4. Void

This transaction is used to Void a previously approved transaction before the settlement.

Tag	Value				
Field Name	**Format**	**Length**	**REQ**	**RSP**	**Description**
msg	String	var	M	M	Message Type: "**SALE**": Sale Transaction
sendMirrorMsg	String	1	O		"Y": POS to send a mirroring messages. Else: disabled.
tid	String	8	M	M	Terminal ID
mid	String	15	O	M	Merchant ID
key	String	var	O		Secure Key
printPosReceipt	Integer	1	O		0: None 1: Merchant Copy 2: Customer Copy 3: Both
datetime	String	14	M		Date & Time "yyyyMMddHHmmss"
username	String	50	M		Tiller Username
fullname	String	var	O		Tiller Full Name
amt	Decimal	13	M	O	REQ: Transaction Amount (Sale Amount) RSP: Final authorized Amount, absent in case of error.
curr	String	3	M		Currency Code (400: JOD)
ref	String	var	M	O	REQ: ECR Reference Number RSP: POS Reference Number
"ecrRef"	String	var	O		Echo ECR Reference Number "ref"
cardSlot	Integer			O	Card Entry Slot
pan	String	var		O	Card PAN, absent in case of error. *Sample: 539526******9063*
issuerName	String	var		O	Issuer Name3
cardtype	String	1		O	L: local F: foreign
clientName	String	Var		O	Client Name
isDCC	String	Var		O	True False
hostTid	String	8		O	Host Terminal ID
dcc	object			O	DCC data, see below description
emv	Object			O	EMV data, see below description
hostMid	String	15		O	Host Merchant ID
Serial	String	Var		O	POS Serial Number
inv	String	var		O	POS Receipt Invoice Number
batchNum	String	var		O	Batch Number
rrn	String	var		O	Host Retravel Reference Number
authCode	String	var		O	Host Authorization Number
isApproved	Boolean			O	Transaction approval result (true, false)
isReversal	Boolean			O	Reversal Transaction (true, false)
isVoided	Boolean			O	Voided Transaction (true, false)
isOffline	Boolean			O	Offline transaction (true, false)
code	String	2		M	Response code
description	String	var		O	Processing errors description

extra	String	Var		O	Extra data
M: *Mandatory* ***O:*** *Optional*					

4.5. Settlement

Tag	Value				
Field Name	**Format**	**Length**	**REQ**	**RSP**	**Description**
msg	String	var	M	M	Message Type: "**SETTLE**": Settlement Transaction
sendMirrorMsg	String	1	O		"Y": POS to send a mirroring messages. Else: disabled.
Hostname	String	Var	O	O	Acquirer Host Name
tid	String	8	M	M	Terminal ID
mid	String	15	O	M	Merchant ID
key	String	var	O		Secure Key
printPosReceipt	Integer	1	O		0: None 1: Merchant Copy 2: Customer Copy 3: Both
datetime	String	14	M		Date & Time "yyyyMMddHHmmss"
username	String	50	M		Tiller Username
fullname	String		M		Tiller Full Name
curr	String	3	M		Currency Code (400: JOD)
ref	String	var	M	O	REQ: ECR Reference Number RSP: POS Reference Number
"ecrRef"	String	var	O		Echo ECR Reference Number "ref"
Serial	String	Var		O	POS Serial Number
batchNum	String	var		O	Batch Number
isApproved	String	Var		O	Transaction approval result (True, False)
code	String	2		M	Response code
description	String	var		O	Processing errors description
M: *Mandatory* ***O:*** *Optional*					

