V2.2
Protocol

GPT-Enhanced Serial Protocol

(Preliminary Copy)

Revision B

3 July 1997

GPT

GLOBAL PAYMENT TECHNOLOGIES

IMPORTANT NOTICE

Global Payment Technologies, Inc., (the Company), reserves the right to make changes to the contents of this publication, including functions and specifications identified herein, without notice.

In the absence of a written agreement to the contrary, the Company, assumes no liability for the applications assistance, customer's product/application/concepts, or infringements of patents or copyrights of third parties arising from the use of products methods and architectures described herein. Nor does the Company, warrant or represent that any license, either expressed or implied, is granted under any patent right, copyright, or other combination of technology, architecture, or software as might be or is already in use.

Note: The material contained in this document is only intended for use by the Company, its employees and any licensed or authorized user of its proprietary products and/or software.

The techniques, procedures and information described herein are Confidential Information and should not be reproduced, disseminated, or otherwise disclosed without prior written consent from an officer of the Company. Any authorized licensee or user specifically acknowledges that such techniques, procedures and information comprise valuable trade secrets and proprietary property of the Company and that all patent, copyright, trade secrets, trade/service mark, and any other right, title or interest therein are the sole property of the Company.

This document has been copyrighted by the Company. The presence of any copyright notice within or upon the medium containing the Company's proprietary products shall not be construed as constituting any publication, any intention to publish or otherwise impairing the confidential nature thereof.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

1. Introduction

This section defines the Global Payment Technologies' enhanced serial protocol. The purpose of this protocol is to provide a reliable and robust communication link between the Validator and the host machine controller. The protocol contains features designed to detect transmission errors and allow for the retransmission of data in case an error occurs. It also allows full access to the validators features and gives detailed information on the current state of the validator.

2. Transmission format

Transmissions consist of message frames. Message frames are sent both by the controller and the validator. Acknowledge and IRQ bytes are sent only by the validator. The format of a message frame (below) consists of a start byte (STX), a length byte (LEN), a command / identifier byte (CMD), a number of data bytes between 0 and 251, and a check byte (CHK).

message frame: STX | LEN | CMD | DATA (0 to 251 data bytes) | CHK

STX : Start character = 02H

LEN : The total number of bytes in the message, (includes STX, LEN, CMD and CHK bytes)

CMD : The ***command/identification byte***.. For a controller to validator command this byte represents the command to the Validator. For a validator to controller status response this byte echoes the command byte of the previous validator command (except for a Retransmit command, where the command byte is the previous command).

DAT : Some number of ***data bytes*** that contain information pertaining to the command .

CHK : ***Check byte***, calculated by taking a one byte binary sum (ignoring the carry) of all bytes in the message frame from (and including) the STX byte to the last data byte. The two's complement is then taken (invert all bits of the result and add 1) to yield the check byte. Note that if the message is correct adding the check byte to the sum of all the other bytes in the message will give a result of 0.

Acknowledge byte : ACK or NAK

ACK : Validator has correctly received message from controller, and operation commanded has been or will be executed.

transmission package from validator to controller [02][04][00][FA]

NAK : Message was not correctly received. Command will be ignored.

transmission package from validator to controller [02][04][FF][FB]

IRQ : Sent by validator to controller, to request the controller, to communicate with the validator.

transmission package from validator to controller [02][04][55][A5]

- Byte Format:
- Baud rate = 9600
- Start bits = 1
- Data bits = 8
- Parity = Even
- Stop bits = 1

The ***Host Machine*** control computer (hereafter referred to as the controller) is the master of the serial link. The controller can initiate a message sequence at any time. The validator may only send a message frame in response to a request by the controller. The validator will respond to a message frame in one of three ways. If the message is a control command to the validator, the validator will respond with an ACK transmission package who's command byte is 00. This is a package signaling the correct reception of the command. If the message is a status command requiring information be returned to the controller, the Validator responds with an answering message frame of the same format (but not necessarily the same number of bytes). If the message was not correctly received (bad check sum or number of bytes etc.) the validator responds with a NAK transmission package, who's command byte is equal to FF hex. The controller must wait for a response from the validator before sending another command. The controller may send another command immediately after it is finished receiving the validator response if desired, but not before this. If no response is received within a specified period of time the controller should assume that its message was not received, and treat it as if it were a NAK response. More detail on this is given in the section on message timing.

3. Validator States and Events

To gain an understanding of how the communication protocol works, what is meant by a validator "state" and an "event" must first be explained.

A ***validator state*** is the action that the validator is performing at that instant of time. As a bill moves through the validator, the validator goes through different states as it performs the different functions necessary to validate and accept the bill. The validator will report that it is in a certain state for as long as the state is active. This may be for a period covering many messages.

A ***validator event*** is some significant occurrence that has taken place in the validator, for example a bill being stacked. When an event occurs a flag is set in the validator software. When the validator status is reported the flag is cleared. In this way the event is only reported once to the controller for each time it happens. What follows is a list and description of the various validator states and events.

3.1 Validator States

3.1.1 Idle state

The validator is "idle" as it waits for the input of a bill. In this state the validator is fully operational and ready for the input of currency.

3.1.2 Accepting state

The front sensor has been blocked and the bill moved into the validator. The validator is in the process of taking data and evaluating the bill to determine if it is valid.

3.1.3 Rejecting state

The validator has determined that the bill is not valid for acceptance. This can be due to a misfeed, currency which does not meet the criteria for validation, or a bill type (value) which is not enabled. Whatever the reason, the bill is in the process of being rejected and returned to the customer. Upon the successful ejection of the bill the validator returns to the idle state.

3.1.4 Stacking state

The validator has determined that the bill is valid and enabled for acceptance. It is in the process of moving the bill into the stacker and then stacking the bill. The state is exited after the bill has been stacked. Note that for stackerless systems this state is exited when the bill clears the rear flag.

3.1.5 Escrow state: (only occurs if escrow mode is enabled)

The validator has determined that the bill is valid and enabled for acceptance. The bill is being held in the validator channel. The bill will be held in the chamber for certain period of time. The controller commands the validator to accept the bill or eject the bill. If the controller commands the validator to accept the bill, the validator will enter the stacking state. If the controller sends the eject command the validator will enter the returning state. Please note that if the controller does not communicate with the validator within the next 10 second to accept or reject the bill, validator will automatically eject the bill.

3.1.6 Returning state

The validator has determined, or being commanded to return a validated bill to the customer. This state is entered from the escrow state when an eject bill command is given. It is also entered if, during the stacking state, the validator is unable to move the bill into the stacker. This can be due to a misfeed of the bill or attempted tampering with the validator by the customer, (such as "stringing" a bill). No credit should be given by the controller. The validator will exit from this state when the bill is ejected from the channel.

3.1.7 Stacker Full/Jam State

The stacker is unable to complete it's cycle of pushing a bill into the stacker due to stacker full or jam condition. The validator is unable to accept any currency. This state can be cleared by detaching the stacker, emptying it, and re-attaching it. Both d3 and d1 of data byte 3 of the validator status (refer to section 9.1.35) will be set since we do not have the ability to distinguish Stacker full from stacker jam at present time.

3.1.8 Fault state:

A ***fault condition*** has occurred which prevents the validator from being able to operate. In this state the validator will self inhibit and stop accepting currency. Validator faults may be temporary and clear themselves or they may require the validator to be serviced. There are several possible reasons for the validator entering the fault state.

A bill may be jammed in the channel, and the validator is unable to move it out of the channel and return it to the customer. The validator may have been unable to stack the previous bill (stacker jam). The stacker may be full or have been removed. This state is also entered if the validator self test determines that the validator has developed a hardware fault and is unable to operate. The validator will attempt to remedy the fault, and if successful, will exit the fault state. If the fault is externally cleared the validator will also exit the fault state.

3.1.9 Inhibited state

The validator is operational but has been inhibited from accepting any bills. This can be due to an inhibit command issued from the controller, or a loss of communication between the validator and the controller. The validator can also be inhibited by a external hardware signal if this line into the validator is used. In this state the motor will not turn if the front sensor is blocked, but the unit will continue to listen for messages from the controller.

3.1.10 Power up with bill in channel

The validator only enters this state when, upon power on reset, the currency channel is not empty. The channel may contain a bill or some type of foreign material. If it contains a bill the value may or may not be known. This can happen if power should fail in the middle of the validation process. Should the validator power up in this condition the following will occur: The “Power-up with bill in the channel “ flag will be set in validator status word. The validator will then wait 5 second for Stack Bill or a return bill command from the controller. The validator will attempt to either stack or return the bill as commanded. If no message is received for 5 second after power-up the validator will automatically enter the returning state and attempt to eject the bill. If the bill is successfully returned or stacked the appropriate bit will be set in the validator status word.

The power-up with bill in channel flag will remain set until the first validator status read after the stacking or returning state is entered. This way if the controller does not start polling the validator until after the above sequence has completed the controller will still know what occurred. (The first read by the controller would have the Power-up, Power-up with bill in channel, and bill returned flags all set).

> **Note:** The Idle State, Accepting State, Rejecting State, Stacking State, Escrow state, and Returning state are also referred to as Bill Processing State.

3.2 Validator Events

3.2.1 Bill Stacked event

A valid bill has been stacked (or for stackerless units moved past the rear flag into the chamber). The bill can no longer be returned to the customer. This signals the successful completion of a bill transaction. The controller may now issue credit for the bill. This is the ONLY time the controller should issue credit. After a bill is stacked a status request message must be received by the validator before it will accept another bill. This prevents the loss of credit by the customer if the controller fails to poll frequently enough.

3.2.2 Bill Returned event

The bill has been successfully ejected from the validator and returned to the customer. This event may occur even if a stack bill command was given, if the validator was unable to complete that command.

3.2.3 Cheated event

The validator detected an attempt by the customer to remove a bill from the escrow position. The validator will automatically go to the returning state and try to eject the bill. A Bill Returned event flag will be generated when the bill has been ejected. The controller should cancel the transaction.

3.2.4 Power On Reset event

The unit has been started up from the power down state. This flag is set during validator initialization. If it occurs in other than initial power-up of the system, it means the validator has momentarily lost power.

3.2.5 Communication time-out event

The unit lost contact with the controller and had inhibited itself, refer to the message timing section. Note that in order to receive this message you must have re-established contact with the validator. It is just an indicator that a failure had taken place.

4. Modes

4.1 Modes of Communication

There are two basic modes of communication: polled mode and interrupt mode.

4.1.1 Polled Mode

In the ***polled mode*** the controller periodically "polls" the validator with a validator status request message. It is the responsibility of the controller to make sure the Validator is polled frequently enough so that proper operation is assured. It is recommended that the validator be polled at least 2 times a second

4.1.2 Interrupt Mode

In this mode the validator will send an interrupt request transmission package (data byte = 55H) to the controller when it has a message it wants to send. The interrupt mode is useful for applications where the controller is normally busy and does not have time to constantly poll the validator. In this mode the validator will send an IRQ message to the controller when it needs the controllers attention. IRQ's are generated by any of the following:

Going from ACCEPTING state to ESCROW state

Going from any state into a FAULT state

Exiting from a FAULT state into an operational state (fault removed)

A Bill Stacked event occurring

A Bill Returned event occurring

A Cheated event occurring

A bill Rejecting state

Stacker full/jam state

The validator expects the controller to respond to the IRQ by sending a status request message. If the validator does not receive a valid status request message within 200 milliseconds, the validator will send another IRQ message. The validator will continue to repeat the IRQ once every 200 msec until the controller answers.

The controller is allowed to poll the validator when it is in the interrupt mode. The programmer of the controller must be aware however that if the controller requests status simultaneously as an IRQ is generated, the event that generated the IRQ may have been cleared by the transmission of status to the controller. It will then appear that an IRQ was generated for no reason. The programmer must allow for this possibility in the controller program. (If the status request shows no activity then the controller should ignore the transaction).

4.2 Modes of operation

In addition, the validator will operate in either an escrow or a non-escrow mode for both of the previously mentioned communications modes. In the ***non-escrow mode*** a valid bill which is input to the validator will be automatically moved to the stacker. In the ***escrow mode*** the bill will be held in the channel of the validator. The controller must then command the validator to accept or eject the bill. The power-up defaults for the validator are the polled communication mode and the non-escrow mode. If the controller wishes to use other than these default settings, it must send the validator a set-mode command.

5. Message Timing

5.1 Validator response time

Each command has a maximum response time associated with it. The response time is measured from the end of the controller command to the start of the validator response, see Figure 5-1 Message Response Time. When the controller sends a command, it must wait for a response from the validator for this amount of time. If the time has expired and the validator response has not started, the controller may assume that the validator is not going to answer. The controller should then try again.

