corporate headquarters
553 Basaltic Road Concord, Ontario Canada L4K 4W8
phone (905) 303-8874 fax (905) 303-8875
web www.cashcode.com

CashCode® NET

Interface

PT# MAN_CCNET Rev.2.3.1 NOV 2002

Contents

1 General Information

1.1 Introduction

This document defines a serial network interface. The interface is Master-Slave arrangement where all peripherals are Slave to a Master Controller.

1.2 Operational and Application Notes

The serial network interface, or serial bus interface, is configured for Master- Slave operation. There is one Master with the capability of communicating with some peripherals. The Master is defined as Controller and Slave as Peripheral.

Each peripheral is assigned a unique address and command set. The Controller will “poll” the Bus for Peripheral activity. That is, each Peripheral is asked for activity, and responds with either acknowledge, negative acknowledge, invalid command acknowledge, or specific data dependent on its current activity. If a Peripheral does not respond within a predefined time, (t-non-response as defined in the peripheral sections) it is assumed that it is not present on the Bus.

Bus interference or “crashes” are prevented because each Peripheral only responds upon being polled.

2 Communication Format.

2.1 Data format

Baud Rate:	9600 bps/19200 bps (no negotiation, hardware selectable)
Start bit:	1
Data bit:	8 (bit 0 = LSB, bit 0 sent first)
Parity:	Parity none
Stop bit:	1

2.2 Message Format

SYNC	ADR	LNG	CMD	DATA	CRC

SYNC:	1 byte	Message transmission start code [02H], fixed
ADR :	1 byte	Peripheral address
LNG :	1 byte*	Data length (Total number of bytes including SYNC and CRC)
CMD :	1 byte	Command
DATA	0 to 250 bytes	Data necessary for command (omitted if not required by CMD)
CRC:	2 bytes	Check code by CRC method, LSB first Object section to be from and including SYNC to end of DATA (Initial value = 0)
Error control method:		Error detection CRC method CRC - CCITT using whole byte shifting into a two-byte frame $P(X) = X^{16} + X^{12} + X^5 + 1$

* if a package cannot be fitted into 250-byte frame a wider frame may be used by setting **LNG** to 0; the actual packet length is inserted into **DATA** block bytes 0 and 1 if **CMD** (if present in the frame) **does not require subcommand**, otherwise in **DATA** block bytes 1 and 2; two-byte **LNG** always follows *MSB* first.

case 1 (CMD present, no subcommand):

SYNC	ADR	0	CMD	LNG HIGH	LNG LOW	DATA	CRC

case 2 (CMD present, subcommand present):

SYNC	ADR	0	CMD	SUBCMD	LNG HIGH	LNG LOW	DATA	CRC

case 3 (CMD not present, no subcommand):

SYNC	ADR	0	LNG HIGH	LNG LOW	DATA	CRC

This allows accommodation of data packages of up to 65528 bytes; please keep in mind that lengthy exchanges compromise bus bandwidth.

Example of CCNET CRC calculation using C-language source code:

```
#define POLYNOMIAL 0x08408

unsigned int GetCRC16(unsigned char* bufData, unsigned int sizeData)
{
  unsigned int TmpCRC, CRC, i;
  unsigned char j;
  CRC = 0;
  for(i=0; i < sizeData; i++)
   {
    TmpCRC = CRC ^ bufData[i];
    for(j=0; j < 8; j++)
     {
      if(TmpCRC & 0x0001) {TmpCRC >>= 1; TmpCRC ^= POLYNOMIAL;}
      else TmpCRC >>= 1;
     }
   }
  return CRC;
}
```

Example of CCNET CRC calculation using PASCAL-language source code:

```
const _CR_CCNET_CRC_POLY = $08408

function GetCRC16(InData: array of byte; DataLng: word): word;
var i,TmpCRC: word;
    j: byte;
begin
result:=0;
for i:=0 to (DataLng-1) do
 begin
  TmpCRC:=result xor InData[i];
  for j:=0 to 7 do
   begin
    if (TmpCRC and $0001)<>0 then
     begin
      TmpCRC:=TmpCRC shr 1;
      TmpCRC:=TmpCRC xor _CR_CCNET_CRC_POLY;
     end
                              else
     TmpCRC:=TmpCRC shr 1;
   end;
  result:=TmpCRC;
 end;
end;
```

2.3 Transmission and reception message formats

Transmission and reception message format is divided into the following four types.

(1) Command transmission **CONTROLLER to PERIPHERAL**

SYNC	ADR	LNG	CMD	DATA	CRC

SYNC : [02H]
ADR : Peripheral address
LNG : Data length
CMD : Command
DATA : Data necessary for command (omitted if not required by CMD)
CRC : Check code by CRC method

(2) ACK response **PERIPHERAL to CONTROLLER/ CONTROLLER to PERIPHERAL**

SYNC	ADR	LNG	DATA	CRC

SYNC : [02H]
ADR : Peripheral address
LNG : [06H]
DATA : [00H]
CRC : Check code by CRC method

(3) NAK response **PERIPHERAL to CONTROLLER/ CONTROLLER to PERIPHERAL**

SYNC	ADR	LNG	DATA	CRC

SYNC : [02H]
ADR : Peripheral address
LNG : [06H]
DATA : [FFH]
CRC : Check code by CRC method

Sent in PERIPHERAL to CONTROLLER direction if command from controller was not correctly received.

(4) Response message **PERIPHERAL to CONTROLLER**

SYNC	ADR	LNG	DATA	CRC

SYNC : [02H]
ADR : Peripheral address
LNG : Data length
DATA : Response's Data
CRC : Check code by CRC method

(5) ILLEGAL COMMAND Response message **PERIPHERAL to CONTROLLER**

SYNC	ADR	LNG	DATA	CRC

SYNC : [02H]
ADR : Peripheral address
LNG : [06]
DATA : [30H]
CRC : Check code by CRC method

Sent by the PERIPHERAL if command from CONTROLLER is not valid in reference to the current peripheral state.

2.4 Peripheral Addresses

The addresses below are defined.

Address	Definition
00H	*Forbidden*
01H	*Bill-to-Bill unit*
02H	*Coin Changer*
03H	*Bill Validator*
04H	*Card Reader*
05H	Reserved for Future Standard Peripherals
.	.
.	.
.	.
0DH	Reserved for Future Standard Peripherals
0EH	*Reserved for Future Broadcast Transmissions*
0FH	Reserved for Future Standard Peripherals

2.5 Software Operational Rules

- During multi-byte messages the most significant byte is sent first.
- If the Peripheral has not responded to a poll for its maximum non-response time, the Controller must continue to poll the Peripheral at least every ten seconds with a RESET command.
- All messages, from Controller or Peripheral, must be sent as quickly as possible. There is no minimum $t_{response}$ time. All data block transmissions must be started within 10 mS.
- Any data (bytes or bits) within a command or response that are not specifically defined must be left in a 0 state.
- The Controller may reset Peripheral by sending the signal BUS RESET for a minimum of 100 mS. This informs Peripheral to abort any activity and return to its power-on reset state. It is recommended that the Controller re-initialize each Peripheral after this type of reset. **WARNING**: BUS RESET is device and implementation dependant and may not be present within some devices.

