



API Protocol Specification

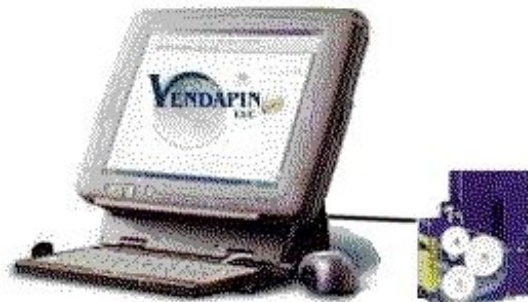
For

CTD-202 USB Version

&

CTD-203 RS-232 Version

Card Dispenser



www.vendapin.com

Email: Robert@VENDAPIN.com

VENDAPIN LLC
40 Elm Street
Dryden, NY 13053
(607) 844-8134
(775) 514-7530 Fax

Revision: August 28, 2005

Command Packet Specification

The command packet, normally transmitted from host controller such as PC, embedded controller to CTD-202/CTD-203 card dispenser controller is the beginning of a communication session. It is used for transmitting the data/requests to other controller via RS-232 or USB to initiates a command of action (such as disable or enable, requests, etc.).

The packet breakdown is as follows:

<STX><ADD><CMD><LEN><DTA><ETX><CHK>

Name	Hex	Definition
STX	02h	Start of Text Data
ADD	00h+	Device Address (Set to 0x01 if not used)
CMD	80h+	Command Code Byte (0x80-0xFF)
LEN	00h+	Length of Data – Byte Size (0x00 or higher)
DTA	--h	Packet Data (Optional, leave empty if not used)
ETX	03h	End of Text Data
CHK	--h	XOR Checksum of data packet

Minimum Packet Length – 6 bytes

Maximum Packet Length – 128 bytes

You can calculate the XOR checksum by using Windows calculator with scientific view, and select HEX radio button to add STX+ADD+CMD+LEN+DATA+ETX together using XOR button to get the XOR hex value.



Response Packet Specification

The response packet, normally transmitted from other controllers to card dispenser controller in response to a command packet, is the second (and usually final) stage of a communication session. It is used to transfer the requested data back to card dispenser controller and give a response code.

Name	Hex	Definition
STX	02h	Start of Text Data
ADD	00h+	Device Address (Set to 0x01 if not used)
CMD	06h, 15h, FD-FFh	Reserved Response Code Byte (see below for details)
LEN	00h+	Length of Data – Byte Size (0x00 or higher)
DTA	--h	Packet Data (Optional, leave empty if not used)
ETX	03h	End of Text Data
CHK	--h	XOR Checksum of data packet

Minimum Packet Length – 5 bytes

Maximum Packet Length – 128 bytes

Reserved Response Codes

A slave/host device, in responding to a request from the master/host device, includes a response byte in its response packet. This byte can be used to replay the success or failure of a particular commands execution or communication error. Most of the response codes are defined in the devices API specification, but a few response codes are reserved as following:

Reserved Response		
Name	Codes	Definition
ACK	06h	Accepted/Positive Status
NAK	15h	Rejected/Negative Status
	FDh	Incomplete Command Packet
	FEh	Unrecognized Command Packet
	FFh	Data Packet Checksum Error

It is imperative that all master/host devices that adhere to this specification handle these response codes.