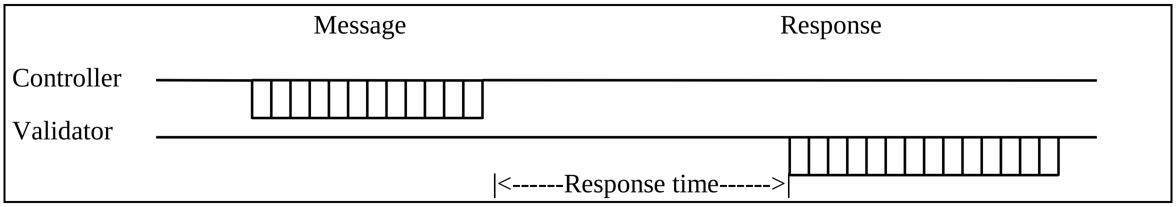


Figure 5-1 Message Response Time

The validator will respond to most of the commands within 20 milliseconds. However, commands that take too long to execute in the interrupt routine will not be answered, and not be processed while the validator is accepting currency. The following are the list of these commands:

Reset Production Date command	code = 30H
Reset Factory Serial Number command	code = 31H
Reset Current Statistic command	code = 32H
Reset thousand events statistics command	code = 33H
Reset bill counter command	code = 34H
Reset unit setup command	code = 35H
Reset validator command	code = 36H
Write System ID Number command	code = 50H
Set Real Time Clock command	code = 51H
Upload Command	code = 60H
Download (Network File only)Command	code = 61H
Send CRC calculation parameter Command	code = 62H
Request CRC value of the code Command	code = 63H
Request System ID Number command	code = D0H
Read Real Time Clock command	code = D1H
Request Validator Serial Number command	code = D2H
Request Asset Number command	code = D3H
Request Factory Number command	code = D4H
Request Production Date Code command	code = D5H
Request Program Revision command	code = D8H
Request Last 5 Transactions command	code = D9H
Request Life Time Counter command	code =DAH
Request Fault Buffer from Stacker command	code =DBH
Request Stacked Bill Information command	code =DCH
Request Current Statistics command	code =DDH
Request Thousand Attempts Statistics command	code =DEH
Request validator setup command from memory	code = E0H
Request current validator setup command	code = E1H
Request validator options command:	code = E2H
Uninhibit command	code = F1H
Change Default command	code = F3H
Set Inhibit Logic Regular	code = F4H
Set Inhibit Logic Reverse	code = F5H

It is strongly recommended that these commands must wait for the validator to finish bill processing before they can be sent. These commands are not intended to be used while the validator is in the bill processing state. It is recommended the controller send an inhibit command to the validator, and wait until the inhibit flag is set. The validator is now guaranteed not busy when the command comes. After the command is executed, the validator may be uninhibited.

5.2 Message Interbyte time

Once a message has started, the maximum allowable time between bytes is 5 milliseconds. This applies to messages from the controller and the validator. This time is measured from the end of one byte to the start of the next. ***See*** Figure 5-2 Inter-byte Time. If the next byte is not received within the 5 msec limit, the receiving unit should treat it as the end of the message, even if it does not match the word count at the start of the message. Note that if the previous condition occurs, the received message will be treated by the validator as invalid. The controller should do the same.

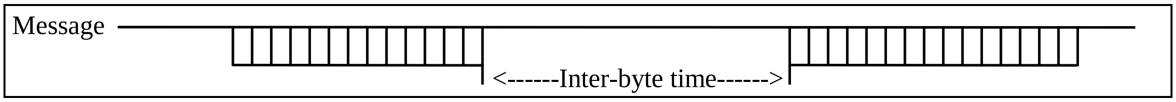


Figure 5-2 Inter-byte Time

5.3 Communication time-out

The validator requires that communication be maintained with the controller. Should the validator be unable to communicate with the controller for a certain period of time, the validator will assume that the communications link with the controller has been lost. The validator will then enter the inhibit state because of ***communication time-out*** and stop accepting currency. In addition any bill in escrow will be returned to the customer. The rules for communication are as follows. In the polled mode the validator must receive a valid command message before a period of 5 seconds has elapsed from the last validator reply. In normal operation, the controller must poll the validator more frequently than once every 5 second (ideally it should poll at least 5 times a second). Note that the time-out is measured from the last validator reply. This means it does not include the time it takes to execute the command as part of the time-out time.

In the interrupt mode it is not necessary to poll the validator, so the 5 second command rule does not apply. When the validator wants the controllers' attention it will send an IRQ message. After the IRQ is sent the validator must receive a valid status request message from the controller within 5 seconds. (Remember that if the controller does not respond to the first IRQ, the validator will send additional IRQ messages every 200 msec.) If the 5 seconds expire with no valid message from the controller, the validator will go into inhibit mode as previously described. In both modes if communication is restored after a time-out has occurred then the validator will automatically Un-inhibit and go back to normal operation.

6. Error handling

One of the main features of this protocol is to allow recovery from communication errors by using the retransmit command. It is important that it be used correctly. Incorrect use of the command will cause a misrepresentation of actual validator status and may result in customer credit being lost, or extra credit being given to the customer. The following is a case by case description of how the controller should handle communication errors.

Case 1: The controller sends a command (other than a status request) and the validator does not respond at all within the specified time or responds with a NAK transmission package. In this case the validator did not receive the command correctly. The validator will ignore the command. No change of status in the validator will take place. The controller should retransmit the original command, it should not use the "retransmit" command. See section 5.2--Validator Response Time.

Case 2: The controller sends a command other then a status command and the validator responds with a message (anything other than a NAK), but the message is not valid. For example the message has an incorrect number of bytes or fails checksum. In this case the validator received the message and understood it, but there was an error when transmitting the answer back to the controller. The controller has two choices in this situation, it can use the retransmit command and asks the validator to repeat the message, or it can send the original command again.

Case 3: The controller sends a status request command and the validator responds with a message, but the message is not valid. Again the validator has received the message and understood it. As it was a status request, the validator will include in the message any events which have occurred since the last status request. The event flags will then be cleared. If the controller repeats the original command it will receive new status, not a repeat of the old. This is a case when the retransmit command must be used. The controller should continue issuing retransmit commands until it receives a valid response, or it becomes obvious that the link between controller and validator has failed.

Case 4: The controller sends a status request and the validator does not respond (message response time-out) the controller should use the retransmit command.

Case 5: The controller sends a message with an undefined command. In other words the validator has received a message with correct word count, format and checksum, but the command byte does not match any valid validator command. The validator will respond with a message with a length of 4 bytes and a command field of 99 hex. This is the invalid command response.

7. Power-Up

7.1 Default

The initial default settings on power up for the validator are:

- Polled communication mode
- Non escrow bill handling mode
- All bills disabled
- All bills set to low security
- Bills accepted in both directions.
- Bills accepted both face-up and face-down (if unit has 4 way acceptance)
- Unit operates only with stacker
- After power up and before enable command, host should execute the Read option command to make the proper bill denomination assignment.

7.2 Credit Recovery

On the power-up, the validator will check if the bill was in the process of being stacked during the last power down. The bill being stacked is defined as time between the bill clears the rear sensor and before the response to a status command with the “Stacked” bit set is sent. If it is the case, then once the validator has been power up, the first status command will be answered with the “bill stacked” bit set along with the “bill type field” containing the bill type of the valid bill. The “power on reset”, “idle” and “communication time-out” flags will be set at same time. Note that this “bill stacked” flag is not preceded by the “accepting” flag nor the “stacking” flag. It is recommended that the controller send the status command as soon as the power being backed up again.

Note that a condition may arise when power goes down at the moment of the validator sending the “stacked” message, the “stacked” message may be lost.

8. Touch Memory Data Format and address allocation

The memory size of the DS1994 memory chip is 512 bytes (4K bits). Base address of information starts at location 0000H and continues to location 01FEH. The last location (1FFH) will be set if the DS1994 has been cleared by PC program.

No.	Purpose	Mem. Addr. (Hex)	# of bytes	Format	Description
1	Time	00, 01, 02	3	BCD	Seconds, Minutes, Hours
2	Day	03	1	BCD	Day of week (Sun=1,..,Sat=7)
3	Date	04, 05, 06	3	BCD	Day, Month, Year
4	Time Attached	07, 08, 09	3	BCD	Seconds, Minutes, Hours
5	Day	0A	1	BCD	Day of week (Sun=1,..,Sat=7)
6	Date	0B, 0C, 0D	3	BCD	Day, Month, Year
7	Asset Number	10 .. 15	6	HEX	MSB-LSB DS2400 number (cable)
8	Bill type 1 Count	20, 21	2	BINARY	MSB-LSB amount of bills accepted
9	Bill type 2 Count	22, 23	2	BINARY	MSB-LSB amount of bills accepted
10	Bill type 3 Count	24, 25	2	BINARY	MSB-LSB amount of bills accepted
11	Bill type 4 Count	26, 27	2	BINARY	MSB-LSB amount of bills accepted
12	Bill type 5 Count	28, 29	2	BINARY	MSB-LSB amount of bills accepted
13	Bill type 6 Count	2A, 2B	2	BINARY	MSB-LSB amount of bills accepted
14	Bill type 7 Count	2C, 2D	2	BINARY	MSB-LSB amount of bills accepted
15	Bill type 8 Count	2E, 2F	2	BINARY	MSB-LSB amount of bills accepted
16	Bill type 9 Count	30, 31	2	BINARY	MSB-LSB amount of bills accepted
17	Bill type 10 Count	32, 33	2	BINARY	MSB-LSB amount of bills accepted
18	Bill type 11 Count	34, 35	2	BINARY	MSB-LSB amount of bills accepted
19	Bill type 12 Count	36, 37	2	BINARY	MSB-LSB amount of bills accepted
20	Bill type 13 Count	38, 39	2	BINARY	MSB-LSB amount of bills accepted
21	Bill type 14 Count	3A, 3B	2	BINARY	MSB-LSB amount of bills accepted
22	Bill type 15 Count	3C, 3D	2	BINARY	MSB-LSB amount of bills accepted
23	Fault No. 1	40	1	BINARY	Bill count for this fault rejection.
.	.	.			
.	.	.			
43	Fault No. 20	53			
44	Denomination Tbl.	60	75		
				HEX	Bill type 1
44	Bill type 1	60	1	HEX	3 digit ASCII
45	ISO country code	61	3	BCD	Upper nibble
48	Denomination	64	1	.	Denomination
			.	.	Lower nibble
.	.	.	.	.	multiplier
.	.	.	.	.	.
.	.	.	.	.	.
117	Bill Type 15	A8	1	HEX	Bill type 15
118	ISO country code	A9	3	HEX	3 digit ASCII
119	Denomination	AA	1	BCD	Upper nibble
					Denomination
					Lower nibble
					multiplier
120	Coupon information	AB .. B7	12	HEX	Number read from coupon

9. Command Description

The following is a description of currently defined commands. New commands may be added to suit specific customer needs. Unused bits in data words sent by the controller should be set to 0's. Unused bits in the validator response data words will be set to 0's, but should be masked off by the controller anyway to allow for expansion of the protocol.

Table 9-1 Example of Note Types

NOTE TYPES		
Bill type	Bill type field	Bill value
Number	[d4-d0]	(dollar, peso, lira, etc.)
1	01	1
2	02	2
3	03	5
4	04	10
5	05	20
6	06	50
7	07	100
8	08	200
9	09	500
10	0A	1,000
11	0B	2,000
12	0C	5,000
13	0D	10,000
14	0E	20,000
15	0F	50,000
Barcode	10	Barcode Coupon

9.1.1 Reset Production Date command (30H)

COMMAND = 30 HEX
DATA WORDS = NONE
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][04][30][CHK=CA]
RESPONSE: [02][04][00][FA]
CLEARS PRODUCTION DATE.