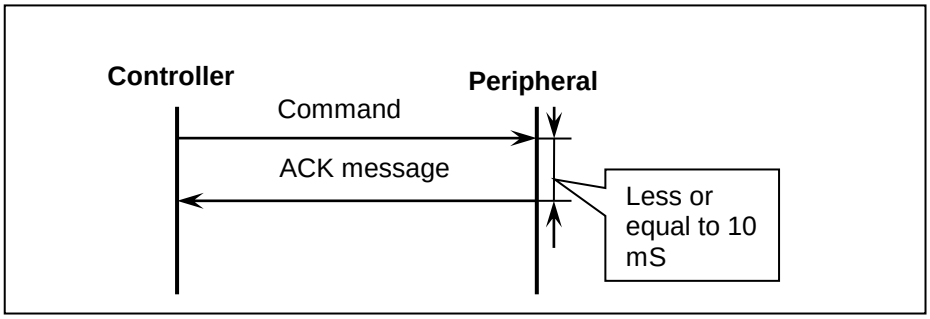
2.6 Typical Session Examples

The Controller must respond to data from a Peripheral with an Acknowledgment (ACK) or Negative Acknowledgment (NAK) message. The 10 mS time-out (t-response) described in the Timing section of this document is the equivalent of a NAK message.

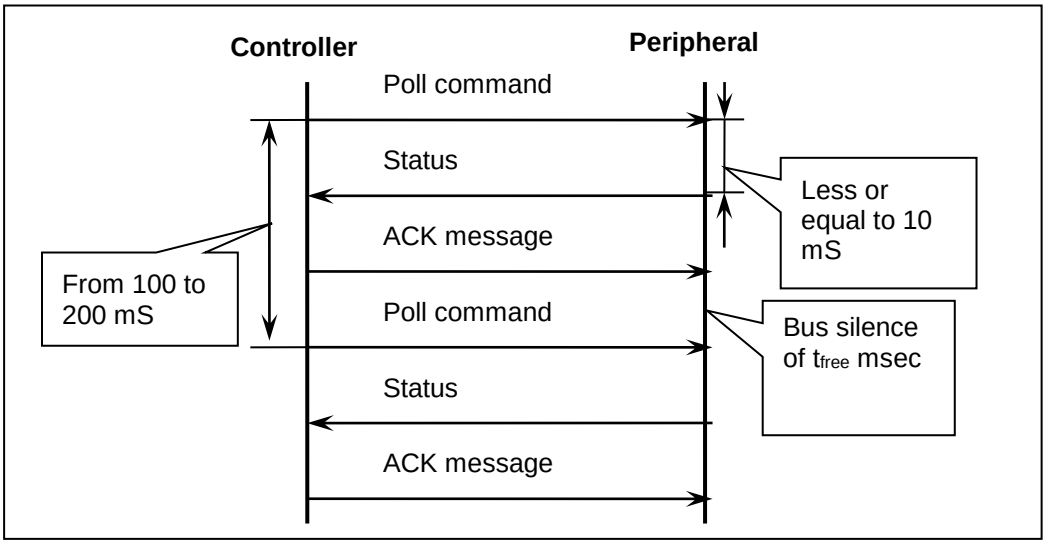
A Peripheral must respond to command from the Controller with response message, or ACK message, or NAK message. The 10 mS time-out (t-response) described in the Timing section of this document is the equivalent of a NAK message.

The t_{free} must be obeyed by the Controller between the end of any ACK or NAK confirmation response and start of the next command transmission. Currently t_{free} is defined as 10 mS of Bus silence, but for reliable operation of future multi-device buses the recommended value of t_{free} is 20mS.

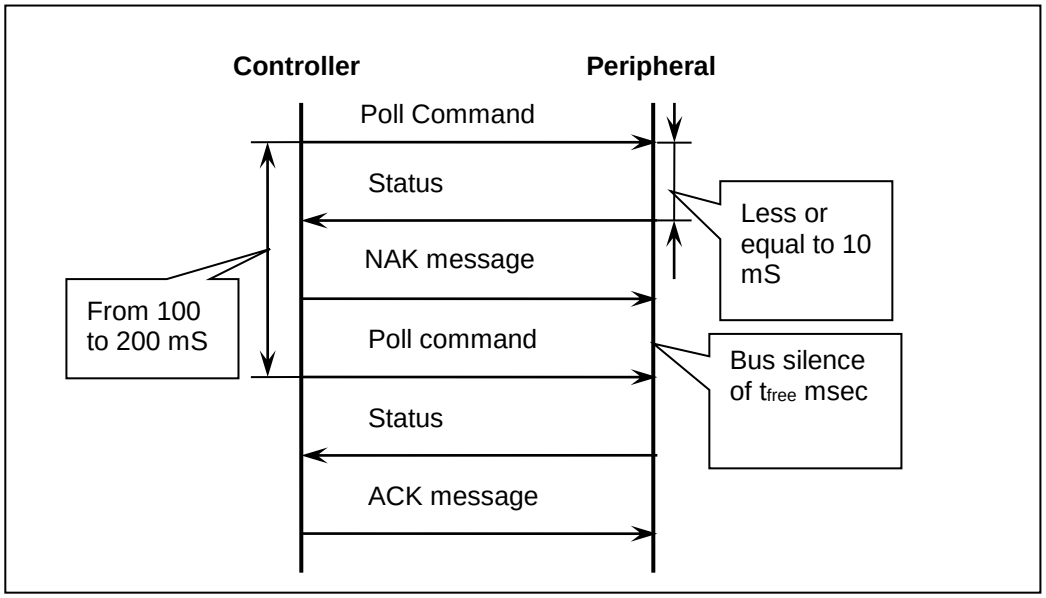
2.6.1 The diagram below represents a typical transmission when PERIPHERAL has no data to return.



