TERMINAL TECHNOLOGIES

TERMINALTECHNOLOGIESTERMINALTECHNOLOGIESTERMINALTECHNOLOGIESTERMINALTECHNOLOGIESTERMINALTECHNOLOGIESTERMINALTECHNOLOGIESTERMINALTECHNOLOGIES

POS terminal software

VTK protocol

Index: VTK-PROTOCOL-EN
Version: 1.12
Date: 2020-05-07

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
1.5	August 21, 2019	TCP connection control feature added
1.6	October 4, 2019	VTK window size, outgoing byte counter, data blocks and product name added
1.7	December 27, 2019	POS management data added
1.8	February 25, 2020	Local time added
1.9	March 17, 2020	System information added
1.10	March 24, 2020	Banking receipt added
1.11	April 9, 2020	NFCReader protocol usage added
1.12	May 7, 2020	Display time 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 default 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	VTK base protocol: 0x96FB - from VMC; 0x97FB - from POS
Application message	0-65533	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	VTK base protocol: 0x96FB - from VMC; 0x97FB - from POS NFCReader protocol: 0x96FC - from VMC; 0x97FC - from POS
Application message	0-65533	TLV coded message
CRC16 (big-endian)	2	CRC16-CCITT (initial value FFFFh)

VMC and POS must ignore all bytes before 0x1F, when waiting for a new message. When starting byte received, other bytes of this message must be waited within 8 seconds timeout.

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.

If message contains unknown protocol discriminator or invalid CRC, such message must be ignored, and new messages waiting must be continued.

For NFCReader protocol application message contains TLV coded data according NFCReader Functional Design Specification [1]. Not all features of NFCReader protocol are supported. Different versions of POS terminals support different set of features.

Example of NFCReader protocol message over serial link (hex dump):

```
1F 0014 96FC 01 0001 01 20 0004 00000001 11 0004 000000FF 72C9
```

For VTK base protocol 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

Example of VTK base protocol message over serial link (hex dump):

```
1F 000F 96FB 01 03 565250 03 01 35 04 39393030 F404
```

There is no common rule of interleaving for VTK base protocol requests and responses. VMC can send several request before receiving of several response. Some restrictions can be applied in particular cases, depending on hardware and software version.

VTK base protocol and NFCReader protocol messages can be mixed. VTK base protocol messages can be transmitted from any side during NFCreader protocol response waiting. Any NFCReader request can be sent within VTK message flow. However, rules of NFCReader protocol requests and responses interleaving must be considered.

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
0x0B	TCP/IP destination	Binary	No limit	VMC, POS
0x0C	Outgoing byte counter	Decimal in ASCII	12	VMC
0x0D	Simple data block	Binary	No limit	VMC, POS
0x0E	Confirmable data block	Binary	No limit	POS
0x0F	Product name	ASCII	No limit	VMC
0x10	POS management data	ASCII	No limit	VMC, POS
0x11	Local time	ASCII	No limit	VMC, POS
0x12	System information	ASCII	No limit	POS
0x13	Banking receipt	UTF-8	No limit	POS
0x14	Display time in milliseconds	Decimal in ASCII	8	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
CON	Connect message	VMC, POS
DAT	Data message	VMC, POS
DSC	Disconnect message	VMC, 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. IDL or DIS message may be sent before keepalive interval finished, if it used to transmit Event name or QR-Code data.

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.

3.11. TCP/IP destination

It is mandatory parameter in CON message and optional parameter in DAT and DSC messages. It has the following format:

Destination index	1 byte	0x00-0x1F
Destination IP address	4 bytes	Binary format, as in IP protocol header
Destination TCP port	2 bytes	Binary format, as in TCP protocol header
Last connection status	1 bytes	0x00 – no service 0x01 – connection established 0x02 – connection attempts were never made 0x03 – link inactive 0x04 – connection timeout 0x05 – closed by local host 0x06 – closed by remote host other values are reserved and must be processed as "closed by remote host"
VTK window size	2 bytes	Binary format, big-endian
Reserved		Must be ignored

VMC may support TCP/IP connection control feature for a POS terminal. It may be multi-connection service, which implies several active TCP-connections simultaneously, or single-connection service, which may connects to different IP hosts in different time. The CON message is used to open a connection and confirm that a connection attempt is finished. The DAT message is used to transmit data to/from a connection. The DSC message is used to close a connection. These messages may be sent in any states, except INACTIVE state.