9.1.2 Reset Factory Serial Number command (31H)

COMMAND = 31 HEX
DATA WORDS = NONE
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][04][31][CHK=C9]
RESPONSE: [02][04][00][FA]
CLEARS FACTORY SERIAL NUMBER

9.1.3 Reset Current Statistic command (32H)

COMMAND = 32 HEX
DATA WORDS = NONE
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][04][32][CHK=C8]
RESPONSE: [02][04][00][FA]
CLEARS CURRENT STATISTICS REGISTERS

9.1.4 Reset thousand events statistics command (33H)

COMMAND = 33 HEX
DATA WORDS = NONE
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][04][33][CHK=C7]
RESPONSE: [02][04][00][FA]
CLEARS THOUSAND EVENTS STATISTICS BUFFER

9.1.5 Reset bill counter command (34H)

COMMAND = 34 HEX
DATA WORDS = NONE
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][04][34[CHK=C6]
RESPONSE: [02][04][00][FA]
CLEARS BILL COUNTER BUFFER

9.1.6 Reset unit setup command (35H)

COMMAND = 35 HEX
DATA WORDS = NONE
RESPONSE = AKC
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][04][35][CHK=C5]
RESPONSE: [02][04][00][FA]
CLEARS ALL SETUP ENTRIES RESULTING IN DEFAULT UNIT SETUP

9.1.7 Reset validator command (36H)

COMMAND = 36 HEX
DATA WORDS = NONE
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][04][36][CHK=C4]
RESPONSE: [02][04][00][FA]
PERFORMS SOFT RESETS OF THE VALIDATOR

9.1.8 Set Mode command (40H)

COMMAND = 40 HEX

DATA BYTES = 1

RESPONSE = ACK

MAX RESPONSE TIME = 20 MSEC

COMMAND: [02][05][40][DATA1][CHK]

RESPONSE: [02][04][00][FA]

THIS COMMAND IS USED TO SET THE OPERATIONAL MODE OF THE VALIDATOR. POLLED MODE IS DEFAULT

DATA1:

D7 - D2 NOT USED, SHOULD BE SET TO ZERO.

D1 - COMMUNICATION MODE BIT. IF BIT IS SET INTERRUPT MODE WILL BE USED IN VALIDATOR - CONTROLLER COMMUNICATIONS. IF BIT IS CLEARED POLLED MODE WILL BE USED.

D0 - ESCROW BIT. IF SET VALIDATOR WILL HOLD VALID BILLS IN THE ESCROW POSITION, AWAITING A STACK OR RETURN COMMAND FROM THE CONTROLLER. IF BIT IS CLEARED, BILLS WILL BE MOVED DIRECTLY TO THE STACKER.

9.1.9 Set Bill Enables command (41H)

COMMAND = 41 HEX

DATA WORDS = 2

RESPONDS = ACK

MAX RESPONSE TIME = 20 MSEC

COMMAND: [02][06][41][DATA1][DATA2][CHK]

RESPONSE: [02][04][00][FA]

THIS COMMAND ALLOWS THE CONTROLLER TO SET WHETHER OR NOT EACH BILL TYPE SUPPORTED BY THE VALIDATOR IS ENABLED FOR ACCEPTANCE. ENABLING UNUSED BILL TYPES WILL NOT CAUSE AN ERROR. BITS CORRESPONDING TO UNUSED BILL TYPES WILL BE IGNORED. FOR ALL BILL TYPE A ONE ENABLES THE BILL FOR ACCEPTANCE. A ZERO DISABLES THE BILL. NOTE THAT BILL ENABLES ARE ONLY CHECKED AT THE END OF BILL VALIDATION (END OF ACCEPTING STATE). IF A BILL IS IN ESCROW AND THAT BILL TYPE IS THEN DISABLED THE BILL WILL REMAIN IN ESCROW. DEFAULT IS ALL BILLS DISABLED.

DATA1:

D7 - BARCODE COUPON ENABLE/DISABLE, 1=ENABLE (ONLY IF BARCODE UNIT)

D6 - BILL TYPE 15 ENABLE/DISABLE 1=ENABLE

D5 - BILL TYPE 14 ENABLE/DISABLE 1=ENABLE

D4 - BILL TYPE 13 ENABLE/DISABLE 1=ENABLE

D3 - BILL TYPE 12 ENABLE/DISABLE 1=ENABLE

D2 - BILL TYPE 11 ENABLE/DISABLE 1=ENABLE

D1 - BILL TYPE 10 ENABLE/DISABLE 1=ENABLE

D0 - BILL TYPE 9 ENABLE/DISABLE 1=ENABLE

DATA2:

D7 - BILL TYPE 8 ENABLE/DISABLE 1=ENABLE

D6 - BILL TYPE 7 ENABLE/DISABLE 1=ENABLE

D5 - BILL TYPE 6 ENABLE/DISABLE 1=ENABLE

D4 - BILL TYPE 5 ENABLE/DISABLE 1=ENABLE

D3 - BILL TYPE 4 ENABLE/DISABLE 1=ENABLE

D2 - BILL TYPE 3 ENABLE/DISABLE 1=ENABLE

D1 - BILL TYPE 2 ENABLE/DISABLE 1=ENABLE

D0 - BILL TYPE 1 ENABLE/DISABLE 1=ENABLE

9.1.10 Set Security Level command (42H)

COMMAND = 42 HEX
DATA WORDS = 2
RESPONDS = ACK
MAX RESPONSE TIME = 20 MSEC
COMMAND: [02][06][42][DATA1][DATA2][CHK]
RESPONSE: [02][04][00][FA]
THIS COMMAND ALLOWS THE CONTROLLER TO SET SECURITY LEVEL TO HIGH OR LOW FOR EACH BILL TYPE SUPPORTED BY VALIDATOR. SETTING UNUSED BILL TYPES WILL NOT CAUSE AN ERROR. BITS CORRESPONDING TO UNUSED BILL TYPES WILL BE IGNORED. FOR ALL BILL TYPES, A ONE SETS THE HIGH SECURITY LEVEL, A ZERO SETS THE LOW SECURITY LEVEL. DEFAULT IS SECURITY LEVEL SET TO LOW FOR ALL BILLS .
DATA1:
D7 - UNUSED
D6 - BILL TYPE 15 HIGH/LOW SECURITY 1=HIGH SECURITY
D5 - BILL TYPE 14 HIGH/LOW SECURITY 1=HIGH SECURITY
D4 - BILL TYPE 13 HIGH/LOW SECURITY 1=HIGH SECURITY
D3 - BILL TYPE 12 HIGH/LOW SECURITY 1=HIGH SECURITY
D2 - BILL TYPE 11 HIGH/LOW SECURITY 1=HIGH SECURITY
D1 - BILL TYPE 10 HIGH/LOW SECURITY 1=HIGH SECURITY
D0 - BILL TYPE 9 HIGH/LOW SECURITY 1=HIGH SECURITY
DATA2:
D7 - BILL TYPE 8 HIGH/LOW SECURITY 1=HIGH SECURITY
D6 - BILL TYPE 7 HIGH/LOW SECURITY 1=HIGH SECURITY
D5 - BILL TYPE 6 HIGH/LOW SECURITY 1=HIGH SECURITY
D4 - BILL TYPE 5 HIGH/LOW SECURITY 1=HIGH SECURITY
D3 - BILL TYPE 4 HIGH/LOW SECURITY 1=HIGH SECURITY
D2 - BILL TYPE 3 HIGH/LOW SECURITY 1=HIGH SECURITY
D1 - BILL TYPE 2 HIGH/LOW SECURITY 1=HIGH SECURITY
D0 - BILL TYPE 1 HIGH/LOW SECURITY 1=HIGH SECURITY

9.1.11 Set Orientation command (43H)

COMMAND = 43 HEX
DATA BYTES = 1
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC
COMMAND: [02][05][43][DATA1][CHK]
RESPONSE: [02][04][00][FA]
THIS COMMAND IS USED TO SET THE ORIENTATION AND DIRECTION FOR BILL ACCEPTANCE. NORMALLY, THE VALIDATOR HAS 4 WAY ACCEPTANCE. THAT IS THE UNIT WILL ACCEPT IN TWO DIRECTIONS, (FACE FIRST AND FACE LAST) AND TWO ORIENTATIONS (FACE UP AND FACE DOWN). THE VALIDATOR CAN BE SET TO ACCEPT BILLS IN ONE DIRECTION (PRIMARY DIRECTION ONLY) AND/OR ONE ORIENTATION (FACE UP ONLY). NOT ALL MODELS ALLOW FOUR WAY ACCEPTANCE AND NOT ALL CURRENCIES ALLOW FOUR WAY ACCEPTANCE. FOR UNITS WHICH DO NOT SUPPORT ACCEPTANCE IN A PARTICULAR ORIENTATION OR DIRECTION THE COMMAND BITS WILL BE IGNORED. DEFAULT IS FOUR WAY ACCEPTANCE.
DATA1:
D7-D2 NOT USED, SHOULD BE 0'S
D1 - ORIENTATION BIT, 1 = PRIMARY AND SECONDARY DIR UP / 0 = PRIMARY DIRECTION UP AND DOWN
D0 - DIRECTION BIT, 1 = PRIMARY DIR UP AND DOWN/ 0 = PRIMARY AND SECONDARY DIR UP

9.1.12 Write Production Date command (44H)

COMMAND = 44 HEX

DATA BYTES = 2

RESPONSE = ACK

MAX RESPONSE TIME = 20 MSEC

COMMAND: [02][06][44][DATA1][DATA2][CHK]

RESPONSE: [02][04][00][FA]

THIS COMMAND ALLOWS THE CONTROLLER TO WRITE A 2 BYTE DATE CODE, 1ST BYTE IS WEEK OF THE YEAR AND SECOND BYTE IS YEAR. BOTH BYTES ARE IN BCD FORMAT.

DATA1, DATA2, 2 BYTE DATE CODE

9.1.13 Set Stacker Optional for Operation Mode command (45H)

COMMAND = 45 HEX

DATA WORDS = NONE

RESPONSE = ACK

MAX RESPONSE TIME = 20 MSEC.

COMMAND: [02][04][45][CHK=B5]

RESPONSE: [02][04][00][FA]

SETS FLAG IN THE VALIDATOR DISABLING STACKER CHECKING ROUTINE. VALIDATOR WILL ACCEPT NOTES REGARDLESS OF PRESENCE OF THE STACKER. DEFAULT IS STACKER REQUIRED MODE.

9.1.14 Set Stacker Required for Operation Mode command (46H)

COMMAND = 46 HEX

DATA WORDS = NONE

RESPONSE = ACK

MAX RESPONSE TIME = 20 MSEC.

COMMAND: [02][04][46][CHK=B4]

RESPONSE: [02][04][00][FA]

SETS FLAGS IN THE VALIDATOR REQUIRING STACKER FOR NORMAL OPERATION. IF STACKER IS NOT PRESENT, VALIDATOR WILL INHIBIT, AS SOON AS STACKER IS CONNECTED VALIDATOR WILL UNINHIBIT AND ENTER THE IDLE STATE. DEFAULT IS STACKER REQUIRED FOR NORMAL OPERATION.

9.1.15 Write Factory Serial number command (47H)

COMMAND = 47 HEX

DATA BYTES = 2

RESPONSE = ACK

MAX RESPONSE TIME = 20 MSEC

COMMAND: [02][06][47][DATA1][DATA2][CHK]

RESPONSE: ACK BYTE

THIS COMMAND ALLOWS THE CONTROLLER TO WRITE TWO BYTE FACTORY SERIAL NUMBER

DATA1, DATA2, 2 BYTE SERIAL NUMBER

DATA1 - MSB [UNIB][LNIB]
DATA2 - LSB [UNIB][LNIB]
DATA FORMAT BCD IN DIGITS

9.1.16 Set stacker plate normally open (48H)

COMMAND = 48 HEX

DATA WORDS = NONE

RESPONSE = ACK

MAX RESPONSE TIME = 20 MSEC.

COMMAND: [02][04][48][CHK=B2]

RESPONSE: [02][04][00][FA]

SETS THE RESTING POSITION OF THE STACKER PUSHER PLATE TO THE OPEN POSITION. DEFAULT IS STACKER PLATE NORMALLY OPEN.

9.1.17 Set stacker plate normally closed (49H)

COMMAND = 49 HEX

DATA WORDS = NONE

RESPONSE = ACK

MAX RESPONSE TIME = 20 MSEC.

COMMAND: [02][04][49][CHK=B1]

RESPONSE: [02][04][00][FA]

SETS THE RESTING POSITION OF THE STACKER PUSHER PLATE TO THE CLOSED POSITION FOR MAXIMUM SECURITY AGAINST BILL REMOVAL FROM STACKER. . DEFAULT IS STACKER PLATE NORMALLY OPEN.

9.1.18 Disable Security Level 2 command (4AH)

COMMAND = 4A HEX

DATA WORDS = NONE

RESPONSE = ACK

MAX RESPONSE TIME = 20 MSEC.

COMMAND: [02][04][4A][CHK=B0]

RESPONSE: [02][04][00][FA]

SETS FLAGS IN THE VALIDATOR ENABLING SPECIAL SECURITY FEATURES. DEFAULT IS DISABLE LEVEL 2 SECURITY.