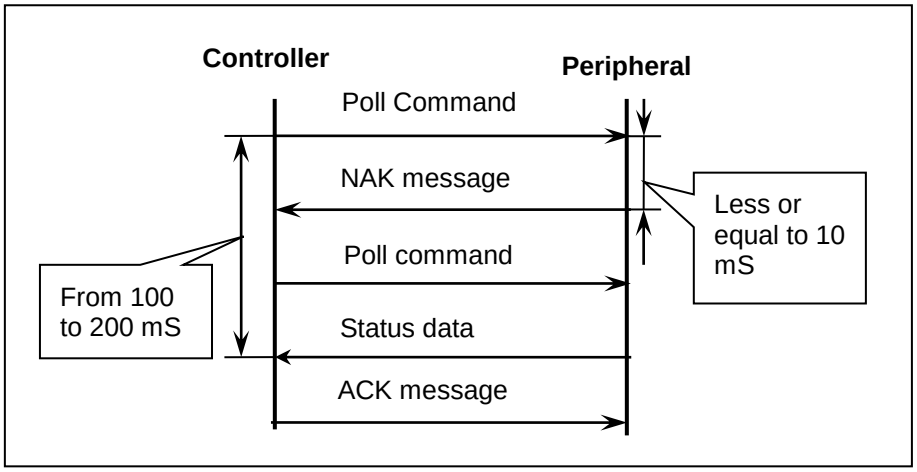
2.6.2 The diagram below represents a typical transmission when PERIPHERAL has data to return.



2.6.3 The diagram below represents a typical transmission when the Controller determines a CRC is not correct.



2.6.4 The diagram below represents a typical transmission when Peripheral determines a CRC is not correct. The Peripheral responds by sending a NAK message to the Controller to indicate that the information was not received correctly.



2.7 Timing Definitions

Baud rate	The rate of bit transfer per second
$t_{inter\text{-}byte(max.)}$	The maximum time allowed between bytes in a block transmission
$t_{response(max.)}$	The maximum time Peripheral will take to respond to a valid communication
$t_{bus\ reset(min.)}$	The minimum time of sending signal BUS RESET
$t_{non\text{-}response\ (max.)}$	The maximum non-response time
t_{poll}	The interval of time between two commands Poll
t_{free}	The interval of time between confirmation ACK or NAK and next command

2.8 Timing Specifications

Baud Rate	9600/19200 +1%/-2% NRZ (non-return to zero), non-negotiable, hardware selectable
$t_{inter\text{-}byte(max.)}$	5.0 ms
$t_{response(max.)}$	10.0 ms
$t_{bus\ reset(min.)}$	100 ms
$T_{non\text{-}response\ (max.)}$	5.0 S
t_{poll}	100-200 ms
t_{free}	10 ms (recommended 20 ms or more)

3 CONTROLLER/BILL-TO-BILL UNIT Communication Specification

3.1 Introduction

This section defines the communication bytes sent and received between Bill-to-Bill unit and the Controller. Unless stated otherwise, all information is assumed to be in a hexadecimal format. The Bill-to-Bill unit's address is 01H.

3.2 Command Protocol

The IDENTIFICATION, GET BILL TABLE and DOWNLOAD commands may be sent by the Controller when Bill-to-Bill unit is in the following states only: Power up, Initialise or Unit Disabled. If a command cannot be executed by the Bill-to-Bill unit in a given state COMMAND INVALID response is issued.

3.3 Controller Commands

Command	HEX Code	Description
RESET	*30H*	Command for Bill-to-Bill unit to self-reset
GET STATUS	*31H*	Request for Bill-to-Bill unit set-up status
SET SECURITY	*32H*	Sets Bill-to-Bill unit Security Mode. Command is followed by set-up data. See command format
POLL	*33H*	Request for Bill-to-Bill unit activity Status
ENABLE BILL TYPES	*34H*	Indicates Bill Type enable or disable. Command is followed by set-up data. See command format
STACK	*35H*	Sent by Controller to stack a bill in escrow into drop cassette or into one of the recycling cassettes
RETURN	*36H*	Sent by Controller to return a bill in escrow
IDENTIFICATION	*37H*	Request for Model, Serial Number, Software Version of Bill-to-Bill unit, Country ISO code, Asset Number
HOLD	*38H*	Command for holding the Bill-to-Bill unit in Escrow state
CASSETTE STATUS	*3BH*	Request for Bill-to-Bill unit cassette status
DISPENSE	*3CH*	Command to dispense a bill of specific type
UNLOAD	*3DH*	Command to unload bills from recycling cassette(s) to drop cassette
ESCROW CASSETTE STATUS	*3EH*	Request for recycling cassette(s) status
ESCROW CASSETTE UNLOAD	*3FH*	Command for routing bills to recycling cassette(s)
SET CASSETTE TYPE	*40H*	Assigns cassettes to bill types
GET BILL TABLE	*41H*	Request for bill type assignments
DOWNLOAD (see details for subcommands)	*50H*	Command for transition to download mode.

3.4 Controller Command Format

RESET

Controller Command	Code	Controller Data
RESET	30H	No data bytes

This command is used to tell the Bill-to-Bill unit that it must return to its default operating mode. It must abort all communication, reject any bills in the validation process, return any bills in the escrow position, and disable all other activity until otherwise instructed by the Controller.

GET STATUS

Controller Command	Code	Bill-to-Bill unit Response Data
GET STATUS	31H	9 bytes: **Z1 – Z9**

Z1-Z3	Bill Type, 3 bytes. Indicates the bill enables for bill types 0 to 23.
Z4-Z6	Bill Security Levels, 3 bytes. Indicates the security level for bill types 0 to 23.
Z7-Z9	Bill Type Routing, 3 bytes. Indicates what bill types can be routed to the Bill-to-Bill unit's cassettes. Valid bill types are 0 to 23.

Bill Type

Byte Z1 bits								Byte Z2 bits								Byte Z3 bits							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bill types enabled if bits set																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bill Security Levels

Byte Z4 bits								Byte Z5 bits								Byte Z6 bits							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bill types set to high security if bits set																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bill Type Routing

Byte Z7 bits								Byte Z8 bits								Byte Z9 bits							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bill types can be routed if bits set																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SET SECURITY

Controller Command	Code	Controller Data
SET SECURITY	32H	3 Bytes: **Y1 – Y3**

Byte Y1 bits								Byte Y2 bits								Byte Y3 bits							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bill types set to high security if bits set																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

A bit is set to indicate the type of bill(s), which are set to a “high” security level.