4.6. Last Transaction Status

Tag	Value				
Field Name	**Format**	**Length**	**REQ**	**RSP**	**Description**
msg	String	var	M	M	Message Type: "**LAST_TXN_STATUS**": Last Transaction STatus
sendMirrorMsg	String	1	O		"Y": POS to send a mirroring messages. Else: disabled.
tid	String	8	M	M	Terminal ID
mid	String	15	O	M	Merchant ID
key	String	var	O		Secure Key
printPosReceipt	Integer	1	O		0: None 1: Merchant Copy 2: Customer Copy 3: Both
datetime	String	14	M		Date & Time "yyyyMMddHHmmss"
username	String	50	M		Tiller Username
fullname	String	var	O		Tiller Full Name
amt	Decimal	13		O	REQ: Transaction Amount (Sale Amount) RSP: Final authorized Amount, absent in case of error.
curr	String	3			Currency Code (400: JOD)
ref	String	var	O	O	If EMPTY, POS will return last transaction status. If present, POS will return specific txn status.
"ecrRef"	String	var	O		Echo ECR Reference Number "ref"
cardSlot	Integer			O	Card Entry Slot
pan	String	var		O	Card PAN, absent in case of error. *Sample: 539526******9063*
issuerName	String	var		O	Issuer Name3
cardtype	String	1		O	L: local F: foreign
clientName	String	Var		O	Client Name
isDCC	String	Var		O	True False
hostTid	String	8		O	Host Terminal ID
dcc	object			O	DCC data, see below description
emv	Object			O	EMV data, see below description
hostMid	String	15		O	Host Merchant ID
Serial	String	Var		O	POS Serial Number
inv	String	var		O	POS Receipt Invoice Number
batchNum	String	var		O	Batch Number
rrn	String	var		O	Host Retravel Reference Number
authCode	String	var		O	Host Authorization Number
isApproved	Boolean			O	Transaction approval result (true, false)
isReversal	Boolean			O	Reversal Transaction (true, false)
isVoided	Boolean			O	Voided Transaction (true, false)
isOffline	Boolean			O	Offline transaction (true, false)
code	String	2		M	Response code
description	String	var		O	Processing errors description

extra	**String**	**Var**		O	Extra data
M: *Mandatory* ***O:*** *Optional*					

5. DLL Interface

A Dynamic-Link Library (DLL) is also provided to simplify the integration with ECR POS retail system. All functionalities are wrapped into one DLL, accessed through "**ApexSerialEcr.dll**".

3.1 DLL Classes

3.1.1 EcrConfig

Member	Type	Max Length	Description
Mid	String	15	Merchant ID
Tid	String	8	Terminal ID
MerchantSecureKey	String	32	Merchant Secure Key
EcrCurrencyCode	String	3	ECR Currency Code ("400")
EcrTillerUserName	String	30	Tiller Username\ID
EcrTillerFullName	String	30	Tiller Full Name
EcrStationName	String	30	ECR Station Name
sendMirrorMsg	String	1	Enable/Disable sending mirror messages

3.1.2 ComPortSettings

Member	Type	Max Length	Description
ComPortName	string		COM Port Name ("COM1")
BaudRate	int		Baud Rate (115200)
Parity	ComParity		Parity Bit: - None - Odd - Even - Mark - Space
DataBits	int		Date Bit (8)
StopBits	ComStopBits		Stop Bits: - None - One - Two - OnePointFive
Handshake	ComHandshake		Handshake

3.1.3 EcrRsp_StartSession

Member	Type	Max Length	Description
code	string	2	Processing Response Code
description	string	var	Description
isApproved	bool	-	Boolean value to indicate if transaction is Approved (true) or Declined (false)
jsonData	string	var	JSON formatted string of the POS response