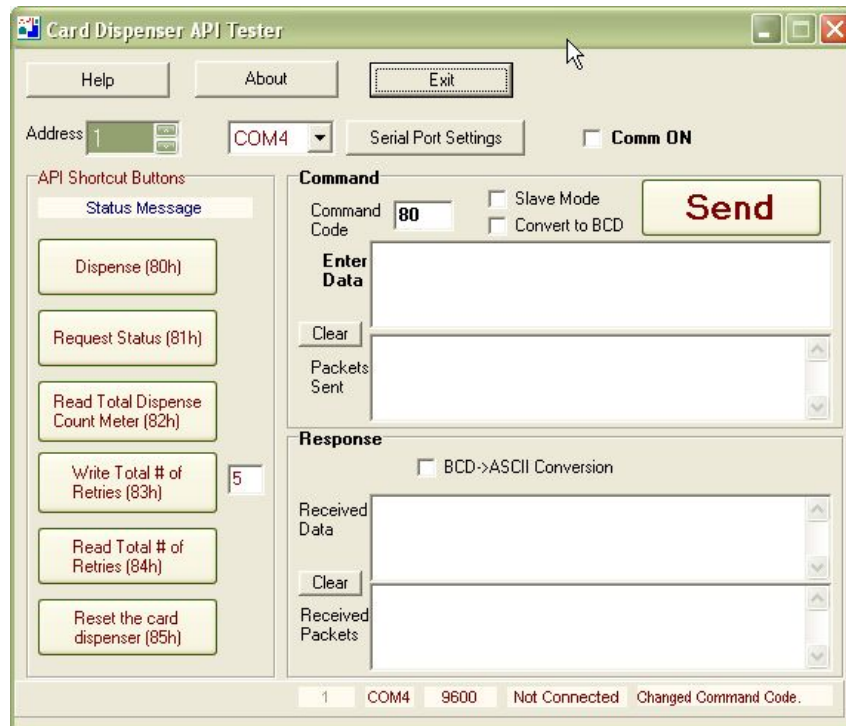
Quick Guide Command Codes List

Command codes will be in the range of 0x80 – 0xFF hex value. Host controller communicating to card dispenser controller must handle the following reserved and standard commands:

Reserved Command Codes	Definition
FFh	Disable Card Dispenser
FEh	Enable Card Dispenser (Default at start-up)
F0h	Reserved – for Factory use only

Command Codes	Definition	Type of Variable (in Data Packet)
80h	Dispense Card	None
81h	Request Status	None
82h	Read Total Dispense Count Meter	None
83h	Write Total Number of Retries	1 byte unsigned CHAR
84h	Read Total Number of Retries	None
85h	Reset the card dispenser	None

You can test the card dispenser using VENDAPIN's Card Dispenser API Tester Windows based program.



Screenshot of
Card Dispenser API Tester Program

Detailed Command Codes & Response Packets

80h – Dispense Card

Command Send Packet: <STX><01><80h><00h><00><ETX><CHK> (02 01 80 00 03 80)

This Command Send packet sends Dispense Card request from the host to the card dispenser to dispense the card.

The Return Response packet from the card dispenser would be the following if the packet command has the correct checksum:

<STX><01><ACK><00><ETX><CHK> (02 01 06 00 03 06)

Where ETX is <03h> and CHK is <06h>. After a good return status message, the card dispenser total dispense count meter is updated automatically, only after the card is completely dispensed.

If a checksum error is found in the packet, the card dispenser will respond with:

<STX><01><FFh><00><ETX><FFh> (02 01 FF 00 03 FF)

If the card dispenser is not ready, stuck, empty or other reasons, it will return with NAK response code along with CODE hex value as described here:

0x30 or '0' – READY
0x31 or '1' – BUSY
0x32 or '2' – EMPTY
0x33 or '3' – STUCK
0x34 or '4' - DISABLED
0x35 or '5' – Other

Return Response Packet: <STX><01><NAK><01h><CODE><ETX><CHK>

An example for return response packet could be read if the card dispenser is **EMPTY**:

Return Response Packet Format:

<STX><01><NAK><01h><32h><ETX><CHK> (02 01 15 01 32 03 26)

Byte	Description
<STX>	02h (Start Transmission)
<01h>	Device address of 1.
<NAK>	06h Acknowledge command accepted and response follows
<01h>	Data Length
<32h >	EMPTY (based on CODE definition)
<ETX>	03h or End of Text (data)
<26h>	XOR Checksum

81h - Request Status

Command Send Packet Format: <STX><01><81h><00h><ETX><CHK> (02 01 81 00 03 81)

This Command Send packet requests the card dispenser status to be sent to host.

The Return Response packet from the card dispenser would be the following if the command packet checksum were correct:

Return Response Packet Format: <STX><01><06h><01h><CODE><ETX><CHK>

This Return Response packet will include CODE (hex value) sent by the card dispenser. The CODE hex definition list would be described here:

0x30 or '0' – READY
0x31 or '1' – BUSY
0x32 or '2' – EMPTY
0x33 or '3' – STUCK
0x34 or '4' - DISABLED
0x35 or '5' – Other

An example for return response packet could be read if the card dispenser is **READY**:

<STX><1><06><01><30><ETX><37> (02 01 06 01 30 03 37)

Byte	Description
<STX>	02h (Start Transmission)
<01h>	Device address of 1.
<ACK>	06h Acknowledge command accepted and response follows
<01h>	Data Length
<30h >	READY (based on CODE definition)
<ETX>	03h or End of Text (data)
<37h>	XOR Checksum

An example for return response packet in hex could be read if the card dispenser is **EMPTY**:

<STX><1><06><01><32><ETX><35> (02 01 06 01 32 03 35)

Byte	Description
<STX>	02h (Start Transmission)
<01h>	Device address of 1.
<ACK>	06h Acknowledge command accepted and response follows
<01h>	Data Length
<32h >	EMPTY (based on CODE definition)
<ETX>	03h or End of Text (data)
<35h>	XOR Checksum

Note: ACK or NAK response code would allow the host to “act” if the card dispenser is not “READY” if NAK response code is read.