POLL

Controller Command	Code	Bill-to-Bill unit Response Data
POLL	33H	1 or 2 bytes: **Z1 or Z1- Z2**

Indicates status of the Bill-to-Bill unit and its activity. The Bill-to-Bill unit may send 1or 2 of the following data bytes:

Response data bytes		Description	
Z1	**Z2**		
10H		Power Up	The state of Bill-to-Bill unit after power up
11H		Power Bill	Power up with bill in validating head. After a RESET command from the Controller, the Bill-to-Bill unit returns the bill and continues initializing.
	Power up with bill in Bill-to-Bill unit. Always followed by location byte (see below).		
12H	**00H**	Power w/Bill in Transport Path	Power up with bill in transport section. After a RESET command from the Controller, the Bill-to-Bill unit stacks the bill to the drop cassette and continues initializing.
12H	**01H**	Power w/Bill in Dispenser	Power up with bills in dispenser. After a RESET command from the Controller, the Bill-to-Bill unit returns the bills to exit bezel and continues initializing.
13H		Initialize	The state, in which Bill-to-Bill unit executes initialization after RESET command from the Controller.
14H		Idling	In this state Bill-to-Bill unit waits for bill insertion.
15H		Accepting	In this state Bill-to-Bill unit executes scanning of a bill and determines its denomination.
17H		Stacking	In this state, the Bill-to-Bill unit transports a bill from Escrow position to the recycling cassette or to the drop box and remains in this state until the bill is stacked or returned if jammed.
18H		Returning	In this state Bill-to-Bill unit transports a bill from Escrow position to entry bezel and remains in this state until the bill is removed by customer.
19H		Unit Disabled	The Bill-to-Bill unit has been disabled by the Controller and also the state in which Bill-to-Bill unit is after

			initialization
1AH		Holding	The state, in which the bill is held in Escrow position after the HOLD command from the Controller.
1BH	**YH**	Device Busy	The state, in which Bill-to-Bill unit cannot answer a detailed command right now. On expiration of time **YH,** peripheral is accessible for polling. **YH** is expressed as multiple of 100 milliseconds.
1CH	Generic rejecting code. Always followed by rejection reason byte (see below).		
1CH	**60H**	Rejecting due to Insertion	Insertion error
1CH	**61H**	Rejecting due to Magnetic	Magnetic error
1CH	**62H**	Rejecting due to bill Remaining in the head	Bill remains in the head, and new bill is rejected.
1CH	**63H**	Rejecting due to Multiplying	Compensation error/multiplying factor error
1CH	**64H**	Rejecting due to Conveying	Conveying error
1CH	**65H**	Rejecting due to Identification1	Identification error
1CH	**66H**	Rejecting due to Verification	Verification error
1CH	**67H**	Rejecting due to Optic	Optic error
1CH	**68H**	Rejecting due to Inhibit	Return by “inhibited denomination” error
1CH	**69H**	Rejecting due to Capacity	Capacitance error
1CH	**6AH**	Rejecting due to Operation	Operation error
1CH	**6CH**	Rejecting due to Length	Length error
1DH		Dispensing	Bill-to-Bill unit enters this state after DISPENSE command. In this state Bill-to-Bill unit transports a bill from recycling cassette(s) to customer through dispenser and remains in this state until the bill(s) are removed by customer or jammed.
1EH		Unloading	Transporting bills to drop cassette.
1FH		Custom returning	Bill-to-Bill unit pass to this state after RECYCLING CASSETTE UNLOAD command if high bit of data byte is set. In this state Bill-to-Bill unit transports bill(s) from recycling cassette to customer through dispenser and remains in this state until the customer removes the bill(s).
20H		Recycling unloading	Bill-to-Bill unit passes to this state after RECYCLING CASSETTE UNLOAD command if high bit of data byte is dropped. In this state Bill-to-Bill unit transports bill from recycling cassette to recycle cassette of appropriate bill type or to drop cassette.
21H		Setting cassette type	Unloading of the recycling cassette is carried out and cassette is reassigned to new bill type
25H		Dispensed	Dispensing is completed.
26H		Unloaded	Unloading is completed
27H		Custom bills returned	Bills are returned to customer.

28H		Recycling cassette unloaded	Recycling Cassette unloading is completed.
29H		Set cassette type	Setting recycling cassette type is completed.
30H		Invalid command	Command from the Controller is not valid.
41H		Drop Cassette Full	Drop Cassette full condition
42H		Drop Cassette out of position	The Bill-to-Bill unit has detected the drop cassette to be open or removed.
43H		Bill Validator Jammed	Bill(s) are jammed in the acceptance path.
44H		Cassette Jammed	A bill are jammed in drop cassette.
45H		Cheated	The Bill-to-Bill unit sends this event if the intentions of the user to deceive the Bill-to-Bill unit are detected.
46H		Pause	The Bill-to-Bill unit reaches this state when the user tries to insert a bill before the previous bill is stacked. Bill-to-Bill unit stops motion of the bill until the entry channel is cleared.
47H	Generic Failure codes. Always followed by failure description byte (see below).		
47H	**50H**	Stack Motor Failure	Drop Cassette Motor failure
47H	**51H**	Transport Motor Speed Failure	Transport Motor Speed out of range
47H	**52H**	Transport Motor Failure	Transport Motor failure
47H	**53H**	Aligning Motor Failure	Aligning Motor failure
47H	**54H**	Initial Box Status Failure	Initial cassette Status failure
47H	**55H**	Optic Canal Failure	One of the optic sensors has failed to provide its response.
47H	**56H**	Magnetic Canal Failure	Inductive Sensor failure
47H	**57H**	Cassette 1 Motor Failure	Recycling Cassette 1 Motor Failure
47H	**58H**	Cassette 2 Motor Failure	Recycling Cassette 2 Motor Failure
47H	**59H**	Cassette 3 Motor Failure	Recycling Cassette 3 Motor Failure
47H	**5AH**	Bill-to-Bill unit Transport Motor Failure	One of the Bill-to-Bill unit Transport Motors failure
47H	**5BH**	Switch Motor 1 Failure	Switch Motor 1 Failure
47H	**5CH**	Switch Motor 2 Failure	Switch Motor 2 Failure
47H	**5DH**	Dispenser Motor 1 Failure	Dispenser Motor 1 Failure
47H	**5EH**	Dispenser Motor 2 Failure	Dispenser Motor 2 Failure
47H	**5FH**	Capacitance Canal Failure	Capacitance sensor failed to respond
	Bill-to-Bill unit Jammed. Always followed by location byte (see below).		
48H	**70H**	Bill Jammed in Cassette 1	A bill is jammed in Recycling Cassette 1
48H	**71H**	Bill Jammed in Cassette 2	A bill is jammed in Recycling Cassette 2
48H	**72H**	Bill Jammed in Cassette 3	A bill is jammed in Recycling Cassette 3
48H	**73H**	Bill Jammed in Transport Path	A bill is jammed in Transport Path
48H	**74H**	Bill Jammed in Switch	A bill is jammed in Switch
48H	**75H**	Bill Jammed in Dispenser	A bill is jammed in Dispenser

	Events with credit.		
80H	**YH**	Escrow position	**Y** = bill type (0 to 23)
81H	**YH**	Bill stacked	**Y** = bill type (0 to 23)
82H	**YH**	Bill returned	**Y** = bill type (0 to 23)

ENABLE BILL TYPES

Controller Command	Code	Controller Data
ENABLE BILL TYPES	34H	6 bytes: **Y1 – Y6**

Byte Y1 bits								**Byte Y2 bits**								**Byte Y3 bits**							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bill types enabled if bits set																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

NOTE: Sending 000000H disables the Bill-to-Bill unit.