If VMC supports N simultaneous TCP sessions (N=1 for single-connection service), it must support destination indexes from 0 to N-1 inclusive; if destination index in CON, DAT, DSC message from POS is out of this range, VMC must immediately send CON message with connection status = 0x00 (no service).

When connection attempt is in a progress, VMC may not send CON messages. However,

it must send CON message immediately when connection attempt finished or when connection timeout expired. Recommended connection timeout for controlled connections is 25 seconds. If a POS terminal sends a CON message and then does not receive any CON message within 30 seconds, it may send CON message for backup host.

TCP/IP destination parameter may be sent in truncated form. For example, DAT and DSC messages may contain 1-byte TCP/IP destination with Destination index only. If DAT or DSC message contains no TCP/IP destination parameter, it means Destination index 0x00.

Both VMC and POS may repeat own CON and DSC messages. If VMC receives CON message when connection is already established, it must send a repeat own CON message. If VMC receives DSC or DAT message when connection is already closed, it must send a repeat of own DSC message.

VTK window size is the number of bytes, which CON message sender is ready to receive over the VTK protocol within data blocks before processing (for VMC "processing" implies sending data to the TCP/IP network and wait for completion of sending). VTK window size in DAT and DSC messages must be ignored. Zero value means sender is able to receive 65536 bytes or more. This version of specification requires POS to receive 65536 bytes or more. If VMC receives non-zero VTK window size in CON messages, it must immediately send CON message with connection status = 0x00 (no service).

3.12. Outgoing byte counter

Number of bytes sent to the remote host over the current TCP session.

3.13. Simple data block

Data bytes to send to/from a connection. When VMC receives a simple data block from the POS, it must just send data to the remote host.

3.14. Confirmable data block

When VMC receives a confirmable data block from the POS, it must perform three steps: 1) send data to the remote host, 2) wait for completion of sending, 3) send DAT message with outgoing byte counter without data block.

3.15. Product name

The product name 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).

3.16. POS management data

It is optional parameter for any message. It has the following format:

	From VMC to POS	From POS to VMC
Byte 1: POS software update status	'L' – download is welcome 'R' – restart is welcome	'L' – download is scheduled 'R' – restart is scheduled
Next bytes reserved	Must be ignored	Must be ignored

A POS may performs an update independently. This document doesn't require VMC to control an update procedure. However, POS may send 'L' byte when download is scheduled,

and VMC may sent 'L' byte to force a download. If a POS receives 'L' byte at least once after start, and if restart required to apply an update, a POS must start to send 'R' byte in all IDL and DIS messages when update finished. VMC may sent 'R' byte to force a restart.

3.17. Local time

The local time in ISO 8601 format with restrictions: only YYYYMMDDThhmmss+xxyy and YYYYMMDDThhmmss-xxyy representations are allowed in this version of protocol. The example is 20210514T105924+0300. If tag length is more than 20 bytes, all additional bytes must be ignored.

If a POS sends a local time, in which the first byte is not equal to "2", a VMC must insert the actual local time to all messages during some transmission interval. The required local time accuracy is determined outside the scope of this document.

The transmission interval begins from a moment when VMC got actual local time from the external sources (for example, from a time server). However, VMC may not request a time server, when VMC already has local time value with sufficient accuracy due to the past request. Therefore the transmission interval may begins from the next message from VMC. The transmission interval must end when VMC receives IDL or DIS message without the Local time or with Local time in which the first byte is "2".

3.18. System information

The system information is intended for diagnostic use.

3.19. Banking receipt

The banking receipt may be transmitted from a POS within VRP and FIN responses. If the vending machine equipped with a printer, VMC can print this receipt. It is recommended to print a receipt from VRP response. If VMC send FIN message with alternate amount (not equal to VRP amount), it is recommended to print a receipt from FIN response as well.

The <LF> symbols (0x0A bytes) within receipt must be processed as new line start.

3.20. Display time in milliseconds

In this version of the document, this is the desired time for displaying the QR code. VMC can only send this parameter together with the QR code data parameter in the same message.

The actual time of displaying the QR code may differ from the desired time.