9.1.19 Enable Security Level 2 command (4BH)

COMMAND = 4B HEX

DATA WORDS = NONE

RESPONSE = ACK

MAX RESPONSE TIME = 20 MSEC.

COMMAND: [02][04][4B][CHK=AF]

RESPONSE: [02][04][00][FA]

SETS FLAGS IN THE VALIDATOR DISABLING SPECIAL SECURITY FEATURE. DEFAULT IS DISABLE LEVEL 2 SECURITY.

9.1.20 Write System ID Number command (50H)

COMMAND = 50 HEX
DATA BYTES = 2
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][06][50][DATA1][DATA2][CHK]
RESPONSE: [02][04][00][FA]
THIS COMMAND ALLOWS THE CONTROLLER TO WRITE A 2 BYTE ID NUMBER TO IDENTIFY THE VALIDATOR FOR ACCOUNTING PURPOSES. THIS NUMBER WILL BE STORED IN NON-VOLATILE MEMORY AND MAY BE READ BACK BY THE CONTROLLER. WRITING THIS NUMBER HAS NO EFFECT ON VALIDATOR OPERATION.
DATA1, DATA2, 2 BYTE BINARY ID NUMBER DATA1-MSB, DATA2-LSB

9.1.21 Set Real Time Clock command (51H)

COMMAND = 51 HEX
DATA BYTES = 8
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][0C][51][DATA1][DATA2] .. [DATA8][CHK]
RESPONSE: [02][04][00][FA]
THIS COMMAND ALLOWS THE CONTROLLER TO SET THE VALIDATOR INTERNAL REAL TIME CLOCK (IN UNITS SO EQUIPPED). THIS CLOCK HAS BATTERY BACKUP AND WILL CONTINUE TO KEEP TIME EVEN IF POWER IS REMOVED. FOR UNITS WITHOUT THE RTC THIS COMMAND WILL BE IGNORED.
DATA1:
YEAR, FORMAT IS PACKED BCD, 2 DIGITS (00 - 99)
DATA2:
MONTH, FORMAT IS PACKED BCD, 2 DIGITS, RANGE = (1 TO 12)
DATA3:
DATE, FORMAT IS PACKED BCD, 2 DIGITS, RANGE = (1 TO 31)
DATA4:
DAY OF WEEK, ONE BCD DIGIT IN D3-D0 (SUN = 1 .. SAT = 7)
DATA5:
HOURS, FORMAT IS
D7 - 12/24 HOUR FORMAT, 0 = 24 HOUR FORMAT, 1 = 12 HOUR FORMAT IF 12 HOUR FORMAT:
D6 = AM/PM (1=PM), D4-D0 = HOUR (1 TO 12) IN BCD FORMAT IF 24 HOUR FORMAT
D5 - D0 = HOUR (1 TO 24) IN BCD FORMAT
DATA6:
MINUTES, FORMAT IS PACKED BCD, 2 DIGITS RANGE = (0 TO 59)
DATA7:
SECONDS, FORMAT IS PACKED BCD, 2 DIGITS, RANGE = (0 TO 59)
DATA8:
HUNDREDTHS OF SECONDS, FORMAT PACKED BCD, 2 DIGITS (0 TO 99)

9.1.22 Upload Command (60H)

COMMAND = 60 HEX
DATA WORDS = NONE
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][04][60][CHK=9A]
RESPONSE: FIRST [02][04][00][FA], THEN THE VALIDATOR START SENDING APPLICATION FILE IN CBV FORMAT.
PLEASE NOTE THAT UPLOAD PROCESS DOES NOT FOLLOW V 2 PROTOCOL MESSAGE FORMAT. AS SOON AS UPLOAD ROUTINE ACTIVATES IT STARTS TO SEND STREAM OF DATA IN BINARY FORMAT. PLEASE DO NOT USE MULTIPLE WINDOW IN THE HOST COMPUTER DURING UPLOAD PROCESS.

9.1.23 Download (Network File only)Command (61H)

COMMAND = 61 HEX
DATA WORDS = NONE
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][04][61][CHK=99]
RESPONSE: FIRST [02][04][00][FA], THEN PROGRAM JUMPS TO KERNEL. KERNEL DOES NOT FOLLOW V2 PROTOCOL MESSAGE FORMAT. IF VALIDATOR IS NOT IN IDLE MODE OR THE APPLICATION PROGRAM IS NOT NETWORK FORMATTED, VALIDATOR SENDS NAK TO HOST COMPUTER
PLEASE SEE THE SECTION SERIAL DOWNLOAD FOR MORE DETAIL

9.1.24 Send CRC calculation parameter Command (62H)

COMMAND = 62 HEX
DATA BYTES = 4
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][08][62][DATA1][DATA2] .. [DATA4][CHK]
RESPONSE: [02][04][00][FA]
THIS COMMAND ALLOWS THE CONTROLLER TO SEND TWO CRC CALCULATION PARAMETER TO VALIDATOR. ONE IS TWO BYTE LONG POLYNOMIAL AND ANOTHER ONE IS TWO BYTE LONG SEED
DATA1:
HIGH BYTE OF THE POLYNOMIAL
DATA2:
LO BYTE OF THE POLYNOMIAL
DATA3:
HIGH BYTE OF THE SEED
DATA4:
LO BYTE OF THE SEED
PLEASE SEE THE SECTION ON CRC CALCULATION FOR MORE DETAIL

9.1.25 Request CRC value of the code Command (63H)

COMMAND = 63 HEX
DATA WORDS = NONE
RESPONSE LENGTH= 6 BYTES (2 DATA BYTES)
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][04][63][CHK=97]
RESPONSE: [02][06][63][CRC HI BYTE][CRC LO BYTE][CHK]
THIS COMMAND CAUSES THE VALIDATOR TO SEND CRC TO THE CONTROLLER.
PLEASE SEE THE SECTION CRC CALCULATION FOR MORE DETAIL.

9.1.26 Retransmit Command (77H)

COMMAND = 77 HEX
DATA WORDS = NONE
RESPONSE = VARIABLE
MAX RESPONSE TIME = 20 MSEC.
COMMAND: [02][04][77][CHK=83]
RESPONSE: [THE LAST COMMAND RESPONSE TRANSMITTED BY THE VALIDATOR]
THIS COMMAND IS USED WHEN THE VALIDATOR HAS RESPONDED TO THE PREVIOUS COMMAND, BUT THE RESPONSE IS INVALID. IT REPEATS THE LAST VALIDATOR RESPONSE EXACTLY AS IT WAS TRANSMITTED. SEE SECTION 6 ON ERROR HANDLING.

9.1.27 Accept bill command (80H)

COMMAND = 80 HEX
DATA WORDS = NONE
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC.
COMMAND: [02][04][80][CHK=7A]
RESPONSE: [02][04][00][FA]
THIS COMMAND CAUSES THE VALIDATOR TO ACCEPT (AND STACK) A BILL IN THE ESCROW POSITION OR BILL IS IN THE CHAMBER AT POWER UP.

9.1.28 Return bill command (81H)

COMMAND = 81 HEX
DATA WORDS = NONE
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC.
COMMAND: [02][04][81[CHK=79]
RESPONSE: [02][04][00][FA]
THIS COMMAND CAUSES THE VALIDATOR TO RETURN A BILL IN THE ESCROW POSITION TO THE CUSTOMER OR IF BILL IS IN THE CHAMBER AT POWER UP.

9.1.29 All Illegal commands

COMMAND = INVALID COMMAND

DATA WORDS = DON'T CARE

RESPONSE = 99 HEX

MAX RESPONSE TIME = 20 MSEC.

COMMAND: [02][XX][INVALID COMMAND] .. [CHK]

RESPONSE: [02][04][99][CHK=61]

ANY COMMAND SENT TO THE VALIDATOR WHICH IS NOT A VALID COMMAND CODE WILL BE RESPONDED TO WITH A COMMAND CODE OF 00 HEX. THE MAXIMUM RESPONSE TIME IS 20 MSEC.

9.1.30 Request Mode command (C0H)

COMMAND = C0 HEX

DATA WORDS = NONE

RESPONSE LENGTH = 4 BYTES

MAX RESPONSE TIME = 20 MSEC

COMMAND: [02][04][C0][CHK=3A]

RESPONSE: [02][05][C0][DATA1][CHK]

RETURNS THE CURRENT OPERATIONAL MODE SETTINGS TO THE CONTROLLER.

DATA1:

D7 - D2 NOT USED, SHOULD BE IGNORED BY CONTROLLER.

D1 - CURRENT COMMUNICATION MODE, SET IF INTERRUPT MODE, CLEARED IF POLLED MODE.

D0 - CURRENT BILL HANDLING MODE, SET IF ESCROW MODE, CLEARED IF NON-ESCROW MODE.

9.1.31 Request Bill Enables command (C1H)

COMMAND = C1 HEX

DATA WORDS = NONE

RESPONSE LENGTH = 6 BYTES (2 DATA)

MAX RESPONSE TIME = 20 MSEC

COMMAND: [02][04][C1][CHK=39]

RESPONSE: [02][06][C1][DATA1][DATA2][CHK]

THIS COMMAND RETURNS THE CURRENT SETTINGS OF THE BILLS ENABLED FOR ACCEPTANCE. BILL TYPES WHOSE BITS SET TO ONE ARE ENABLED. BILL TYPES WHOSE BITS SET TO ZERO ARE DISABLED. BILL TYPES NOT SUPPORTED WILL BE REPORTED AS DISABLED, EVEN IF A PREVIOUS SET BILL ENABLES COMMAND ATTEMPTED TO ENABLE THEM.

DATA1:

D7 - BARCODE COUPON ENABLED/DISABLED, 1=ENABLED

D6 - BILL TYPE 15 ENABLED/DISABLED 1=ENABLED

D5 - BILL TYPE 14 ENABLED/DISABLED 1=ENABLED

D4 - BILL TYPE 13 ENABLED/DISABLED 1=ENABLED

D3 - BILL TYPE 12 ENABLED/DISABLED 1=ENABLED

D2 - BILL TYPE 11 ENABLED/DISABLED 1=ENABLED

D1 - BILL TYPE 10 ENABLED/DISABLED 1=ENABLED

D0 - BILL TYPE 9 ENABLED/DISABLED 1=ENABLED

DATA2:

D7 - BILL TYPE 8 ENABLED/DISABLED 1=ENABLED

D6 - BILL TYPE 7 ENABLED/DISABLED 1=ENABLED

D5 - BILL TYPE 6 ENABLED/DISABLED 1=ENABLED

D4 - BILL TYPE 5 ENABLED/DISABLED 1=ENABLED

D3 - BILL TYPE 4 ENABLED/DISABLED 1=ENABLED

D2 - BILL TYPE 3 ENABLED/DISABLED 1=ENABLED

D1 - BILL TYPE 2 ENABLED/DISABLED 1=ENABLED

D0 - BILL TYPE 1 ENABLED/DISABLED 1=ENABLED

9.1.32 Request Security Level command (C2H)

COMMAND = C2 HEX
DATA WORDS = NONE
RESPONSE LENGTH = 6 BYTES
MAX RESPONSE TIME = 20 MSEC
COMMAND: [02][04][C2][CHK=38]
RESPONSE: [02][06][C2][DATA1][DATA2][CHK]
THIS COMMAND RETURNS THE CURRENT SETTINGS OF SECURITY LEVEL FOR ALL BILL TYPES SUPPORTED BY VALIDATOR. FOR ALL BILL TYPE, A ONE SET THE HIGH SECURITY LEVEL, A ZERO SET THE LOW SECURITY LEVEL. DEFAULT IS SECURITY LEVEL SET TO LOW FOR ALL BILLS .
DATA1:
D7 - UNUSED
D6 - BILL TYPE 15 HIGH/LOW SECURITY 1=HIGH SECURITY
D5 - BILL TYPE 14 HIGH/LOW SECURITY 1=HIGH SECURITY
D4 - BILL TYPE 13 HIGH/LOW SECURITY 1=HIGH SECURITY
D3 - BILL TYPE 12 HIGH/LOW SECURITY 1=HIGH SECURITY
D2 - BILL TYPE 11 HIGH/LOW SECURITY 1=HIGH SECURITY
D1 - BILL TYPE 10 HIGH/LOW SECURITY 1=HIGH SECURITY
D0 - BILL TYPE 9 HIGH/LOW SECURITY 1=HIGH SECURITY
DATA2:
D7 - BILL TYPE 8 HIGH/LOW SECURITY 1=HIGH SECURITY
D6 - BILL TYPE 7 HIGH/LOW SECURITY 1=HIGH SECURITY
D5 - BILL TYPE 6 HIGH/LOW SECURITY 1=HIGH SECURITY
D4 - BILL TYPE 5 HIGH/LOW SECURITY 1=HIGH SECURITY
D3 - BILL TYPE 4 HIGH/LOW SECURITY 1=HIGH SECURITY
D2 - BILL TYPE 3 HIGH/LOW SECURITY 1=HIGH SECURITY
D1 - BILL TYPE 2 HIGH/LOW SECURITY 1=HIGH SECURITY
D0 - BILL TYPE 1 HIGH/LOW SECURITY 1=HIGH SECURITY