Byte Y4 bits								**Byte Y5 bits**								**Byte Y6 bits**							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bill types with escrow enabled if bits set																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

NOTE: On power-up or reset all bill acceptance and escrow are disabled.

STACK

Controller Command	Code	Controller Data
STACK	35H	No data bytes

This command forces a bill in escrow position to be sent to drop cassette or one of the recycling cassettes.

NOTE*:* After a STACK command the Bill-to-Bill unit should respond to a POLL command with the BILL STACKED message within 30 seconds. If this command is sent when the Bill-to-Bill unit is not in ESCROW state the ILLEGAL COMMAND message is returned.

RETURN

Controller Command	Code	Controller Data
RETURN	36H	No data bytes

This command causes the Bill-to-Bill unit to return a bill in escrow position to the customer.

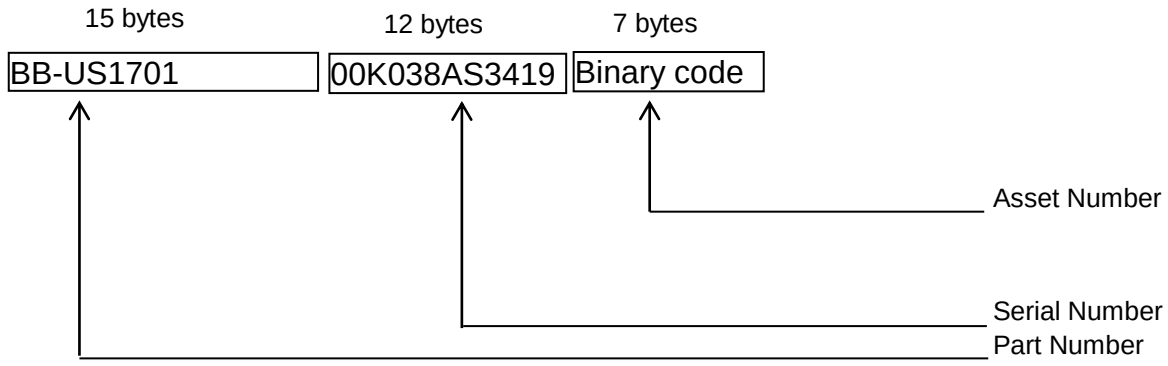
NOTE*:* After a RETURN command the Bill-to-Bill unit should respond to a POLL command with the BILL RETURNED message within 30 seconds. If this command is sent when the Bill-to-Bill unit is not in ESCROW state the ILLEGAL COMMAND message is returned.

IDENTIFICATION

Controller Command	Code	Bill-to-Bill unit Response Data
IDENTIFICATION	37H	34 bytes: **Z1 – Z34**

Bytes	Description
Z1-Z15	Part Number – 15 bytes, ASCII characters
Z16-Z27	Serial Number – 12 bytes Factory assigned serial number, ASCII characters
Z28-Z34	Asset Number – 7 bytes, unique to every Bill Validator, binary data

Bytes Z1-Z27 must be sent as ASCII Characters. Zero (30H) and Blank (20H) are acceptable. Asset Number must be sent as binary code.



HOLD

Controller Command	Code	Controller Data
HOLD	38H	No data bytes

This command allows the Controller to hold Bill-to-Bill unit in a state Escrow during 10 s. After this time the Controller should send the STACK or RETURN command. For continued holding in an Escrow state it is necessary to resend this command. Otherwise Bill-to-Bill unit will execute return of a bill.

CASSETTE STATUS

Controller Command	Code	Bill-to-Bill unit Response Data
CASSETTE STATUS	3BH	32 bytes**: Z1-Z32**

Z1-Z32 Cassette Status – 32 bytes.

Indicates the greatest number of bills that the Bill-to-Bill unit "knows" definitely is present in the bill cassettes. The 32-byte string consists from 16 two-byte words. Each two-byte position in the 32-byte string indicates the number of cassette. For example, the first two bytes sent indicate the number of cassette 1. First byte of word has the following format:
(**abc** zzzzz)
If **a** = 0 Cassette not present.
If **b** = 1 Cassette hasn't any type (type NULL).
If **c** = 1 Cassette has type ESCROW.
zzzzz - Type of bill. This bits is valid if bits b,c = 0 and a = 1 only

Second byte of word indicates number of bills in cassette.
Unsent bytes are assumed to be zero.

DISPENSE

Controller Command	Code	Controller Data
DISPENSE	3CH	27 bytes: **Y1-Y27**

Y1-Y3 Bill Type to be dispensed - 3 bytes. A bit is set to indicate bill type to be dispensed.

Byte Y1 bits								**Byte Y2 bits**								**Byte Y3 bits**							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bill Types to be dispensed if bits set																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Y4-Y27 These bytes contain number of bills to dispense for those types, which are specified in bytes Y1-Y3. Zero bytes are not sent. For example, if in bytes Y1-Y3 two bits are set, Y4 and Y5 will be sent only.

NOTE: The total number of bills to dispense must not exceed 20 bills.

UNLOAD

Controller Command	Code	Controller Data
UNLOAD	3DH	2 bytes: **Y1-Y2**

This command unloads the bills from the recycling cassettes to drop cassette.

Y1 Cassette number code

Y2 Iindicates the number of bills to be unloaded.

EXTRA CAPABILITIES – RECYCLING CASSETTE COMMANDS

RECYCLING CASSETTE STATUS

Controller Command	Code	Bill-to-Bill unit Response Data
RECYCLING CASSETTE	3EH	1 byte: **Z1**

Z1 (**F**yyy yyyy)

F = Recycling Cassette Full Status.

yyy yyyy = number of bills in the recycling cassette.

NOTE: If a Bill-to-Bill unit can detect a cassette jam, defective cassette sensor, or other malfunction, it will indicate the cassette is “bad” by sending a recycling cassette full status and a count of zero.

RECYCLING CASSETTE UNLOAD

Controller Command	Code	Controller Data
RECYCLING CASSETTE	3FH	1 byte: **Y1**

Y1 (**D**yyy yyyy)

If Y1 = 1yyy yyyy; yyyy yyy = Number of bills to return to customer.

