POS terminal software

VTK protocol

Index: VTK-PROTOCOL-EN
Version: 1.4
Date: 2019-03-20

Table of Contents

Document Modification History

Version	Date	Description of Changes
1.0	August 25, 2016	Initial version
1.1	September 15, 2016	Protocol applied to a POS terminal
1.2	December 12, 2017	CDP message added
1.3	October 2, 2018	Event registration and Product id added
1.4	March 20, 2019	MFR message and QR-Code data added

1. Introduction

«VTK protocol» (previously known as «VENDOTEK protocol») is an application-level protocol between Point-of-Sale terminal (POS) and Vending Machine Controller (VMC), communicated over serial link or over TCP/IP subnet.

Serial port setting: 115200 bits per second, mode 8N1, no flow control.

Timeout for the last byte receiving after first byte receiving: 8 seconds.

If TCP/IP used, VMC must open TCP-connection to the POS IP-address and TCP port. Recommended port is 62801. Recommended connection timeout is 15 seconds.

2. Message format

Message format for communication over TCP/IP:

Field	Length (bytes)	Description
Length (big-endian)	2	Length of the following data
Protocol discriminator (big-endian)	2	0x96FB - from VMC; 0x97FB - from POS
Application message	0-65533	BER-TLV coded message

Message format for communication over serial link without TCP/IP:

Field	Length (bytes)	Description
Starting byte	1	0x1F
Length (big-endian)	2	Length of the following data, excluding CRC16
Protocol discriminator (big-endian)	2	0x96FB - from VMC; 0x97FB - from POS
Application message	0-65533	BER-TLV coded message
CRC16 (big-endian)	2	CRC16-CCITT (initial value FFFFh)

CRC16-CCITT should be calculated from the starting byte to the last byte of application message. Example code on C Programming Language is specified at the end of this document.

Application message contains set of BER TLV (ISO/IEC 8825-1) parameters with primitive encoding. List of parameters depends on processed operation. Sequence of parameters is free. Unsupported parameters must be ignored, and message analysis must be continued.

Each ISO/IEC 8825-1 parameter has the following format:

Field name	Length (bytes)	Description
Tag	1-3	Parameter identifier, according ISO/IEC 8825-1
Length	1-3	Length of value, according ISO/IEC 8825-1
Value	variable	Value

3. Description of parameters

3.1. List of parameters

Identifier	Description	Format	Maximal length	Source
0x01	Message name	ASCII	3	VMC, POS
0x03	Operation number	Decimal in ASCII	8	VMC, POS
0x04	Amount in minor currency unit	Decimal in ASCII	12	VMC, POS
0x05	Keepalive interval in seconds	Decimal in ASCII	3	POS
0x06	Operation timeout in seconds	Decimal in ASCII	3	POS
0x07	Event name	ASCII	5	VMC, POS
0x08	Event number	Decimal in ASCII	8	VMC, POS
0x09	Product id	Decimal in ASCII	6	VMC, POS
0x0A	QR-Code data	ASCII	No limit	VMC

3.2. Message names

Message name defines a function of the message:

Name	Function	Source
IDL	IDLE state handshake	VMC, POS
DIS	DISABLED state handshake	VMC, POS
STA	Start of session, if required from POS	POS
VRP	Vend request positive	VMC, POS
FIN	Finalization of operation	VMC, POS
ABR	Abort of operation	VMC
CDP	Cash deposit	VMC, POS
MFR	MIFARE card detected	POS

After power-up, VMC and POS are in the INACTIVE state. Normally, VMC starts to send IDL messages. POS sends IDL responses. When first IDL message received, each side moves to the IDLE state.

In IDLE and INACTIVE states, VMC may start to send DIS messages. POS sends DIS responses. When first DIS message received, each side moves to the DISABLE state. POS must not send STA message in DISABLED state. VMC must not send VRP/CDP message in DISABLED state.

In IDLE state, VMC may send VRP message, when buyer has selected a product. POS sends VRP response. In some configurations, product selection allowed only after STA message is received from POS, when buyer have inserted a card or have pressed some “START“ key.

If vend request is approved by the POS, VMC must send FIN message when vend is finished with either success or failure. POS sends FIN response.

If vend request is declined or FIN message is received after approval, VMC must send IDL or DIS message immediately.

In IDLE state, VMC may accept money from a client, then send CDP message.

3.3. Operation number

VMC must get operation number from any message and save it in temporary variable. POS must get operation number from VRP/CDP message and save in non-volatile memory. Default operation number must be zero.

Before sending of VRP/CDP message, VMC must increment operation number and save it in the same variable. After that, VMC must send this value in all messages before the new vending request.

If POS received several VRP/CDP requests or several FIN requests with the same operation number, it treats it as repeats and return previous result.

3.4. Amount in minor currency unit

If amount in STA message is non-zero, it defines maximal amount for VRP requests.

If amount in VRP response is non-zero, it defines amount of approved operation. If amount in VRP response is zero, the financial operation is not approved.