9.1.33 Request Orientation command (C3H)

COMMAND = C3 HEX
DATA BYTES = NONE
RESPONSE LENGTH = 5 BYTES (1 DATA)
MAX RESPONSE TIME = 20 MSEC
COMMAND: [02][04][C3][CHK=37]
RESPONSE: [02][05][C3][DATA1][CHK]
THIS COMMAND RETURNS TO THE CONTROLLER THE CURRENT VALIDATOR SETTINGS FOR ACCEPTANCE DIRECTION AND ORIENTATION. FOR UNITS WHICH ALLOW ACCEPTANCE IN ONLY ONE DIRECTION AND/OR ORIENTATION THE CORRESPONDING BITS WILL BE REPORTED AS 1'S REGARDLESS OF PREVIOUS CONTROLLER COMMANDS.
DATA1:
D7-D2 NOT USED, SHOULD BE 0'S
D1 - ORIENTATION BIT, 1 = FACE UP ONLY / 0 = FACE UP AND FACE DOWN
D0 - DIRECTION BIT, 1 = FACE FIRST ONLY / 0 = FACE FIRST AND LAST

9.1.34 Request Video Level Setup (C4H)

COMMAND = C4 HEX
DATA WORDS = NONE
RESPONSE LENGTH = 15 BYTES
MAX RESPONSE TIME = 20 MSEC
COMMAND: [02][04][C4[CHK=36]
RESPONSE: [02][0F][C4][DATA1][DATA2][DATA3][DATA4][DATA5][DATA6][DATA7][DATA8][DATA9][DATA10][DATA11][CHK]
RETURNS THE CURRENT D/A SETUP TO THE CONTROLLER.
DATA1 TO DATA11 REPRESENTS RESPECTIVELY D/A SETUP FOR LED 1..7, BLUE LED, SIDE LOOKING SENSOR AND TWO BAR CODE DIODES.

9.1.35 Status command (CCH)

COMMAND CODE = CC HEX

DATA BYTES = NONE

RESPONSE LENGTH = VARIABLE (3 DATA BYTES EXCEPT IF BARCODE COUPON FOUND)

MAX RESPONSE TIME = 20 MSEC

COMMAND: [02][04][CC][CHK=2E]

RESPONSE: [02][07][CC][DATA1][DATA2][DATA3][CHK] (NO BAR CODE COUPON)

OR [02][N+7][CC][DATA1][DATA2][DATA3][N BYTES BARCODE DATA][CHK]

THE STATUS COMMAND IS USED TO DETERMINE THE ACTIVITY OF THE VALIDATOR. THREE DATA BYTES OF STATUS INFORMATION ARE ALWAYS RETURNED. IF A BARCODE COUPON HAS BEEN STACKED OR IS IN THE ESCROW POSITION THEN(N) ADDITIONAL BYTES OF BARCODE INFORMATION WILL BE RETURNED AFTER DATA BYTE 3. FOR UNITS WITHOUT THE BARCODE OPTION THIS MESSAGE WILL ALWAYS HAVE ONLY 3 DATA BYTES.

DATA1:

D7 - BILL STACKED FLAG, SET IF A BILL HAS BEEN STACKED (SET ONCE PER EVENT). IF THIS FLAG IS SET THEN A BILL HAS BEEN IRRETRIEVABLY STACKED OR IF IN A STACKERLESS UNIT MOVED OUT OF THE VALIDATOR INTO THE CHAMBER. THE CONTROLLER MAY NOW GIVE CREDIT. WHEN THIS BIT IS SET THE BILL TYPE FIELD IS VALID.

D6 - BILL RETURNED, SET IF A VALIDATED BILL HAS BEEN RETURNED TO THE CUSTOMER (EJECTED). THIS MAY HAPPEN AS A RESULT OF A RETURN BILL IN ESCROW COMMAND OR THE VALIDATOR BEING UNABLE TO MOVE THE BILL INTO THE STACKER. IF THIS BIT IS SET THE BILL TYPE FIELD IS VALID. SET ONCE PER OCCURRENCE

D5 - BILL IN ESCROW FLAG, SET IF A BILL IS BEING HELD IN THE CHANNEL IN THE ESCROW POSITION. THE CONTROLLER MAY COMMAND THE BILL TO BE STACKED OR RETURNED. IF THIS BIT IS SET THE BILL TYPE FIELD IS VALID. WILL REMAIN SET AS LONG AS BILL IS IN ESCROW POSITION.

D4-D0 - BILL TYPE FIELD. THIS FIELD IS VALID IF EITHER THE BILL STACKED, BILL RETURNED, OR BILL IN ESCROW FLAGS ARE SET. IT IS NOT VALID AT OTHER TIMES. IN GENERAL THE BILL TYPE CORRESPONDS TO THE DENOMINATIONS IN THE TABLE 1.1. IN SOME CASES IT MAY BE NECESSARY TO REDEFINE THE BILL TYPES FOR A PARTICULAR COUNTRY OR APPLICATION. AS AN EXAMPLE, A COUNTRY HAS BOTH A NEW AND OLD 20 DOLLAR BILL, BOTH OF WHICH NEED BE TREATED SEPARATELY. ANY DEVIATIONS FROM THE TABLE WILL BE NOTED IN THE DOCUMENTATION FOR THAT VALIDATOR. NOT ALL BILL TYPES WILL BE SUPPORTED IN ALL UNITS.

ALTERNATE USE OF D4-D0

IF D4(REJECTING FLAG) OF DATA2 IS SET THEN D4-D0 OF DATA1 BECOME ERROR CODE. SO THE ERROR CODE VALUE COULD RANGE ANYWHERE FROM 0 TO 1FH(31 DECIMAL).

D4-D0
07H - BILL FAILS OPTICAL TEST
08H - BILL DENOMINATION DISABLED
0CH - BILL FAILS MAGNETIC TESTS
0EH - MAXIMUM CREDIT STORED OR UNIT STRUNG
0FH - BILL FAILS BLUE RATIO TEST
10H - ALL BILLS INHIBITED
11H - BILL CAN NOT BE STACKED
12H - BILL FAIL HIGH SECURITY TEST

DATA2:

D7 - CHEATED FLAG, SET IF THE VALIDATOR SENSES AN ATTEMPT TO TAMPER WITH A BILL IN ESCROW. IF THIS OCCURS THE BILL WILL AUTOMATICALLY EJECTED, (VALIDATOR ENTERS RETURNING STATE). SET ONCE PER OCCURRENCE.

D6 - POWER ON RESET FLAG, SET IF THE VALIDATOR HAS GONE THROUGH A POWER ON RESET SEQUENCE. SET ONCE AT POWER UP. AN OCCURRENCE AT ANY TIME AFTER POWER-UP INDICATES A MOMENTARY VALIDATOR POWER FAILURE.

D5 - INVALID ESCROW REQUEST FLAG, SET IF AN ATTEMPT WAS MADE TO STACK OR EJECT A BILL FROM ESCROW, (ACCEPT BILL OR EJECT BILL COMMAND) WHEN THERE WAS NO BILL IN THE ESCROW POSITION.

D4 - REJECTING FLAG, SET IF THE VALIDATOR IS CURRENTLY IN THE PROCESS OF EJECTING A INVALID BILL. (REJECTING STATE).

D3 - STACKING FLAG, SET IF THE VALIDATOR IS IN THE PROCESS OF MOVING THE BILL INTO THE STACKER AND STACKING THE BILL. (STACKING STATE).

D2 - RETURNING FLAG, SET IF THE VALIDATOR IS IN THE PROCESS OF RETURNING A BILL TO THE CUSTOMER. (RETURNING STATE).

9.1.35 Status command (CCH)(Continued)

D1 - ACCEPTING FLAG, SET IF THE VALIDATOR IS IN THE ACCEPTING STATE.

D0 - IDLE FLAG, SET IF THE VALIDATOR IS IN THE IDLE STATE.

DATA 3:

D7 - INHIBITED FLAG, SET IF THE VALIDATOR IS INHIBITED FROM ACCEPTING CURRENCY BY A COMMAND FROM THE CONTROLLER OR BY EXTERNAL HARDWARE. DOES NOT INDICATE A FAULT CONDITION. NOTE THAT THIS BIT WILL NOT BE SET UNTIL THE VALIDATOR ACTUALLY ENTERS THE INHIBITED STATE. IF THE VALIDATOR IS BUSY, SAY RETURNING A BILL, THIS BIT WILL NOT BE SET UNTIL THE VALIDATOR IS FINISHED RETURNING THE BILL.

D6 - VALIDATOR DISABLED. THIS BIT IS SET IF A CONDITION EXISTS WHERE THE VALIDATOR CAN NOT FUNCTION AND MANUAL INTERVENTION IS REQUIRED. THIS BIT WILL BE SET IN A STACKER JAM OR BILL JAM CONDITION AFTER ALL ATTEMPTS BY THE VALIDATOR TO CLEAR THE JAM HAS FAILED. IT IS ALSO SET IF THE STACKER IS OPEN OR REMOVED OR AN UNCORRECTABLE HARDWARE FAULT IS DETECTED.(IN STACKER MODE)

D5 - COMMUNICATION TIME-OUT FLAG, SET IF THE VALIDATOR HAS NOT RECEIVED A POLLING MESSAGE (POLL MODE) OR AN ANSWER TO ITS IRQ BYTE (INTERRUPT MODE) IN THE REQUIRED TIME AND THE VALIDATOR HAD ENTERED ITS TIME-OUT INHIBIT. NOTE THAT THIS FLAG MAY BE SET AT THE FIRST POLL AFTER POWER ON, AND SHOULD BE IGNORED AT THAT TIME.

D4 - POWER UP WITH BILL IN CHANNEL. THE VALIDATOR HAS POWERED UP WITH A BILL IN THE CHANNEL. THE VALIDATOR WAITS 5 SECONDS FOR AN ACCEPT OR EJECT BILL . IF NO MESSAGE IS RECEIVED THE UNIT WILL ATTEMPT TO EJECT THE BILL.

D3 - STACKER FULL SET IF THE STACKER IS UNABLE TO COMPLETE IT'S CYCLE OF PUSHING A BILL INTO THE STACKER DUE TO STACKER FULL CONDITION.

D2 - STACKER OFF/OPEN, SET IF THE STACKER HAS BEEN REMOVED FROM THE MACHINE. FOR STACKERLESS UNITS THIS BIT IS NOT USED AND WILL ALWAYS BE SET TO ZERO.

D1 - STACKER JAM, SET IF THE STACKER IS UNABLE TO COMPLETE IT'S CYCLE OF PUSHING A BILL INTO THE STACKER. NOTE THAT IF A VALID BILL WAS PUSHED ONTO THE STACKER THE BILL STACKED FLAG WILL BE SET. THE CONTROLLER SHOULD GIVE CREDIT FOR THE BILL EVEN THOUGH THE STACKER HAS JAMMED.

D0 - BILL JAM, SET IF A BILL IS STUCK IN THE VALIDATOR CHANNEL. THE VALIDATOR WILL MAKE SEVERAL ATTEMPTS TO RETURN THE BILL TO THE CUSTOMER. IF THE VALIDATOR IS ABLE TO CLEAR THE JAM, OR THE JAM IS CLEARED MANUALLY THEN THE BIT WILL BE CLEARED, AND THE VALIDATOR RETURN TO NORMAL OPERATION.

DATA4 TO DATA(N+4): BARCODE INFORMATION.

N BYTES OF BARCODE DATA SENT IN PACKED BCD FORMAT. DATA IS SENT MOST SIGNIFICANT BYTE FIRST. THIS INFORMATION WILL ONLY BE SENT IF THE BILL TYPE FIELD INDICATES A BARCODE COUPON IS PRESENT AND EITHER THE BILL STACKED, OR BILL IN ESCROW FLAG IS SET.