If Y1 = 0; Unload one bill to recycling cassette of appropriate bill type or drop cassette

Command for routing bills from recycling cassette. This command allows returning customer bills in case of cancelling.

NOTE: The total number of bills must not exceed 20 bills.

SET RECYCLING CASSETTE TYPE

Controller Command	Code	Controller Data
SET RECYCLING CASSETTE TYPE	40H	2 bytes: **Y1-Y2**

Command for assigning recycling cassette bill type. If the cassette is not empty, the command forces complete unloading of the cassette.

Y1 Number cassette code
Y2 (**abc** zzzzz)
If **a** = 1 Cassette is present.
If **b** = 1 Lets Cassette type NULL.
If **c** = 1 Lets Cassette type ESCROW.
Only one of this bit can be set. zzzzz - Type of bill. This bits is valid if bits b,c = 0 and a = 1 only

Notes: It is strongly recommended to send RECYCLING CASSETTE STATUS command before sending SET RECYCLING CASSETTE STATUS TYPE command to assure proper cassette type assignment.

GET BILL TABLE

Controller Command	Code	Bill-to-Bill unit Response Data
GET BILL TABLE	41H	120 bytes: **Z1-Z120**

Command for request bill type description.

Z1-Z120 The 120 - byte string consists of 24 five-byte words.
Byte 1 of word – most significant digit of the denomination.
Bytes 2-4 of word – country code in ASCII characters.
Byte 5 of word –decimal placement or proceeding zeros. If bit D7 is 0, the bits D0-D6 indicate the number of proceeding zeros. If bit D7 is 1, the bits D0-D6 indicates the decimal point position starting from the right and moving to the left.
A five-byte position in the 120-bytes string indicates bill type description for the particular bill type. For example, first five bytes correspond to bill type=0, second five bytes correspond to bill type=1 and so on.

Example:

Bill Type	Denomination Code First Byte	Country Code 3 bytes	Denomination Code Second Byte	Denomination	
0	0x1	ITL	0x3	1,000	Lira
1	0x2	ITL	0x3	2,000	Lira
2	0x5	ITL	0x3	5,000	Lira
3	0x5	NLC	0x82	.05	Crown
4	0x19	NLC	0x82	.25	Crown
5	0x19	NLC	0x81	2.5	Crown
6	0x19	NLC	0x0	25	Crown
7	0x19	NLC	0x2	2500	Crown

8	0x1	USA	0x0	1	Dollar
9	0x5	USA	0x0	5	Dollar
10	0x1	USA	0x1	10	Dollar
11	0x2	USA	0x1	20	Dollar

Unsent bytes are assumed to be zero.

DOWNLOAD

Controller Command	Code	Controller Data
DOWNLOAD	50H	1 byte: **Y1**

Y1 – subcommand byte.

This command is not implemented in current Bill-to-Bill unit software releases and will always return an INV response.

4 CONTROLLER/COIN CHANGER Communication Specification

4.1 Introduction.

This section defines the communication bytes sent and received by a Coin Changer. The Coin Changer's address is 02H. Unless stated otherwise, all information is assumed to be in hexadecimal format.

The coin changer must be compatible with Controller that is designed according to the operational rules defined earlier in this document.

4.2 Command Protocol

The commands IDENTIFICATION, GET COIN TABLE, and DOWNLOAD should be sent by the Controller, when Coin Changer is in the following states: Power up, Initialize or Unit Disabled.

If the Coin Changer, which is not executable in its present state, receives a Controller command Coin Changer issues ILLEGAL COMMAND message.

4.3 Controller Commands

Command	HEX Code	Description
RESET	*08H*	Command for Coin Changer to self-reset
GET STATUS	*09H*	Request for Coin Changer set-up status
TUBE STATUS	*0AH*	Request for Coin Changer tube status.
POLL	*0BH*	Request for Coin Changer activity Status
ENABLE COIN TYPES	*0CH*	Indicates Coin Type enable or disable. Command is followed by set-up data. See command format
DISPENSE	*0DH*	Command to dispense a coin type. Command is followed by set-up data. See command format
IDENTIFICATION	*0FH*	Request for Model, Serial Number, Software Version of Coin Changer, Country ISO code
GET COIN TABLE	*10H*	Request for coin type description
DOWNLOAD	*50H*	Command for transition to download mode

4.4 Controller Command Format

RESET

Controller Command	Code	Controller Data
RESET	08H	No data bytes

This command is used to tell the Coin Changer that it must return to its default-operating mode. It must abort all communication and disable all acceptance until otherwise instructed by the Controller.

GET STATUS

Controller Command	Code	Coin Changer Response Data
GET STATUS	0CH	6 bytes: **Z1 – Z6**

Z1-Z3	Coin Type, 3 bytes. Indicates the coin enables for coin types 0 to 23.
Z4-Z6	Coin Type Routing, 3 bytes. Indicates what coin types can be routed to the Coin Changer's tubes. Valid coin types are 0 to 23.

Coin Type

Byte Z1 bits								Byte Z2 bits								Byte Z3 bits							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Coin types enabled if bits set																							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Coin Type Routing

Byte Z4 bits								Byte Z5 bits								Byte Z6 bits							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Coin types can be routed if bits set																							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

NOTE: Coin type credits sent as FFH are assumed to be Free Vend tokens and their value is assumed to be worth one vend.

TUBE STATUS

Controller Command	Code	Coin Changer Response Data
TUBE STATUS	0AH	18 byte**: Z1-Z27**

Z1-Z3 Tube Full Status - 3 bytes.

Indicates status of coin tube for coin types 0 to 23.
A bit is set to indicate a full tube.

Byte Z1 bits								Byte Z2 bits								Byte Z3							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Tube for Coin types is full if bits set																							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Z3-Z27 Tube Status – 24 bytes.

Indicates the greatest number of coins that the changer "knows" reliably can be paid out. A bytes position in the 24-byte string indicates the number of coins in a tube for a particular coin type. For example, the first byte sent indicates the number of coins in a tube for coin type 0. Bytes not sent are assumed to be zero.

NOTE: If a Coin Changer can detect a tube jam, defective tube sensor, or other malfunction, it will indicate the tube is “bad” by sending a tube full status and a count of zero for the malfunctioning coin type.

