16381 Cherokee Road
Brooksville, Florida 34601-4202
Sales & Marketing Telephone: (352) 796-2693
Direct Factory Telephone: (607) 844-8134
Direct Fax: (775) 521-1673 Alternate Fax: (775) 514-7530
Cellular Phone: (727) 243-8590
Sales Web Site: http://www.vendapin.com
E-mail: Darrell@vendapin.com

June 27, 2004

5580 API Revision 4.00 Protocol Specification

Table of Contents

1.1 Functional Overview

Our customers require in this day and age the ability to interface their hardware/software programs of their products to our products. Our CECB4 is exactly that type of product. The CECB4 "Cash Escrow Controller Board Version #4" is the most powerful vending controller on the market today! The CECB4 will interface to virtually any other computer driven product. Examples are; Print Vending Software Programs, Library Patron Check Out Software, Copier, Printer and other computer driven office machines. Virtually any microprocessor driven machine can be interfaced to our CECB4 Vending Controller Board.

1.2 Introduction

This document provides detailed information on the API commands used to interface the CECB4 5580 vending controller board and a host 5880. The communications interface from the 5580 is a master/slave where the 5880 is the master and the 5580 is the slave. Note on nomenclature - in all following descriptions, hexadecimal values are indicated by a trailing 'h' character after a number, while decimal values are indicated by a trailing 'd'. Numbers encased in brackets are assumed to be hexadecimal if there is no trailing character. (E.g. <80> means 80h, or 128d).

3.2 API Protocol Format

API Protocol Format sequence uses a common protocol consisting of a command packet and a response packet. The packet structure ensures a method of error detection and recovery. A program called the COMM-TESTER can be used for testing the 5580 that communicates through the 5580 RS232 port to a PC running the COMM-TESTER (available from VendaPin, L.L.C. This allows manual simulated testing of commands from the 5880 to the CECB4 5580.

The RS232 format is 9,600 baud, 1 start bit, 8 data bits, and 1 stop bit. This is a four wire connection with Transmit (TX), Receive (RX), Clear To Send (CTS) and Ground (GND). The 5580 interface is half duplex where the host is the master. The initial handshaking for a command is for the 5880 to assert RTS to raise the CTS signal to the CECB4 5580. The 5580 acknowledges with an acknowledge packet (06h) to signify that it is ready to receive data and then the 5880 issues a command packet. The CTS is held high by the 5880 until a good command packet has been acknowledged from the 5580 with an (06h) response packet.

3.3 API Command Packet Structures

The command packet is transmitted from the host to the 5580. It is the beginning of a communications sequence. The packet has the following structure:

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

Where:

STX ASCII Character For Start Of Text - 02h
ADD Device Address - this byte must be one (1)
CMD Command Byte - 80h as described in section III
LEN Length of Data Packet (5d)
DTA Packet Data (contains the number of bytes specified in LEN)
ETX ASCII Character For End of Text - 03h
CHK Checksum Of Packet Data - the XOR (exclusive OR) of all bytes from STX to ETX
Minimum Packet Length - 6 Bytes
Maximum Packet Length - 70 bytes (6 bytes packet format + 64 bytes data)

API Response Packets - The response packet is transmitted from the 5580 to the host. It is the termination of a communications sequence. The packet has the following structure:

<STX><ADD><RSP><LEN><DTA><ETX><CHK>
Where:
STX ASCII Character For Start Of Text - 02h
ADD Device Address - this byte is not used in this application and must be one (1)
RSP Command Byte – as described below LEN Length of Data Packet (1 - 22d)
DTA Packet Data (contains the number of bytes specified in LEN)
ETX ASCII Character For End Of Text - 03h
CHK Checksum Of Packet Data - the XOR (exclusive OR) of all bytes from STX through ETX
Minimum Packet Length - 6 Bytes
Maximum Packet Length – Theoretical 261 bytes (6 bytes packet format + 255 bytes data)

Responses to a Host Command Packet are:

06h Command accepted
Error return codes:
F8h Trying to send a second <80h> Send Escrow Value command without an <81h> Request Remaining Escrow command.
F9h Trying to send a <81h> command without first sending an <80h> command.
FAh Error in size of data packet (ETX not found where expected).
FCh Data value is invalid.
FDh Command packet not completed (timeout occurred – 100 Msec without a complete packet received).
FEh Unrecognized command code.
FFh Checksum error in command packet.

If the response code is not 'Command Accepted' then the host must retransmit the command packet.

These error responses can also be sent from the 5880 host to the 5580 if a data packet from a '81h' command is received incorrectly.

Command Formats are:

80h – SEND ESCROW VALUE

<STX><01><80h><02h><AMT><ETX><CHK>

This command sends an Escrow value from the host to the 5580. The AMT is two bytes, MSB (Most Significant Byte) followed by LSB (Least Significant Byte). The least significant bit of the least significant byte (LSB) is five cents.

An example command in hex could be:

<02><01><80><02><00><96><03><14>

Where :

<02h> = ‘SYN’ character, beginning of packet.

<01h> = Device address of 1 (which will be returned in the return packet).

<02h> = Data Byte Count of 2

<0h><96h> = 96 hex nickels (150) which is a $7.50 Escrow (note that Escrow must be rounded to five cents before transmitting)

<03h> = ‘ETX’ or End of Text (data)

<14h> = Checksum

The return message from the 5580 would be the following if the packet command has the correct checksum:

<STX><01><06><00><ETX><CHK>

Where ETX is <03h> and CHK is <06h>. After a good return status message, the 5580 Escrow is updated and displayed. The bill acceptor and coin acceptor are both disabled so that only the new Escrow is utilized.

Example is <02><01><06><00><03><06>.

If a checksum error is found in the packet, the CECB4 will respond with:

<STX><01><FFh><00><ETX><FFh>

For example.

81h REQUEST REMAINING ESCROW <STX><01><81h><00><ETX><CHK>

This command requests the remaining CECB4 5580 Escrow immediately be sent to the 5880 host. After a successful transmission, the Escrow is cleared to zero in the CECB4 5580. Also the bill acceptor and coin acceptor are enabled and normal operation occurs in the 5580.

The command in hex would be

<02><01><81><00><03><81>

The Return Response messages from the 5580 would be the following if the command packet checksum was correct:

<STX><01><06h><02h><AMT><ETX><CHK>

This command sends an Escrow value from the 5580. The AMT is two bytes, MSB (Most Significant Byte) followed by LSB (Least Significant Byte). The least significant bit of the least significant byte (LSB) is five cents.

An example command in hex could be:

<02><01><06><02><00><96><03><92>

Where :

<02h> = Data Byte Count of 2

<01h> = Device address of 1.

<06h> = Acknowledge command accepted and response follows

<0h><96h> = 96 hex nickels (150) which is a $7.50 Escrow (note that Escrow must be rounded to five cents before transmitting)

<03h> = ‘ETX’ or End of Text (data)

<92h> = Checksum

If the packet is received correctly by the 5880 it will drop the RTS line (CTS line to 5580).

If a checksum error is found in the packet, the host 5880 would respond with:

<STX><01><FFh><00><ETX><FFh>

See above for possible error codes.

The 5580 will respond with a retransmit of the packet on this or any error return from the 5880.

[END OF DOCUMENT]

Any questions, please write to us for any support required.

Sincerely,

Darrell G. Rademacher
Managing Director