9.1.36 Request System ID Number command (D0H)

COMMAND = D0 HEX

DATA BYTES = NONE

RESPONSE LENGTH = 6 BYTES (2 DATA)

MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE
NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE

COMMAND: [02][04][D0][CHK=2A]

RESPONSE: [02][06][D0][DATA1][DATA2][CHK]

THIS COMMAND RETURNS TO THE CONTROLLER A PREVIOUSLY WRITTEN SYSTEM ID NUMBER. IF NO NUMBER WAS EVER WRITTEN TO THE VALIDATOR THE DATA RETURNED IS INDETERMINATE.

DATA1-MSB, DATA2-LSB. FORMAT BINARY

9.1.37 Read Real Time Clock command (D1H)

COMMAND = D1 HEX

DATA BYTES = NONE

RESPONSE LENGTH = 12 BYTES (8 DATA)

MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE
NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE

COMMAND: [02][04][D1][CHK=29]

RESPONSE: [02][0C][D1][DATA1][DATA2] .. [DATA8][CHK]

THIS COMMAND RETURNS TO THE CONTROLLER THE CURRENT VALUE OF THE VALIDATOR RTC. IF THE UNIT IS NOT EQUIPPED WITH A RTC ALL OF THE DATA BYTES RETURNED WILL BE ZEROS.

FORMAT - SEE SET REAL TIME CLOCK COMMAND:

9.1.38 Request Validator Serial Number command (D2H)

COMMAND = D2 HEX

DATA BYTES = NONE

RESPONSE LENGTH = 8 BYTES (4 DATA)

MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE
NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE

COMMAND: [02][04][D2][CHK=28]

RESPONSE: [02][08][D2][DATA1][DATA2][DATA3][DATA4][CHK]

THIS COMMAND RETURNS THE FACTORY ASSIGNED SERIAL NUMBER OF THE VALIDATOR.

DATA1 TO DATA4 - 8 DIGIT SERIAL NUMBER, SENT IN PACKED BCD FORMAT MSB FIRST.

9.1.39 Request Asset Number command (D3H)

COMMAND = D3 HEX

DATA BYTES = NONE

RESPONSE LENGTH = 8 BYTES (4 DATA)

MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE
NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE

COMMAND: [02][04][D3][CHK=27]

RESPONSE: [02][08][D3][DATA1][DATA2] .. [DATA6][CHK]

THIS COMMAND RETURNS THE ASSET NUMBER ASSIGNED. THIS ASSET NUMBER IS CONTAINED IN THE DS2400 CHIP. FOR INSTALLATIONS WITHOUT THIS FEATURE THIS COMMAND WILL RETURN ALL ZEROS.

DATA1 - DATA6, 6 BYTES OF BINARY DATA SENT MSB FIRST.

9.1.40 Request Factory Number command (D4H)

COMMAND = D4HEX

DATA BYTES = NONE

RESPONSE LENGTH = 6 BYTES (2 DATA)

MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE
NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE

COMMAND: [02][04][D4[CHK=26]

RESPONSE: [02][06][D4][DATA1][DATA2] [CHK]

THIS COMMAND RETURNS THE FACTORY SERIAL NUMBER.

DATA1 - DATA2, 2 BYTES OF PACKED BCD DATA SENT MSB FIRST.

9.1.41 Request Production Date Code command (D5H)

COMMAND = D5 HEX

DATA BYTES = NONE

RESPONSE LENGTH = 6 BYTES (2 DATA)

MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE
NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE

COMMAND: [02][04][D5][CHK=25]

RESPONSE: [02][06][D5][DATA1][DATA2][CHK]

THIS COMMAND RETURNS THE VALIDATOR DATE OF MANUFACTURE.

DATA1-WEEK OF THE YEAR

DATA2- YEAR

9.1.42 Request Program Revision command (D8H)

COMMAND = D8 HEX

DATA WORDS = NONE

RESPONSE LENGTH = 13 BYTES (9 DATA)

MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE
NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE

COMMAND: [02][04][D8][CHK=22]

RESPONSE: [02][0D][D8][DATA1][DATA2] .. [DATA9][CHK]

THIS COMMAND RETURNS THE PROGRAM REVISION FOR THE VALIDATOR.

DATA1 - DATA8, 8 ASCII CHARACTERS (RIGHT TO LEFT) EX "CRM245 "UNUSED CHARACTERS ARE SENT AS SPACES (20 HEX)

9.1.43 Request Last 5 Transactions command (D9H)

COMMAND = D9 HEX

DATA WORDS = NONE

RESPONSE LENGTH = 9 (5 DATA)

MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE
NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE

COMMAND: [02][04][D9][CHK=21]

RESPONSE: [02][09][D9][DATA1][DATA2] .. [DATA5][CHK]

RETURNS THE BILL TYPE CODES FOR THE LAST 5 BILLS ACCEPTED.

IF LESS THEN 5 BILLS ACCEPTED THEN UNUSED BYTES WILL BE 0.

DATA1 - DATA5, BILL TYPE CODES (1 - 10 HEX) FOR LAST 5 BILL TYPES ACCEPTED. ONE CODE PER BYTE.

9.1.44 Request Life Time Counter command (DAH)

COMMAND = DA HEX

DATA WORDS = NONE

RESPONSE LENGTH = 7(3 DATA)

MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE
NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE

COMMAND: [02][04][DA][CHK=20]

RESPONSE: [02][07][DA][DATA1][DATA2] [DATA3] [CHK]

RETURNS BILL COUNTER INFORMATION. FORMAT BINARY MSB FIRST

TOTAL AMOUNT OF BILLS IS A SUM OF ALL 3 BYTES

9.1.45 Request Fault Buffer from Stacker command (DBH)

COMMAND = DB HEX

DATA WORDS = NONE

RESPONSE LENGTH = 24(20 DATA)

MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE
NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE

COMMAND: [02][04][DB][CHK=1F]

RESPONSE: [02][18][DB][DATA1].......[DATA20] [CHK]

RETURNS 20 BYTES OF ERROR CODES IN ORDER SPECIFIED BY FAULT TABLE. EVERY BYTE HOLDS NUMBER OF REJECTIONS SPECIFIED BY THE TABLE ENTRIES. FORMAT BINARY.

9.1.46 Request Stacked Bill Information command (DCH)

COMMAND = DC HEX

DATA WORDS = NONE

RESPONSE LENGTH = 34(30 DATA)

MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE
NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE

COMMAND: [02][04][DC][CHK=1E]

RESPONSE: [02][22][DC][DATA1].......[DATA30] [CHK]

RETURNS INFORMATION ABOUT BILLS STORED IN THE STACKER.

TWO BYTES PER DENOMINATION: MSB, LSB, STARTING WITH BILL TYPE 1

9.1.47 Request Current Statistics command (DDH)

COMMAND = DD HEX

DATA WORDS = NONE

RESPONSE LENGTH = 76(72DATA)

MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE
NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE

COMMAND: [02][04][DD][CHK=1D]

RESPONSE: [02][4C][DD][DATA1].....[DATA72] [CHK]

RETURNS FULL INFORMATION ABOUT ACCEPTANCE PERFORMANCE OF THE UNIT. STATISTICS ARE BASED ON CURRENT PERFORMANCE, THAT IS UP TO 1000 EVENTS(SUCCESSFUL AND UNSUCCESSFUL TRANSACTIONS) AND INCLUDE COUNT REGISTER OF DENOMINATION ACCEPTED AND COUNT REGISTER OF REJECTION CODES.
EACH COUNT REGISTER IS REPRESENTED BY 2 BYTES(MSB,LSB). FIRST 2 BYTES COUNTER, 15 BILL TYPES REPRESENTED BY 30 BYTES AND 40 BYTES OF FAULT LOCATION, TOTAL IS 72 BYTES.
SEE APPENDIX B FOR EXAMPLE

9.1.48 Request Thousand Attempts Statistics command (DEH)

COMMAND = DE HEX

DATA WORDS = NONE

RESPONSE LENGTH = 76(72DATA)

MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE
NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE

COMMAND: [02][04][DE][CHK=1C]

RESPONSE: [02][4C][DE][DATA1].....[DATA72] [CHK]

RETURNS FULL INFORMATION ABOUT ACCEPTANCE PERFORMANCE OF THE UNIT. STATISTICS ARE BASED ON 1000 EVENTS (SUCCESSFUL AND UNSUCCESSFUL TRANSACTIONS) AND INCLUDE COUNT REGISTER OF REJECTION CODES.
EACH COUNT REGISTER IS REPRESENTED BY 2 BYTES(MSB,LSB). FIRST IS BILL TYPE 1
SEE APPENDIX FOR EXAMPLE.

9.1.49 Request validator setup command from memory (E0H)

COMMAND = E0 HEX
DATA WORDS = NONE
RESPONSE LENGTH = 6 BYTES
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][04][E0][CHK=1A]
RESPONSE: [02][06][E0][DATA1][DATA2][CHK]
RETURNS VALIDATOR SETUP FROM MEMORY. PLEASE NOTE THAT IF SET DEFAULT COMMAND(F3 HEX) HAS NEVER BEEN SET, OR IT WAS SET AND LATER HAS BEEN CLEARED, THE RESPONSE DATA WILL BE 0,0. THAT IS THE VALIDATOR DOES NOT HAVE ANY PRE-SET DEFAULT SETUP PERMANENTLY STORED IN THE MEMORY.
DATA1:
D7 - CLEAR IF INHIBIT LOGIC REGULAR, SET IF INHIBIT LOGIC REVERSE MODE.
D6 - CLEAR IF VALIDATOR IS UNINHIBITED, SET IF VALIDATOR IS INHIBITED.
D5 - CLEAR IF STACKER MODE NORMALLY OPEN, SET IF STACKER MODE NORMALLY CLOSED
D4 - CLEAR IF STACKER IS REQUIRED FOR OPERATION, SET IF STACKER IS NOT REQUIRED
D3 - CLEAR IF VALIDATOR COMMUNICATE IN POLLING MODE, SET IF COMMUNICATE IN INTERRUPT MODE
D2 - CLEAR IF VALIDATOR IN NON-ESCROW MODE, SET IF IN ESCROW MODE
D1 - CLEAR IF CURRENCY ACCEPTANCE IS BOTH WAY, SET IF ACCEPTANCE IS ONE WAY.
D0 - CLEAR IF CURRENCY INSERTION IS BOTH SIDES, SET IF INSERTION IS FACE UP.
DATA 2
D0 - CLEAR IF SECURITY LEVEL 2 IS DISABLED, SET IF SECURITY LEVEL2 ENABLED.

9.1.50 Request current validator setup command (E1H)

COMMAND = E1 HEX
DATA WORDS = NONE
RESPONSE LENGTH = 6 BYTES
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][04][E1][CHK=19]
RESPONSE: [02][06][E1][DATA1][DATA2][CHK]
RETURNS CURRENT VALIDATOR SET UP TO CONTROLLER.
DATA1:
D7 - CLEAR IF INHIBIT LOGIC REGULAR, SET IF INHIBIT LOGIC REVERSE MODE.
D6 - CLEAR IF VALIDATOR IS UNINHIBITED, SET IF VALIDATOR IS INHIBITED.
D5 - CLEAR IF STACKER MODE NORMALLY OPEN, SET IF STACKER MODE NORMALLY CLOSED
D4 - CLEAR IF STACKER IS REQUIRED FOR OPERATION, SET IF STACKER IS NOT REQUIRED
D3 - CLEAR IF VALIDATOR COMMUNICATE IN POLLING MODE, SET IF COMMUNICATE IN INTERRUPT MODE
D2 - CLEAR IF VALIDATOR IN NON-ESCROW MODE, SET IF IN ESCROW MODE
D1 - CLEAR IF CURRENCY ACCEPTANCE IS BOTH WAY, SET IF ACCEPTANCE IS ONE WAY.
D0 - CLEAR IF CURRENCY INSERTION IS BOTH SIDES, SET IF INSERTION IS FACE UP.
DATA 2
D0 - CLEAR IF SECURITY LEVEL 2 IS DISABLED, SET IF SECURITY LEVEL2 ENABLED.