3.1.4 EcrRsp_Sale

Member	Type	Max Length	Description
code	string	2	Processing Response Code
description	string	var	Description
isApproved	bool	-	Boolean value to indicate if transaction is Approved (true) or Declined (false)
isReversal	bool	-	Boolean value to indicate if transaction is Reversal (true) or Not (false)
isVoided	bool	-	Boolean value to indicate if transaction is Voided (true) or Not (false)
isOffline	bool	-	Boolean value to indicate if transaction is Offline (true) or Not (false)
jsonData	string	var	JSON formatted string of the POS response
cardSlot	int	1	1: Manual PAN Entry. 2: Magnetic Card (Swipe). 3: Chip Card (Card Inserted). 4: Contactless (Tap).
cardType	string	var	"L": Local card "F": Foreign Other: Unknown
pan	string	13-19	First 6 and Last 4 digits of the PAN (Field 2). (XXXXXX******XXXX)
issuerName	string	var	Issuer Name (Visa, MasterCard ..)
hostTid	string	8	Terminal ID
hostMid	String	15	Merchant ID
invoice	string	6	POS Invoice Number
batchNum	string	6	Batch Number
rrn	String	12	Retrieval Reference Number
authCode	string	6	Authorization response number
clientName	string	var	Cardholder\Client name
amt	string	var	Final authorized amount
isDCC	bool	-	Boolean value to indicate if DCC is selected
dcc	DccInfo	-	DCC Info object
emv	EmvInfo	-	EMV Info object

3.1.5 EcrRsp_Refund

Member	Type	Max Length	Description
code	string	2	Processing Response Code
description	string	var	Description
isApproved	bool	-	Boolean value to indicate if transaction is Approved (true) or Declined (false)
isReversal	bool	-	Boolean value to indicate if transaction is Reversal (true) or Not (false)
isVoided	bool	-	Boolean value to indicate if transaction is Voided (true) or Not (false)
isOffline	bool	-	Boolean value to indicate if transaction is Offline (true) or Not (false)
jsonData	string	var	JSON formatted string of the POS response
cardSlot	int	1	1: Manual PAN Entry. 2: Magnetic Card (Swipe). 3: Chip Card (Card Inserted). 4: Contactless (Tap).
cardType	string	var	"L": Local card "F": Foreign Other: Unknown
pan	string	13-19	First 6 and Last 4 digits of the PAN (Field 2). (XXXXXX******XXXX)
issuerName	string	var	Issuer Name (Visa, MasterCard ..)
hostTid	string	8	Terminal ID
hostMid	String	15	Merchant ID
invoice	string	6	POS Invoice Number
batchNum	string	6	Batch Number
rrn	String	12	Retrieval Reference Number
authCode	string	6	Authorization response number
clientName	string	var	Cardholder\Client name
amt	string	var	Final authorized amount
isDCC	bool	-	Boolean value to indicate if DCC is selected
dcc	DccInfo	-	DCC Info object
emv	EmvInfo	-	EMV Info object

3.1.6 EcrRsp_Void

Member	Type	Max Length	Description
code	string	2	Processing Response Code
description	string	var	Description
isApproved	bool	-	Boolean value to indicate if transaction is Approved (true) or Declined (false)
isReversal	bool	-	Boolean value to indicate if transaction is Reversal (true) or Not (false)
isVoided	bool	-	Boolean value to indicate if transaction is Voided (true) or Not (false)
isOffline	bool	-	Boolean value to indicate if transaction is Offline (true) or Not (false)
jsonData	string	var	JSON formatted string of the POS response
cardSlot	int	1	1: Manual PAN Entry. 2: Magnetic Card (Swipe). 3: Chip Card (Card Inserted). 4: Contactless (Tap).
pan	string	13-19	First 6 and Last 4 digits of the PAN (Field 2). (XXXXXX******XXXX)
issuerName	string	var	Issuer Name (Visa, MasterCard ..)
hostTid	string	8	Terminal ID
hostMid	String	15	Merchant ID
Inv	string	6	POS Invoice Number
batchNum	string	6	Batch Number
rrn	String	12	Retrieval Reference Number
authCode	string	6	Authorization response number

3.1.7 EcrRsp_Settle

Member	Type	Max Length	Description
code	string	2	Processing Response Code
description	string	var	Description
isApproved	bool	-	Boolean value to indicate if transaction is Approved (true) or Declined (false)
jsonData	string	var	JSON formatted string of the POS response

3.1.8 EcrRsp_MirroringMsg

Member	Type	Max Length	Description
code	string	2	Processing Response Code
description	string	var	Description
isApproved	bool	-	Boolean value to indicate if transaction is Approved (true) or Declined (false)
jsonData	string	var	JSON formatted string of the POS response
msg	String	var	"MSG_MIRRORING"
title	String	var	Title
text	String	var	Text message