POLL

Controller Command	Code	Bill-to-Bill unit Response Data
POLL	0BH	1 or 2 bytes: **Z1 or Z1- Z2**

Indicates state of the Coin Changer and its activity. The Coin Changer may send 1or 2 of the following data bytes:

Response data bytes		Description	
Z1	**Z2**		
10H		Power Up	The state of Coin Changer after power up
11H		Initialize	The state, in which Coin Changer executes initialization on the RESET command of the Controller.
12H		Idling	In this state Coin Changer waits for an inserting of coin into its front bezel.
13H		Accepting	In this state Coin Changer validates a coin and determines its denomination.
14H		Unit Disable	The Coin Changer has been disabled by the Controller and also the state in which Coin Changer is after initialization
14H		Changer Busy	The Coin Changer is busy and cannot answer a detailed command right now.
16H		Changer Pay Out Busy	The Coin Changer is busy activating Pay Out devices.
17H		Escrow request	An escrow lever activation has been detected.
18H		Double Arrival	Two coins were detected too close together to validate either one.
19H		No Credit	A coin was validated but did not get to the place in the system when credit is given.
13H		Coin Routing Error	A coin has been validated, but did not follow the intended routing.
13H		Generic Acceptor Error	The Coin Changer has detected that the validator has been removed or not responding.
13H		Defective Tube Sensor	The Coin Changer has detected one of the tube sensors behaving abnormally.
13H		Coin Jam	A coin(s) has jammed in the acceptance path.
13H		Tube Jam	A tube Pay Out attempt has resulted in jammed condition.
40 - 4FH (0100xxxx)	**Z**	Coin accepted to cash box	xxxx = coin type deposited (0 to 15). **Z** = number of coins in the tube for the coin type accepted.
50 - 5FH (0101xxxx)	**Z**	Coin accepted to tube	xxxx = coin type deposited (0 to 15). **Z** = number of coins in the tube for the coin type accepted.
60 - 6FH		Not used	
70 - 7FH (0111xxxx)	**Z**	Coin was rejected	xxxx = coin type deposited (0 to 15). **Z** = number of coins in the tube for the coin type accepted.

80 – FFH (1yyyxxxx)	**Z**	Coin dispensed manually	yyy = number of coins dispensed. xxxx = coin type dispensed (0 to 15) **Z** = number of coins in the tube for the coin type accepted.

ENABLE COIN TYPES

Controller Command	Code	Controller Data
ENABLE COIN TYPES	0CH	4 bytes: **Y1 – Y6**

Y1 – Y3 Coin enable – 3 bytes

Byte Y1 bits								Byte Y2 bits								Byte Y3 bits							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Coin types enabled if bits set																							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

A bit is set to indicate a coin type is accepted. For example, bit 6 is set to indicate coin type 6, bit 15 is set to indicate coin type 15, and so on. To disable the changer, disable all coin types by sending a data block containing 000000H. All coins are automatically disabled upon reset.

Y6 – Y6 Manual Dispense enable – 3 bytes

Byte Y4 bits								Byte Y5 bits								Byte Y6 bits							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Dispense enabled if bits set																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

A bit is set to indicate dispense enable. For example, bit 2 is set to enable dispensing of coin type 2. This command enables/disables manual dispensing using optional inventory switches. All manual dispensing must be disabled while in the sales mode.

DISPENSE

Controller Command	Code	Controller Data
DISPENSE	0DH	2 bytes: **Y1-Y2**

Y1 = Indicates coin types 0 to 23 to be dispensed.

Y2 = Indicates number of coins to be dispensed.

If two coin types have the same value, the highest coin type must be paid out first.

IDENTIFICATION

Controller Command	Code	Acceptor Response Data
IDENTIFICATION	0FH	19 bytes: **Z1 – Z19**

Bytes	Description
Z1-Z5	Model No. & Model Code – 5 bytes
Z6-Z12	Serial Number – 7 bytes Factory assigned serial number
Z13-Z16	Version – 4 bytes Current software version
Z17-Z19	Country – 3 bytes ISO code

Bytes Z1-Z19 must be sent as ASCII Characters zero (30H) and blanks (20H) are acceptable.

GET COIN TABLE

Controller Command	Code	Acceptor Response Data
GET COIN TABLE	10H	80 bytes: **Z1-Z80**

Command for request coin type description.

Z1-Z80 The 80 - byte string consists from 24 five-byte words.
Byte 1 of word – hex representation most significant digit of the denomination.
Bytes 2-4 of word – country code in ASCII characters.
Byte 5 of word – this byte used to determine decimal placement or proceeding zeros. If bit D7 is 0, the bits D0-D6 indicate the number of proceeding zeros. If bit D7 is 1, the bits D0-D6 indicates the decimal point position starting from the right and moving to the left.
A five-byte position in the 80-bytes string indicates coin type description for the particular coin type. For example, first five byte correspond bill type=0, second five byte correspond bill type=1 and so on.
Example of coin type description for 25-cent USA: 0x19“USD”, 0x82;
Unsent bytes are assumed to be zero.

DOWNLOAD

Controller Command	Code	Controller Data
DOWNLOAD	50H	NONE

This command is not implemented in current Coin Changer software releases and will always return an INV response.

5 CONTROLLER/BILL VALIDATOR Communication Specification

5.1 Introduction.

This section defines the communication bytes sent and received between the Bill Validator and the Controller. Unless stated otherwise, all information is assumed to be in a hexadecimal format. The Bill Validator's address is 03H.

5.2 Command Protocol

If a Controller command is received by the Bill Validator, which is not executable in its present state, the Bill Validator issues ILLEGAL COMMAND message.

5.3 Controller Commands

Command	HEX Code	Description
RESET	*30H*	Command for Bill Validator to self-reset
GET STATUS	*31H*	Request for Bill Validator set-up status
SET SECURITY	*32H*	Sets Bill Validator Security Mode. Command is followed by set-up data. See command format
POLL	*33H*	Request for Bill Validator activity Status
ENABLE BILL TYPES	*34H*	Indicates Bill Type enable or disable. Command is followed by set-up data. See command format
STACK	*35H*	Sent by Controller to send a bill in escrow to the drop cassette
RETURN	*36H*	Sent by Controller to return a bill in escrow
IDENTIFICATION	*37H*	Request for Software Part Number, Serial Number, Asset Number
HOLD	*38H*	Command for holding of Bill Validator in Escrow state
SET BARCODE PARAMETERS	*39H*	Command for settings the barcode format and number of characters
EXTRACT BARCODE DATA	*3AH*	Command for retrieving barcode data if barcode coupon is found. If this command is sent when barcode coupon is not found the Bill Validator returns ILLEGAL COMMAND response.
GET BILL TABLE	*41H*	Request for bill type description
GET CRC32 OF THE CODE	*51H*	Request for Bill Validator's firmware CRC32.
DOWNLOAD (see details for subcommands)	*50H*	Command for transition to download mode.
REQUEST STATISTICS	*60H*	Command for retrieving full information about acceptance performance.

The IDENTIFICATION, GET BILL TABLE, DOWNLOAD and REQUEST STATISTICS commands should be sent by the Controller when Bill Validator is in the following states: Power up, Initialize, one of the Failure states (41H-47H) or Unit Disabled. Otherwise an ILLEGAL COMMAND response will be returned.