If amount in FIN request is non-zero, it defines amount of successfully completed vending operation. If amount in FIN request is zero, it means vending failure. POS may decline FIN request, if its amount differs from VRP response amount.

If amount in FIN response is the same as in FIN request, the finalization is approved.

If amount in CDP response is zero, VMC must return all money to a client.

3.5. Keepalive interval in seconds

VMC must use received value as a interval of sending IDL messages in IDLE state as well as DIS messages in DISABLED state. Any non-zero value must be got from any message and saved in temporary variable. Zero interval is prohibited. Default value of variable must be 10 seconds at VMC side.

If VMC does not receive any message during 3*Tk+8 seconds, where Tk is keepalive interval, VMC must move to INACTIVE state and must restore default values in keepalive interval and operation timeout variables. If TCP/IP protocol is used, VMC should close TCP connection before moving to INACTIVE state.

For example, if keepalive interval is 30 seconds, and VMC does not receive any message during 3*30+8=98 seconds, VMC must move to INACTIVE state.

3.6. Operation timeout in seconds

VMC must use received value as a timeout between VRP/CDP request and VRP/CDP response, and between FIN request and FIN response. Any non-zero value must be got from any messages and saved in temporary variable. Zero timeout is prohibited. Default value of variable must be 60 seconds at VMC side.

If operation timeout occurs, VMC must move to INACTIVE state and must restore default values in keepalive interval and operation timeout variables. If TCP/IP protocol is used, VMC should close TCP connection before coming to INACTIVE state.

3.7. Event name

The name of the event that previously occurred in the vending machine. The following event names are defined in this document:

Name	Description
CSAPP	Cash Sale Approved
CSDEN	Cash Sale Denied

Other names are reserved for the future and must be ignored. Even if some event name is unknown for POS, event number must be processed as defined in this document.

3.8. Event number

Starting from INACTIVE state, VMC must not sent messages about events before any event number received from POS.

If POS supports event registration, it must send event number in each IDL/DIS message. It must be number of the last received event or “0” (zero) if no events yet received. VMC must get event number from IDL/DIS message and save it in temporary variable. Before sending message about a new event, VMC must increment event number and save it in the same variable. After that, VMC must send this value together with event name in all IDL/DIS messages until the same event number is received from POS.

If POS received several IDL/DIS requests with the same event number, it treats it as repeats. VMC may send IDL/DIS message in IDLE/DISABLED state immediately after new event occurred. However, next IDL/DIS message must be sent after keepalive interval delay.

The queue of events must be implemented in VMC because a new event may occurred before previous event is confirmed by POS.

3.9. Product id

The Product id defined in the vending machine. It may be referred to a vending request (if transmitted in VRP or FIN message) or an event (if transmitted in IDL or DIS message).

3.10. QR-Code data

The QR-Code data for POS to show on the screen. Transmitted in FIN or IDL message. This data does not require a confirmation.

4. Message flow examples

4.1. Normal power-up sequence

VMC event	Message from VMC	Message from POS	POS event
INACTIVE state after power-up			
	IDL --->		
10 seconds delay			INACTIVE state after power-up
	IDL --->		
		<--- IDL	Move to IDLE state
Move to IDLE state			
Keepalive interval delay			
	IDL --->		
		<--- IDL	Still in IDLE state

4.2. IDLE – DISABLED - IDLE sequence

Still in IDLE state			
Disable card reception	DIS --->		
		<--- DIS	Move to DISABLED state
Move to DISABLED state			
Keepalive interval delay			
	DIS --->		
		<--- DIS	Still in DISABLED state
Any delay			
	IDL --->		
		<--- IDL	Move to IDLE state
Move to IDLE state			

4.3. Optional start of operation from POS

			Still in IDLE state
		<--- STA	Card inserted or START button pressed
Maximal amount indication			Still in IDLE state

4.4. Approved operation sequence

Product is selected	VRP --->		
		(<--MFR)	
	(ABR -->)		
			Vend approved
		<--- VRP	
Vend success or failure	FIN --->		
		<--- FIN	
Session completed	IDL --->		
		<--- IDL	Move to IDLE state
Move to IDLE state			

4.5. Declined operation sequence

Product is selected	VRP --->		
		(<--MFR)	
	(ABR -->)		
			Vend declined
		<--- VRP	
Session completed	IDL --->		
		<--- IDL	Move to IDLE state
Move to IDLE state			

4.6. CDP operation

Money is accepted	CDP --->		
	(ABR -->)		
			Operation finished
		<--- CDP	
Session completed	IDL --->		
		<--- IDL	Move to IDLE state
Move to IDLE state			

4.7. Event registration in IDLE state

Still in IDLE state			Still in IDLE state
Some event occurred	IDL --->		
		(<--IDL)	Message with an incorrect event number or no messages at all
Keepalive interval delay			
	IDL --->		
		<--- IDL	Message with a correct event number
Still in IDLE state			Still in IDLE state