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 --->		
	(ABR --->)	(<--- MFR)	
	(CON --->)	(<--- CON)	
	(DAT --->)	(<--- DAT)	
	(DSC --->)	(<--- DSC)	
			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 --->		
	(ABR --->)	(<--- MFR)	
	(CON --->)	(<--- CON)	
	(DAT --->)	(<--- DAT)	
	(DSC --->)	(<--- DSC)	
			Vend declined
		<--- VRP	
Session completed	IDL --->		
		<--- IDL	Move to IDLE state
Move to IDLE state			

4.6. CDP operation

Money is accepted	CDP --->		
	(ABR --->)	(<--- MFR)	
	(CON --->)	(<--- CON)	
	(DAT --->)	(<--- DAT)	
	(DSC --->)	(<--- DSC)	
			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

4.8. TCP connection control

		<-- CON	Open a connection
Connecting			
Connection established	CON -->		
	<--> DAT		
		<--DSC	Close a connection
Disconnecting			
Closed by local host	DSC -->		
		(<-- DSC)	Close a connection - repeat
Closed by local host - repeat	(DSC -->)		

		<-- CON	Open a connection
Link inactive	CON -->		
		<-- CON	Open a connection
Connecting			
Connection established	CON -->		
	<--> DAT		
Closed by remote host	DSC -->		
		<-- CON	Open a connection
Connecting			
Connection timeout	CON -->		
		<-- CON	Open a connection - backup
Connecting			
Closed by remote host	CON -->		
		<-- CON	Open a connection
Connecting			
Connection established	CON -->		Message lost
		<-- CON	Open a connection - backup
Connection established - repeat	CON -->		
		<-- DAT	Simple data block
Data is sent to the remote host			
		<-- DAT	Confirmable data block
Data is sent to the remote host			
Confirming	DAT -->		No data blocks
Data is received from remote host			
	DAT -->		Data block received
		<--DSC	Close a connection
Disconnecting			
Closed by local host	DSC -->		

4.9. POS software update

VMC event	Message from VMC	Message from POS	POS event
INACTIVE state after power-up			
	IDL/DIS --->		
10 seconds delay			INACTIVE state after power-up
	IDL/DIS --->		
		<--- IDL/DIS	Message contains «download is scheduled» status
Keepalive interval delay			
Message contains «download is welcome» status	IDL/DIS --->		
		<--- IDL/DIS	Message contains «download is scheduled» status

(normal work continues, new software is loading in parallel)

			Download is finished
Message contains «download is welcome» status	IDL/DIS --->		
		<--- IDL/DIS	Message contains «restart is scheduled» status
Keepalive interval delay			
Message contains «restart is welcome» status	IDL/DIS --->		
		<--- IDL/DIS	Message does not contain «restart is scheduled» status
Keepalive interval delay			
Message does not contain «restart is welcome» status	IDL/DIS --->		

(POS restarts)

4.10. Local time transmission

VMC event	Message from VMC	Message from POS	POS event
INACTIVE state after power-up			
	IDL/DIS --->		
10 seconds delay			INACTIVE state after power-up
	IDL/DIS --->		
		<--- IDL/DIS	Message contains local time «19700101T000000+0300»
VMC may communicates with time server if required			
The exact time is not yet known	IDL/DIS --->		
		<--- IDL/DIS	Message contains local time «19700101T000000+0300»
Keepalive interval delay			
Message contains local time «20210514T105924+0300»	IDL/DIS --->		
		<--- IDL/DIS	Message contains local time «19700101T000000+0300»
Keepalive interval delay			
Message contains local time «20210514T105934+0300»	IDL/DIS --->		
		<--- IDL/DIS	Message contains local time «20210514T115935+0300» or no local time in the message
Keepalive interval delay			
No local time in the message	IDL/DIS --->		

5. CRC calculation example

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

```c
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;
}
```

6. References

№	Title	Author	Version	File name
[1]	NFCReader protocol. Functional Design Specification.	Terminal Technologies	1.x	nfcreader-protocol-en.pdf

TERMINAL TECHNOLOGIES

TERMINALTECHNOLOGIESTERMINALTECHNOLOGIESTERMINALTECHNOLOGIESTERMINALTECHNOLOGIESTERMINALTECHNOLOGIESTERMINALTECHNOLOGIESTERMINALTECHNOLOGIES

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