9.1.51 Request validator options command (E2H)

COMMAND = E2 HEX
DATA WORDS = NONE
RESPONSE LENGTH = 76 BYTES
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][04][E2][CHK=18]
RESPONSE: [02][80][E2][DATA1].....[DATA76][CHK]
RETURNS INFORMATION ABOUT THE VALIDATOR CAPABILITIES.
DATA1:
D7 - D3 NOT USED, SHOULD BE IGNORED BY CONTROLLER.
D2 - CLEAR IF BAR-CODE IS NOT SUPPORTED, SET IF BAR-CODE SUPPORTED.
D1 - CLEAR IF REAL TIME CLOCK ABSENT, SET IF REAL TIME CLOCK PRESENT.
D0 - CLEAR IF ASSET NUMBER NOT SUPPORTED, SET IF SUPPORTED.
DATA2 ..... DATA76 AS FOLLOWS
DATA2:
BILL TYPE, VALUE RANGES FROM 1 TO 15.
DATA3,DATA4,DATA5:
THREE ASCII CHARACTERS IDENTIFYING THE COUNTRY FOR WHICH THIS VALIDATOR ACCEPT CURRENCY. EXAMPLE(USD =UNITED STATES, AUD = AUSTRALIA ETC.)
DATA6:
BILL DENOMINATION INFORMATION. UPPER NIBBLE MOST SIGNIFICANT DIGIT OF THE VALUE OF THE BILL TYPE, LOWER NIBBLE - NUMBER OF ZEROS AFTER FIRST DIGIT. EXAMPLE 12=$100, 51=$50, 10=$1 ETC.
DATA 7 TO DATA76 REPEATS ITSELF SAME AS DATA2 TO DATA6.
REFER TO APPENDIX A -- COUNTRY AND CURRENCY CODE

9.1.52 Inhibit command (F0H)

COMMAND = F0 HEX
DATA WORDS = NONE
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC
COMMAND: [02][04][F0][CHK=0A]
RESPONSE: [02][04][00][FA]
INHIBITS THE VALIDATOR FROM ACCEPTING CURRENCY. NOTE THAT BILLS WHICH ARE IN ESCROW OR BEING STACKED ARE NOT AFFECTED. IF A BILL IS BEING HELD IN ESCROW WHEN THIS COMMAND IS RECEIVED IT WILL REMAIN IN ESCROW. IF A BILL IS BEING STACKED THE PROCESS WILL CONTINUE, BUT NO ADDITIONAL NOTES WILL BE ACCEPTED. (SEE INHIBITED STATE). IF THIS COMMAND IS SENT DURING THE BILL PROCESSING STAGE (ACCEPTING, STACKING OR ESCROW), THE VALIDATOR WILL SEND ACK BACK, BUT WILL NOT PROCESS THE COMMAND UNTIL THE BILL PROCESSING IS FINISHED. SEE SECTION 5.1 VALIDATOR RESPONSE TIME.

9.1.53 Uninhibit command (F1H)

COMMAND = F1 HEX
DATA WORDS = NONE
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][04][F1][CHK=09]
RESPONSE: [02][04][00][FA]
THE VALIDATOR WILL BE TAKEN OUT OF THE INHIBITED STATE AND ACCEPT CURRENCY.

9.1.54 Change Default command (F3H)

COMMAND = F3 HEX
DATA WORDS = NONE
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][04][F3][CHK=07]
RESPONSE: [02][04][00][FA]
THIS COMMAND ALLOWS THE USER TO SAVE THE CURRENT SETTINGS OF THE VALIDATOR AS THE POWER-UP DEFAULT SETTINGS. THIS INFORMATION INCLUDES BILL ENABLES, COMMUNICATION MODES, DIRECTION OPTIONS, AND BILL HANDLING (ESCROW) MODES. THE CONTROLLER SHOULD FIRST SET UP THE VALIDATOR AS DESIRED AND THEN SEND THIS COMMAND TO SAVE THE OPTIONS IN BATTERY POWERED MEMORY. SPECIFIED OPTIONS ARE STORED IN MEMORY.

9.1.55 Set Inhibit Logic Regular (F4H)

COMMAND = F4 HEX
DATA BYTES = NONE
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][04][F4][CHK=06]
RESPONSE: [02][04][00][FA]
THIS COMMAND IS USED TO SET THE VALIDATOR INHIBIT LOGIC TO REGULAR MODE.

9.1.56 Set Inhibit Logic Reverse (F5H)

COMMAND = F5 HEX
DATA BYTES = NONE
RESPONSE = ACK
MAX RESPONSE TIME = 20 MSEC IF VALIDATOR IS NOT IN THE BILL PROCESSING STAGE NO RESPONSE IF VALIDATOR IS IN THE BILL PROCESSING STAGE
COMMAND: [02][04][F5][CHK=05]
RESPONSE: [02][04][00][FA]
THIS COMMAND IS USED TO SET THE VALIDATOR INHIBIT LOGIC TO REVERSE MODE.

10. CRC Calculation

The polynomial code also known as cyclic redundancy code or CRC code is widely used in signature checking of any file. This section will only illustrate how to utilize CRC in V2 protocol, it is assumed that the user already has fundamental mathematical understanding about CRC calculation. The steps which take place in IDS to calculate CRC are summarized below.

A. Two parameters are sent from the host to the validator. One selects which of two polynomials are used and the other is the SEED. For CRC-16 ($X^{16} + X^{15} + X^{2} + 1$) it is 0x8005 in hex or for CCITT ($X^{16} + X^{12} + X^{5} + 1$) it is 0x1021 in hex. The host can send either polynomial value to validator. Seed value can be anywhere from 0 to FFFF Hex. Total number of data bytes including polynomial and the SEED is 4. The command package from the host to the controller will be [02][08][62][data1][data2][data3][data4][CHK]. Example host is sending CRC_16 polynomial and SEED value to be 6090 Hex to validator. The transmission package will be [02][08][62][80][05][60][90][1F].

B. As soon as the validator receives these values from the host, it will generate a CRC table look up by using the supplied polynomial and CRC hardware mathematical model. Thereupon, it will do the CRC calculation for each byte from program code by using the table look up. The program code will be transformed to CBV format prior to the CRC calculation process. The SEED will be exclusive ored (XORed) with each CRC value being calculated. The value of the seed will change with every byte of the calculated CRC code data. This entire process will proceed till the end of the code space has been reached.

C. The host will have to wait at least 10 seconds while the calculation process is in progress.

D. The host then requests the validator to send calculated CRC value. The transmission package from the validator will be [02][06][63][data1][data2][CHK].

The CRC has proven to be a versatile and dependable way of validating the signature of a file. The SEED works as a password. There are 65535 possible combinations for this value. A small change in the SEED will create a large change in the CRC value. Due to this fact, it is strongly recommended that the user document the SEED and polynomial values chosen before each CRC check. This will guarantee that the comparison is based on the same values used previously.

11. Serial DownLoad(Network File Only)

11.1 General Description

As mentioned earlier <u>this option is only applicable to a network format file</u>. The Serial Downloader is a program resident in all IDS validators. Through this program the validator application program (the program which controls the actual acceptance of currency) is loaded into the validator. The Serial Download program allows the validator application software to be upgraded without the disassembly of the unit or replacement of a PROM. The program also allows access to unit data stored in nonvolatile memory (such as validator performance data). Finally the serial download program provides a method of securing the validator by use of a password. This prevents unauthorized personnel from loading a new application program.

The Serial network Download program may be entered in one of two ways. On power up, if the dip switch 7, 8 and 9 are ALL in the ON position, the validator will enter the download mode. The serial downloader can be entered from the application program, if it is a network application program, by sending "[02][04][60][chk]" command to IDS. When the Downloader is entered using this method the settings of the DIP switches are ignored. The Downloader can only be exited if no new command is received in 20 seconds. At the end of 24 seconds with no command, the validator resets. If there is an error during download, the validator will not start the application program; it will remain in Downloader program. After the successful download, program will wait for 2 second for any other command to execute; if there is no command, Downloader will start the application program.

11.2 Communication format

In general the serial download program communicates in ASCII characters. All commands are sent as strings of characters terminated by a carriage return character (char = 13). Linefeed characters are ignored by the command interpreter and so are optional when sending a command line. The Downloader will echo back each character as it is input. Note that lower case letters are automatically converted to upper case by the Downloader. The program load and program verify operations are the only ones which are not entirely ASCII based. These functions use a binary file format to send program data to the validator. The character format for data transfer is:

Speed -- 9600 baud
Data -- 8 bits
Stop bits --1
Parity -- none

11.3 Password protection

Password protection is provided to prevent unauthorized personnel from changing the vaildator program. If password protection is active the user is required to enter a password before entering the load command. Failure to do so will cause the load to immediately abort and an error message sent (see the section on error codes). Note that the password is not required to verify the current program (verify command), only to load a new program. Passwords are set using the SP command. A password must be between 1 and 16 characters long and contain only the letters

A-Z, (note that case is not significant, "a" is treated the same as "A"), the numbers 0-9, and the following special characters :;<=>?@ .

11.4 Commands

The serial download program provides the following commands, commands are not case sensitive.:

Command	**- Function**
R <cr>	- Read data from validator memory
RB <cr>	- Read a block of data from data memory
W<cr>	- Write data to data memory
SP<cr>	- Set password (Sets a new password)
PW<cr>	- Enter password (Enters the current password)
LOADN <cr>	- Begin application program download in CBV format
VE<cr>	- Begin application program verification
ID<cr>	- Read program identification number
EX<cr>	- Exit the download program and vector to the application

Detailed command description:

The following is a detailed description of all of the Downloader commands. Note that when entering parameters, bytes must always be entered as 2 hex characters (for example a 2 must be entered as 02). The maximum number of characters on a line is 36. If this limit is exceeded an error message will be generated and the line input aborted. All data bytes and address are in hex.

11.4.1 Read Data Memory Command (R<cr>):

Function: Reads a byte from memory

Format: "R aaaa" where aaaa is the data memory address to be read.

Returns: "bb" where bb is the data byte at that memory address

Range: The address may range from the partition address up to 7FEFH. The area below the partition address is designated as program memory and so can not be accessed. Address 7FF0h to 7FFFh is reserved.

11.4.2 Read Data Block command (RB<cr>):

Function : Reads a block of bytes from the data memory

Format: "RB nn aaaa" where nn is the number of bytes in hex to read (n>=1), and aaaa is the data memory address for the start of that block.

Returns: "bb bb bb bb ... bb" a string of bytes separated by spaces. After every 16 bytes a

<CR><LF> will be inserted instead of the space. This produces rows of data each with 16 values.

Range : The address may range from the partition address up to 7FEFH. The area below the partition address is designated as program memory and so can not be accessed. Address 7FF0h to 7FFFh is reserved.

11.4.3 Write Data Memory command(W<cr>):

Function: Write a byte of data or a string of bytes to memory.

Format: "W aaaa bb[bb ... bb] " where aaaa is the data memory address to start the write operation at, and bb is the data to be written. If a string of bytes is specified the additional bytes will be written in consecutive memory addresses starting at the specified address. Additional bytes may be separated by spaces if desired.

Returns: "OK" if operation was successful

Range: The address may range from the partition address up to 7EFFH. The area below the partition address is designated as program memory and so can not be accessed. Address 7F00h to 7FFFh is reserved.

Examples: W 7000 121314 writes 12 to 7000h, 13 to 7001h and 14 to 7002h. W 7000 12 13 14 does the same thing as above.

11.4.4 Set Password command(SP<cr>):

Function: Enters a new password.

Format: "SP" ccccc where ccccc is a string of 1 to 16 characters containing only upper and lower case letters (A-Z, a-z), numbers (0-9) and :;<=>?@ . After this command is processed the vaildator will respond with "SP>". At this point the user will be required to retype the password for verification purposes. If the password verifies the vaildator will respond with "OK". Password protection becomes effective upon exit from the Downloader. There is a special form of this command, "SP *". This deletes the current password and thus removes password protection from the unit. In either cases, user must submit the old password before erase.

Example:
```
SP my:password       <- Set password command
SP>my:password       <- password is verified
OK                   <- operation complete
```

11.4.5 Input password command(PW<cr>):

Function: Input the current password so that a program load can be done, or a new password entered.

Format: "PW ccccc" where ccccc is the password.

Response: "OK" if password was correct, E:15 if incorrect.

11.4.6 Load program command (LOADN<cr>):

Function: Begin program load sequence.

Format: "LOADN"

Operation :

1) The host computer sends the load command to the vaildator ("LOADN" followed by a <CR>).
2) The host computer waits for an STX char from the validator.
3) The validator sends an STX char (02 hex) to signal it is ready to receive binary data.
4) The host computer sends the CBV format data file out the serial port to the validator. The binary data is not echoed back to host computer. The maximum time allowed between bytes for the download is 7 seconds. A pause between bytes greater then this will cause a time-out fault.
5) After receiving the last byte the validator will respond within 200 MSec with a "LS" if the load was successful. If an error occurs validator immediately aborts the download and returns to the ASCII command mode.

11.4.7 Verify program commandVE<cr>):

Function: Begin program verifying.