3.1.9 DccInfo

Member	Type	Max Length	Description
indicator	string	1	"Y": DCC selected "N": DCC not selected
chca	string	var	Cardholder Currency Alphanumeric "USD"
chcn	string	var	Cardholder Currency code "400"
comm	string	var	Commission fees
markup	string	var	Markup fees
chta	string	var	Cardholder Amount (in foreign currency)
exch	string	var	Exchange Rate

3.1.10EmvInfo

Member	Type	Max Length	Description
aid	string	var	Application ID "A000000003"
tvr	string	var	Terminal verification results (TVR) or Tag95
appName	string	var	Application Name

3.2 DLL Enums

3.2.1 Ecr_Txns

Member	Value
Unknown	0
START	1
SALE	10
VOID	11
REFUND	13
SETTLE	1000
DISCOUNT	2200

3.3. DLL Functions

3.3.1 General Functions

```
string GetVersion();
```
Returns the current DLL version in a String format.

3.3.2 Session Functions

```
EcrRsp_StartSession StartSession(int timeOut);
Start Session is used prior calling any other functions to validate the connectivity.
```

3.3.3 Financial Functions

```
EcrRsp_Sale Sale(String amount, String ecrReference, String ecrInvocie);
```
Start SALE Transaction.

```
EcrRsp_Sale Refund(String amount, String ecrReference, String ecrInvocie);
```
Start REFUND Transaction.

```
EcrRsp_Void Void(String invoice)
```
Void a transaction by Invoice.

```
EcrRsp_Settle Settle();
```
Initiate a Settlement request.

```
EcrRsp_Common LastTxnStatus(String ecrReference);
```
Get last transaction status.

```
Ecr_Txns LastTxnType(EcrRsp_Common response);
```
Get last transaction type (SALE, REFUND, VOID).

4. Appendix

4.3. Response Codes

Code	Description
"00"	Approved
Others	Declined

4.4. Card Slot

Code	Description
1	Manual Entry
2	Magnetic Stripe (Swipe)
3	ICC (Chip)
4	Contactless
5	QR Code

4.5. Sample Request (Low-Level)

Request	
Raw	02010 0C07B226B6579223A22222C226461746574696D65223A2232303230303130313 23334353539222C22757365726E616D65223A22666C616E222C2266756C6C6E616D65 223A22466C616E20416C466C616E79222C2263757272223A22343030222C22696E762 23A2230303032312222C22726566223A227265665F303031222C22746964223A223131 313131313131222C226D6964223A22313131313131313131313131313131222C226D7 367223A2253414C45222C22616D74223A31302E3132337D03D4

JSON	{
	"key": "",
	"datetime": "20200101234559",
	"username": "flan",
	"fullname": "Flan AlFlany",
	"curr": "400",
	"inv": "00021",
	"ref": "ref_001",
	"tid": "11111111",
	"mid": "111111111111111",
	"msg": "SALE",
	"amt": 10.123
	}

4.6. Sample Code (C#)

Configure & Open COM Port

```
{
    ApexSerialEcrDLL.EcrConfig ecrConfig = new ApexSerialEcrDLL.EcrConfig();
    ecrConfig.Tid = "11111111";
    ecrConfig.Mid = "111111111111111";
    ecrConfig.MerchantSecureKey = "";
    ecrConfig.EcrCurrencyCode = "400";
    ecrConfig.EcrTillerUserName = "flan";
    ecrConfig.EcrTillerFullName = "Flan AlFlany";
    ecrConfig.EcrStationName = "1";

    ApexSerialEcrDLL.ComPortSettings comConfig = new
                                                 ApexSerialEcrDLL.ComPortSettings();
    comConfig.ComPortName = “COM4”;
    comConfig.BaudRate    = 115200;
    comConfig.Parity      = ApexSerialEcrDLL.ComPortSettings.ComParity.None;
    comConfig.DataBits    = 8;
    comConfig.StopBits    = ApexSerialEcrDLL.ComPortSettings.ComStopBits.One;
    comConfig.Handshake   = ApexSerialEcrDLL.ComPortSettings.ComHandshake.None;

    apexEcrDll = new ApexSerialEcrDLL.Interface(comConfig, ecrConfig);

    if (apexEcrDll.Open() == false)
    {
        return;
    }
}
```