82h – Read Total Dispense Count Meter

Command Send Packet: <STX><01><82h><00h><ETX><CHK> (02 01 82 00 03 82)

This Command Send packet requests the read value to be sent to host.

The return response packet sent by the card dispenser would be the following if the command packet checksum were correct:

Return Response Packet Format:

<STX><01><06h><04h><4 BYTES READ VALUE><ETX><CHK>

4 Bytes Read Hex Value would be formatted as BCD as shown here:

Hex Value	Actual Value
00000000h	0
00999999h	99999
00000345h	345

An example for return response packet in hex could be read for requesting the total dispense count meter:

<STX><1><06><04><00002345><ETX><37> (02 01 06 04 00 00 23 45 03 64)

Byte	Description
<STX>	02h (Start Transmission)
<01h>	Device address of 1.
<ACK>	06h Acknowledge command accepted and response follows
<04h>	Data Length
<00002345h>	2345 (Total Dispense Count Value)
<ETX>	03h or End of Text (data)
<64h>	XOR Checksum

83h – Write Total Number of Retries

Command Send Packet: <STX><01><83h><01h><1 byte - RETRIES><ETX><CHK>

This Command Send packet requests to write/update the total number of retries value to EEPROM memory.

This “number of retries” feature will allow the card dispenser to re-attempt to dispense a card if the card dispense cycle is not successful. If the total number of retries reaches to “maximum” total level, the card dispenser will stop dispensing the card and set the status to “STUCK”. This “STUCK” status will remain until:

- 1) The “Reset” API Command is sent to reset the card dispenser to “READY” status or “EMPTY” status (out of cards).
- 2) Reset button (on VCB-2 board) is pressed to reset the card dispenser, and return to “READY” or “EMPTY” status.
- 3) Power-cycle the card dispenser to return this unit to “READY” or “EMPTY” status.

1 Byte Hex Value would be formatted as BCD as shown here:

Hex Value	Actual “number of retries” Value
00h	10
10h	10
20h	20

Notes:

- 1) If the total of retries value exceed 25, it will set to 25 as the maximum value automatically.
- 2) “EMPTY” status occurs only when there are no cards in the card dispenser stack.

An example for Command Send packet in hex is formatted to write to card dispenser to write the value to EEPROM memory:

<STX><01><83><01><12><ETX><90> (02 01 83 01 12 03 90)

Byte	Description
<STX>	02h (Start Transmission)
<01h>	Device address of 1.
<83h>	83h Command Code for “Write Total Number of Retries”
<01h>	Data Length
<12h>	12 “number of retries” BCD Value
<ETX>	03h or End of Text (data)
<90h>	XOR Checksum

The return response packet sent by the card dispenser would be the following if the command packet checksum were correct:

Return Response Packet Format:

<STX><01><06h><00h><ETX><CHK> (02 01 06 00 03 06)

84h – Read Total Number of Retries

Command Send Packet: <STX><01><84h><00h><ETX><CHK> (02 01 84 00 03 84)

This Command Send packet requests the read value to be sent to host.

The return response packet sent by the card dispenser would be the following if the command packet checksum were correct:

Return Response Packet Format:

<STX><01><06h><01h><1 BYTES READ VALUE><ETX><CHK>

1 Byte Hex Value would be formatted as BCD as shown here:

Hex Value	Actual “number of retries” Value
00h	10
10h	10
20h	20

An example for return response packet in hex could be read for requesting the total dispense count meter:

<STX><1><06><01><05><ETX><02> (02 01 06 01 05 03 02)

Byte	Description
<STX>	02h (Start Transmission)
<01h>	Device address of 1.
<ACK>	06h Acknowledge command accepted and response follows
<01h>	Data Length
<05h>	5 “number of retries” BCD Value
<ETX>	03h or End of Text (data)
<02h>	XOR Checksum

85h – Reset the card dispenser settings

Command Send Packet: <STX><01><85h><00h><ETX><CHK> (02 01 85 00 03 85)

This Command Send packet requests the card dispenser reset to be sent to host.

The return response packet sent by the card dispenser would be the following if the command packet checksum were correct:

Return Response Packet Format:

<STX><01><06h><00h><ETX><CHK> (02 01 06 00 03 06)

(Acknowledgement response packet)