5. CRC calculation example

This is example of CRC16-CCITT calculation on the C programming language:

```
static const uint16_t crc16_ccitt_table[ 256 ] =
{
0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50A5, 0x60C6, 0x70E7, 0x8108, 0x9129, 0xA14A,
0xB16B, 0xC18C, 0xD1AD, 0xE1CE, 0xF1EF, 0x1231, 0x0210, 0x3273, 0x2252, 0x52B5, 0x4294,
0x72F7, 0x62D6, 0x9339, 0x8318, 0xB37B, 0xA35A, 0xD3BD, 0xC39C, 0xF3FF, 0xE3DE, 0x2462,
0x3443, 0x0420, 0x1401, 0x64E6, 0x74C7, 0x44A4, 0x5485, 0xA56A, 0xB54B, 0x8528, 0x9509,
0xE5EE, 0xF5CF, 0xC5AC, 0xD58D, 0x3653, 0x2672, 0x1611, 0x0630, 0x76D7, 0x66F6, 0x5695,
0x46B4, 0xB75B, 0xA77A, 0x9719, 0x8738, 0xF7DF, 0xE7FE, 0xD79D, 0xC7BC, 0x48C4,
0x58E5, 0x6886, 0x78A7, 0x0840, 0x1861, 0x2802, 0x3823, 0xC9CC, 0xD9ED, 0xE98E, 0xF9AF,
0x8948, 0x9969, 0xA90A, 0xB92B, 0x5AF5, 0x4AD4, 0x7AB7, 0x6A96, 0x1A71, 0x0A50, 0x3A33,
0x2A12, 0xDBFD, 0xCBDC, 0xFBBF, 0xEB9E, 0x9B79, 0x8B58, 0xBB3B, 0xAB1A, 0x6CA6,
0x7C87, 0x4CE4, 0x5CC5, 0x2C22, 0x3C03, 0x0C60, 0x1C41, 0xEDAE, 0xFD8F, 0xCDEC,
0xDDCD, 0xAD2A, 0xBD0B, 0x8D68, 0x9D49, 0x7E97, 0x6EB6, 0x5ED5, 0x4EF4, 0x3E13,
0x2E32, 0x1E51, 0x0E70, 0xFF9F, 0xEFBE, 0xDFDD, 0xCFFC, 0xBF1B, 0xAF3A, 0x9F59,
0x8F78, 0x9188, 0x81A9, 0xB1CA, 0xA1EB, 0xD10C, 0xC12D, 0xF14E, 0xE16F, 0x1080, 0x00A1,
0x30C2, 0x20E3, 0x5004, 0x4025, 0x7046, 0x6067, 0x83B9, 0x9398, 0xA3FB, 0xB3DA, 0xC33D,
0xD31C, 0xE37F, 0xF35E, 0x02B1, 0x1290, 0x22F3, 0x32D2, 0x4235, 0x5214, 0x6277, 0x7256,
0xB5EA, 0xA5CB, 0x95A8, 0x8589, 0xF56E, 0xE54F, 0xD52C, 0xC50D, 0x34E2, 0x24C3,
0x14A0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405, 0xA7DB, 0xB7FA, 0x8799, 0x97B8, 0xE75F,
0xF77E, 0xC71D, 0xD73C, 0x26D3, 0x36F2, 0x0691, 0x16B0, 0x6657, 0x7676, 0x4615, 0x5634,
0xD94C, 0xC96D, 0xF90E, 0xE92F, 0x99C8, 0x89E9, 0xB98A, 0xA9AB, 0x5844, 0x4865, 0x7806,
0x6827, 0x18C0, 0x08E1, 0x3882, 0x28A3, 0xCB7D, 0xDB5C, 0xEB3F, 0xFB1E, 0x8BF9,
0x9BD8, 0xABBB, 0xBB9A, 0x4A75, 0x5A54, 0x6A37, 0x7A16, 0x0AF1, 0x1AD0, 0x2AB3,
0x3A92, 0xFD2E, 0xED0F, 0xDD6C, 0xCD4D, 0xBDAA, 0xAD8B, 0x9DE8, 0x8DC9, 0x7C26,
0x6C07, 0x5C64, 0x4C45, 0x3CA2, 0x2C83, 0x1CE0, 0x0CC1, 0xEF1F, 0xFF3E, 0xCF5D,
0xDF7C, 0xAF9B, 0xBFBA, 0x8FD9, 0x9FF8, 0x6E17, 0x7E36, 0x4E55, 0x5E74, 0x2E93,
0x3EB2, 0x0ED1, 0x1EF0,
};
uint16_t get_crc( const uint8_t* data, uint16_t size )
{
uint16_t i, tmp, crc = 0xffff;
for ( i = 0; i < size; i++ )
{
tmp = ( crc >> 8 ) ^ ( 0x00ff & data[ i ] );
crc = ( crc << 8 ) ^ crc16_ccitt_table[ tmp ];
}
return crc;
}
```

http://www.termt.com
email: inf@termt.com

ООО Терминальные Технологии ©
Terminal Technologies, Ltd. ©