Close COM Port

```
apexEcrDll.Close();
```

Sale Transaction

```
apexEcrDll.Sale("1.234", "ref_0001", "inv_0001");
```

Refund Transaction

```
apexEcrDll.Refund("1.234", "ref_0002", "inv_0002");
```

Void Transaction

```
apexEcrDll.Void("000021");
```

Settle Transaction

```
apexEcrDll.Settle(null);
```

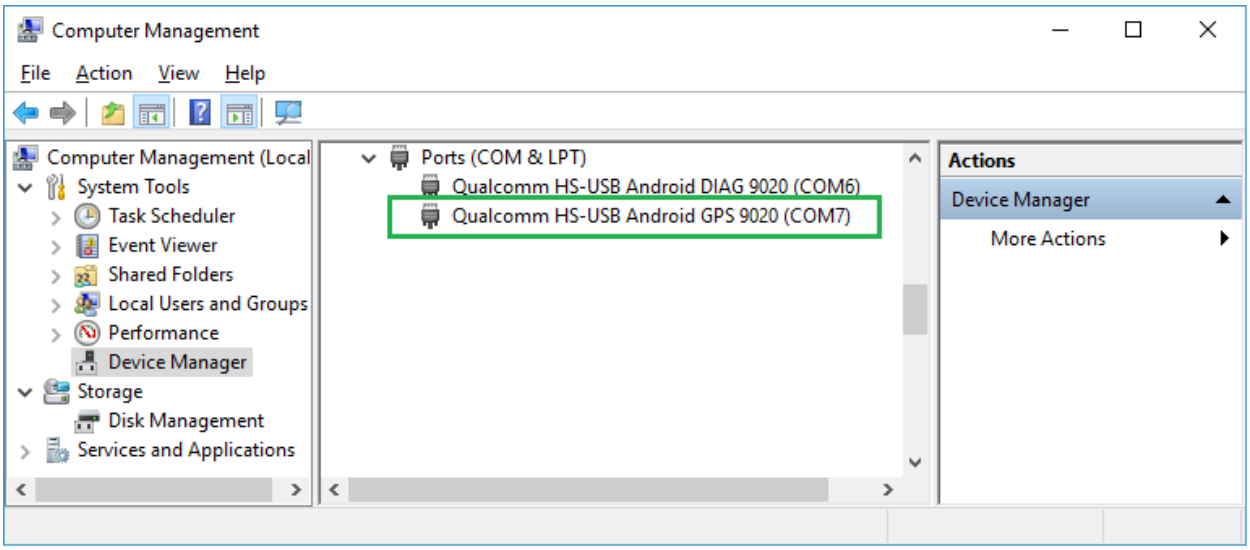
Last Transaction Status

```
EcrRsp_Common rsp = apexEcrDll.LastTxnStatus(null);


Ecr_Txns txnType = apexEcrDll.LastTxnType(rsp);

switch (txnType)
{
	case Ecr_Txns.SALE:
		EcrRsp_Sale rspSale = EcrRsp_Sale.Deserialize(rsp.jsonData);
	break;
	case Ecr_Txns.REFUND:
		EcrRsp_Refund rspRefund = EcrRsp_Refund.Deserialize(rsp.jsonData);
	break;
	case Ecr_Txns.VOID:
		EcrRsp_Void rspVoid = EcrRsp_Void.Deserialize(rsp.jsonData);
	break;
}
```

4.7. A90 COM Port configuration



Apex Payment Solutions

www.apex.jo