Format: "VE"

Operation: The program verify sequence is almost the same as the download:

1) The host computer sends the verify command to the validator ("VE" followed by a <CR>).
2) The host computer waits for an STX char from the vaildator.
3) The validator sends an STX char (02 hex) to signal it is ready to receive binary data.
4) The host computer sends the CBV format data file out the serial port to the validator. The binary data is not echoed back to host computer. The maximum time allowed between bytes for the download is 7 seconds. A pause between bytes greater then this will cause a time-out fault.

5) After receiving the last byte the validator will respond within 200 MSec with a "VS" if the program in the validator verifies with the transmitted data. If the program is not the same the validator responds with E:11.

11.4.8 Get program identification number(ID<cr>):

Function: Gets the program ID number. This number is stored along with the application program and is used to identify the program.

Format: "ID"

Returns: If a valid application program is found a six character hex string representing the program ID number is returned. If no valid application program is found "XXXXXX" is returned.

11.4.9 Exit serial download program command(EX<cr>):

Funtion: Exit the serial download program

Format: "EX"

Operation: If a valid application program resides in the program memory above 2K the validator jumps to the start of the program, terminating the Downloader. If no valid application program is found the validator returns to the command line input mode and sends a prompt ">" character.

11.5 *Error Messages*

Error messages are output in the form E:xx, where xx is an eight bit binary number in hex format. Bits D6-D0 contain the error number from 01 to 22. Bit D7 is set if the error is a "fatal" error. That is an error after which the validator can no longer be guaranteed to function correctly. Fatal errors can occur only during download. They occur when a partial download has been made, but due to an error it is unable to complete.

Error 01 - Command line to long. A command line greater then 36 characters long has been typed in.

Error 02 - Invalid command. The command string typed in does not correspond to any valid command.

Error 03 - Address out of bounds. The address of the data memory read or write operation is outside of the allowable range. The address is either below the partition address or above the limit of accessible memory.

Error 04 - Invalid parameter. The parameter input is not a valid hex value.

Error 05 - Time-out waiting for byte, during the Load and Verify operations. There is a 7 second time limit between the input of the command and the transmission of the first byte. There is also a 7 second time limit between transmission of individual bytes of the program file. If the validator returns with error 5 then this time has elapsed and the operation has been terminated. For Load operations this may result a fatal error if it occurs in the middle of a download.

Error 06 - Invalid record type. The format of the download file is incorrect. This is caused either by a data transmission error or a corrupted program file, or incorrect format (network inteated of non-network).

Error 07 - Program address out of range. The program start address is below 01000H. This

area is reserved for the serial Downloader. Address 7FF0h-7FFFh is also reserved.

Error 08 - File format error. The format of the download file is incorrect. This is caused either by a data transmission error or a corrupted program file.

Error 09 - Bad checksum. A file block has failed the checksum test. This is caused either by a data transmission error or a corrupted program file.

Error 10 - Partition out of bounds. The partition specified in the program file is less then 1000H or greater then 6FFFH.

Error 11 - Verify failed. The program file does not match what is stored in the validator program memory. This error can only occur as a result of a verify operation.

Error 12 - Country Authorization failure. The country code of the program file does not match any of the allowed countries stored in the validator protected memory area.

Error 13 - Password required. The requested operation can not be done until the password is first entered using the PW (password) command.

Error 14 - No password active. A password has been input using the PW command, but no password was set. Note, this is a warning only, the load operation is still enabled. To activate password protection use the SP command to set a password.

Error 15 - Incorrect password. The password entered does not match the password stored in then vaildator protected memory.

Error 16 - Password did not match. When using the SP command the user is required to input the password a second time to verify correctness. If the second password does not match the first then this error occurs.

Error 17 - not used
.

Error 18 - Key not found. A valid download key was not found.

Error 19 - No loads left in key. The number of authorized downloads has been exhausted.

Error 20 - Bad Program ID. The program ID in the program file does not match that found in the key.

Error 21 - Key update failure. The validator was unable to update the key after the load operation. This indicates a hardware failure in the key.

Error 22 - Invalid Upload request. The upload command can not transfer program data up because the program memory is invalid (no application program is present)

Error 23 - Invalid file header. The file header does not identify this program as a valid program. Either the file is corrupt or the program is for a different type of validator.

Error 24 - Invalid hardware configuration. The hardware configuration of the validator does not match the configuration specified in the program file header. You were attempting to load into the validator a program which is incompatible with the current validator hardware.

Revision History

Revision B ---------- July 3, 1997

1. Add new command description -- Set Security Level.
2. Add new command description -- Request Security Level.
3. Change the command codes of Set Stacker Mode commands.

   Set stacker mode normally open -- 48 Hex

   Set stacker mode normally closed -- 49 Hex
4. Change the command code of All Illegal command from 00H to 99H.
5. The message timing section has been rewritten.
6. The command description section has been rewritten.
7. The validator states section has been rewritten.
8. Added section 7.2 -- Power-Up credit recovery.

Revision A ---------- May 26, 1997

This release of CBV enhanced serial protocol features two new functions not available in the previous version.

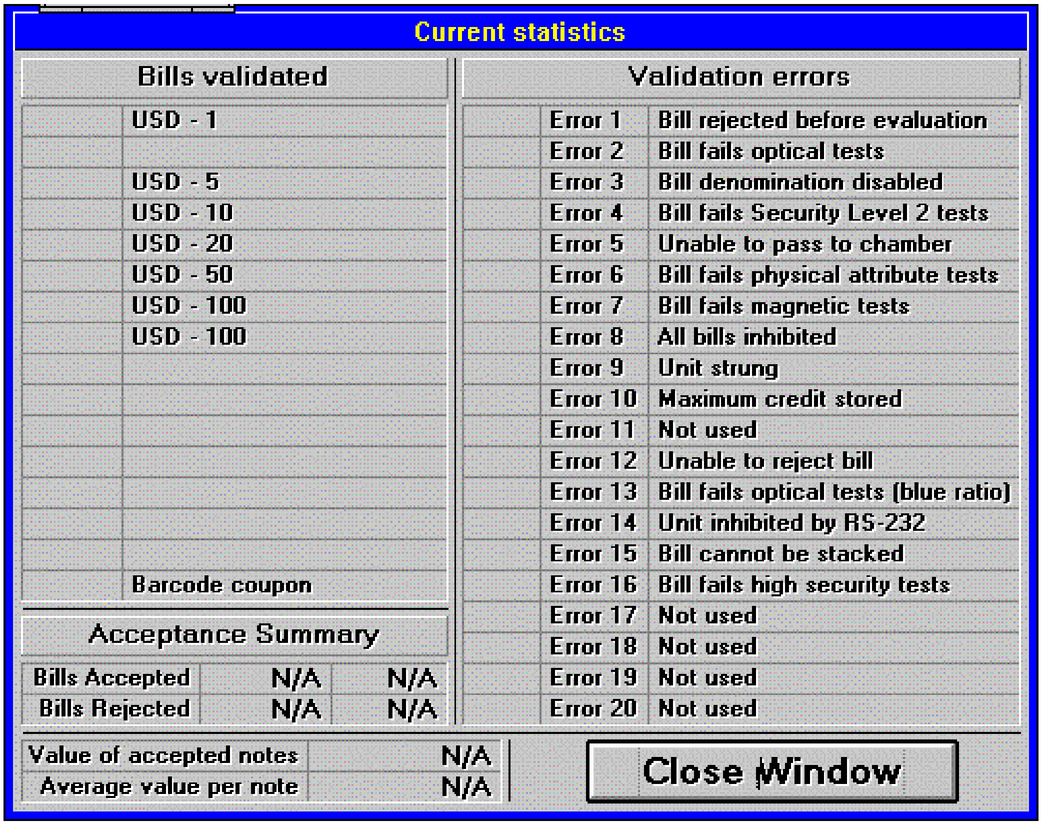
1. In the interrupt mode, interrupt message will be sent to the controller when bill is rejected before evaluation or bill fails the evaluation process. Also in the polling mode, the rejecting flag will be set if the bill is rejected before the evaluation.

2. The Stacker Full/Jam message is implemented. Both bits D3 and D1 in the status data byte 3 are set to report the Stacker Full/Jam state, since we do not have the ability to distinguish Stacker full from Stacker jam. Please refer to section 3.

Appendix A Country and Currency Code

COUNTRY	ISO	CUR_NAME
ARGENTINA	ARS	PESO
AUSTRALIA	AUD	DOLLAR
AUSTRIA	ATS	SCHILLING
BARBADOS	BBD	DOLLAR
BELGIUM	BEF	FRANC
BERMUDA	BMD	DOLLAR
BOLIVIA	BOB	BOLIVIANO
BOTSWANA	BWP	PULA
BRAZIL	BRL	REAIS
CANADA	CAD	DOLLAR
CAYMAN ISLANDS	KYD	DOLLAR
CBV	CBV	NA
CHILE	CLP	PESO
CHINA PEOPLES REP	CNY	YUAN
CNY-5-2	50	0
COLOMBIA	COP	PESO
COSTA RICA	CRC	COLON
CYPRUS	CYP	POUND
CZECH REP	CZK	KORUN
DENMARK	DKK	KRONA
DOMINICAN REP	DOP	PESO
EASTERN CARIBBEAN	XCD	DOLLAR
ECUADOR	ECS	SUCRE
EGYPT	EGP	POUND
ENGLAND	GBP	ST POUND
ESTONIA	EEK	KROON
FINLAND	FIM	MARKKA
FRANCE	FRF	FRANC
GERMANY	DEM	D MARK
GREECE	GRD	DRACHMA
GUATEMALA	GTQ	QUETZAL
HONG KONG	HKD	DOLLAR
HUNGARY	HUF	FORINT
ICELAND	ISK	KRONA
INDIA	INR	RUPEE
INDONESIA	IDR	RUPIAH
ISRAEL	ILS	SHEQEL
ITALY	ITL	LIRA
JAMAICA	JMD	DOLLAR
JAPAN	JPY	YEN
KOREA REPUBLIC	KRW	WON
KUWAIT	KWD	DINAR
LEBANON	LBP	POUND
MALAYSIA	MYR	RINGGIT
MALTA	MTL	LIRA
MEXICO	MXN	PESO
NAMIBIA	NMD	DOLLAR
NETHERLANDS	NLG	GULDEN
NETHERLANDS	ANG	GULDEN
NEW ZEALAND	NZD	DOLLAR
NORWAY	NOK	KRONE
OMAN	OMR	RIAL
PERU	PEN	NUEVO SOL
PHILIPPINES	PHP	PISO
POLAND	PLZ	ZLOTY
PORTUGAL	PTE	ESCUDO
QATAR	QAR	RIYAL
ROMANIA	ROL	LEU
RUSSIA	RUR	RUBLE
SAUDI ARABIA	SAR	RIYAL
SCOTLAND	SCP	POUND
SINGAPORE	SGD	DOLLAR
SLOVENIA	SIT	TOLAR
SOUTH AFRICA	ZAR	RAND
SPAIN	ESP	PESETA
SWAZILAND	SZL	LILANGENI
SWEDEN	SEK	KRONA
SWITZERLAND	CHF	FRANC
TAIWAN	TWD	DOLLAR
THAILAND	THB	BAHT
TRINIDAD/TOBAGO	TTD	DOLLAR
TURKEY	TRL	LIRA
UNITED ARAB	AED	DIRHAM
URUGUAY	UYU	NUEVO PESO
USA	USD	DOLLAR
VENEZUELA	VEB	BOLIVAR
ZIMBABWE	ZWD	DOLLAR

Appendix B Example of Statistics Data



Current statistics

	Bills validated
	USD - 1
	USD - 5
	USD - 10
	USD - 20
	USD - 50
	USD - 100
	USD - 100
	Barcode coupon

Acceptance Summary		
Bills Accepted	N/A	N/A
Bills Rejected	N/A	N/A

Value of accepted notes	N/A
Average value per note	N/A

	Validation errors	
	Error 1	Bill rejected before evaluation
	Error 2	Bill fails optical tests
	Error 3	Bill denomination disabled
	Error 4	Bill fails Security Level 2 tests
	Error 5	Unable to pass to chamber
	Error 6	Bill fails physical attribute tests
	Error 7	Bill fails magnetic tests
	Error 8	All bills inhibited
	Error 9	Unit strung
	Error 10	Maximum credit stored
	Error 11	Not used
	Error 12	Unable to reject bill
	Error 13	Bill fails optical tests (blue ratio)
	Error 14	Unit inhibited by RS-232
	Error 15	Bill cannot be stacked
	Error 16	Bill fails high security tests
	Error 17	Not used
	Error 18	Not used
	Error 19	Not used
	Error 20	Not used

Index