5.4 Controller Command Format

RESET

Controller Command	Code	Controller Data
RESET	30H	No data bytes

This command is used to tell the Bill Validator that it must return to its default operating mode. It must abort all communication, reject any bills in the validation process, return any bills in the escrow position, and disable all other activity until otherwise instructed by the Controller.

GET STATUS

Controller Command	Code	Validator Response Data
GET STATUS	31H	6 bytes: **Z1 – Z6**

Z1-Z3	Bill Type, 3 bytes. Indicates the bill enables for bill types 0 to 23.
Z4-Z6	Bill Security Levels, 3 bytes. Indicates the security level for bill types 0 to 23.

Bill Type

Byte Z1 bits								Byte Z2 bits								Byte Z3 bits							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bill types enabled if bits set																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bill Security Levels

Byte Z4 bits								Byte Z5 bits								Byte Z6 bits							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bill types set to high security if bits set																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SET SECURITY

Controller Command	Code	Controller Data
SET SECURITY	32H	3 Bytes: **Y1 – Y3**

Byte Y1 bits								Byte Y2 bits								Byte Y3 bits							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bill types set to high security if bits set																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

A bit is set to indicate the type of bill(s), which are set to a “high” security level.

POLL

Controller Command	Code	Bill Validator Response Data
POLL	33H	1 or 2 bytes: **Z1 or Z1- Z2**

Indicates state of the Bill Validator and its activity. The Bill Validator may send 1or 2 of the following data bytes:

Response data bytes		Description	
Z1	**Z2**		
10H		Power Up	The state of the Bill Validator after power up
11H		Power Up with Bill in Validator	Power up with bill in the Bill Validator. After a RESET command from the Controller Bill Validator returns the bill and continues initializing.
12H		Power Up with Bill in Stacker	Power up with bill in Stacker (Bill was transported too far to be returned). After the Bill Validator is reset and INITIALIZING is complete, status will immediately change to STACKED(81H) (Credit Recovery feature).
13H		Initialize	Bill Validator executes initialization after the RESET command from Controller.
14H		Idling	Bill Validator waits for an inserting of bill into its bill path.
15H		Accepting	Bill Validator executes scanning of a bill and determines its denomination.
17H		Stacking	Bill Validator transports a bill from Escrow position to drop cassette and remains in this state until the bill is stacked or jammed.
18H		Returning	Bill Validator transports a bill from Escrow position back to customer and remains in this state until customer removes the bill or the bill is jammed.
19H		Unit Disabled	Bill Validator has been disabled by the Controller or just came out of initialization
1AH		Holding	The state, in which the bill is held in Escrow position after the HOLD command of the Controller.
1BH	**YH**	Device Busy	Bill Validator cannot answer with a full-length message right now. On expiration of time **YH,** peripheral is accessible to polling. **YH** is expressed in multiple of 100 milliseconds.
1CH	**Generic rejecting code. Always followed by rejection reason byte (see below).**		
1CH	**60H**	Rejecting due to Insertion	Insertion error
1CH	**61H**	Rejecting due to Magnetic	Dielectric error
1CH	**62H**	Rejecting due to Remained bill in head	Previously inserted bill remains in head
1CH	**63H**	Rejecting due to Multiplying	Compensation error/multiplying factor error
1CH	**64H**	Rejecting due to	Bill transport error

		Conveying	
1CH	**65H**	Rejecting due to Identification1	Identification error
1CH	**66H**	Rejecting due to Verification	Verification error
1CH	**67H**	Rejecting due to Optic	Optic Sensor error
1CH	**68H**	Rejecting due to Inhibit	Return by “inhibit denomination” error
1CH	**69H**	Rejecting due to Capacity	Capacitance error
1CH	**6AH**	Rejecting due to Operation	Operation error
1CH	**6CH**	Rejecting due to Length	Length error
41H		Drop Cassette Full	Drop Cassette full condition
42H		Drop Cassette out of position	The Bill Validator has detected the drop cassette to be open or removed.
43H		Validator Jammed	A bill(s) has jammed in the acceptance path.
44H		Drop Cassette Jammed	A bill has jammed in drop cassette.
45H		Cheated	Bill Validator sends this event if the intentions of the user to deceive the Bill Validator are detected.
46H		Pause	When the user tries to insert a second bill when the previous bill is in the Bill Validator but has not been stacked. Thus Bill Validator stops motion of the second bill until the previous bill is stacked.
47H	**Generic Failure codes. Always followed by failure description byte (see below).**		
47H	**50H**	Stack Motor Failure	Drop Cassette Motor failure
47H	**51H**	Transport Motor Speed Failure	Transport Motor Speed failure
47H	**52H**	Transport Motor Failure	Transport Motor failure
47H	**53H**	Aligning Motor Failure	Aligning Motor failure
47H	**54H**	Initial Cassette Status Failure	Initial Cassette Status failure
47H	**55H**	Optic Canal Failure	One of the optic sensors has failed to provide its response.
47H	**56H**	Magnetic Canal Failure	Inductive sensor failed to respond
47H	**5FH**	Capacitance Canal Failure	Capacitance sensor failed to respond
	Events with credit.		
80H	**YH**	Escrow position	**Y** = bill type (0 to 23). If bill type is enabled with escrow the Bill Validator waits command from Controller to stack or to return bill. If during 10 sec command will not be sent bill will be returned.
81H	**YH**	Bill stacked	**Y** = bill type (0 to 23)
82H	**YH**	Bill returned	**Y** = bill type (0 to 23)

YH =17H(23_{10}) corresponds to a barcode coupon.

ENABLE BILL TYPES

Controller Command	Code	Controller Data
ENABLE BILL TYPES	34H	6 bytes: **Y1 – Y6**

Byte Y1 bits								Byte Y2 bits								Byte Y3 bits							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bill types enabled if bits set																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

NOTE: Sending 000000H disables the Bill Validator.

Byte Y4 bits								Byte Y5 bits								Byte Y6 bits							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bill types with escrow enabled if bits set																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

NOTE: On power-up or reset all bill acceptance and escrow are disabled.

STACK

Controller Command	Code	Controller Data
STACK	35H	No data bytes

This command causes the Bill Validator to send the bill in escrow position to the drop cassette.

NOTE*:* After a STACK command the Bill Validator should respond to a POLL command with the BILL STACKED message within 30 seconds. If this command is sent when the Bill Validator is not in ESCROW state the ILLEGAL COMMAND message is returned.

RETURN

Controller Command	Code	Controller Data
RETURN	36H	No data bytes

This command causes the Bill Validator to return bill in escrow position to the customer.

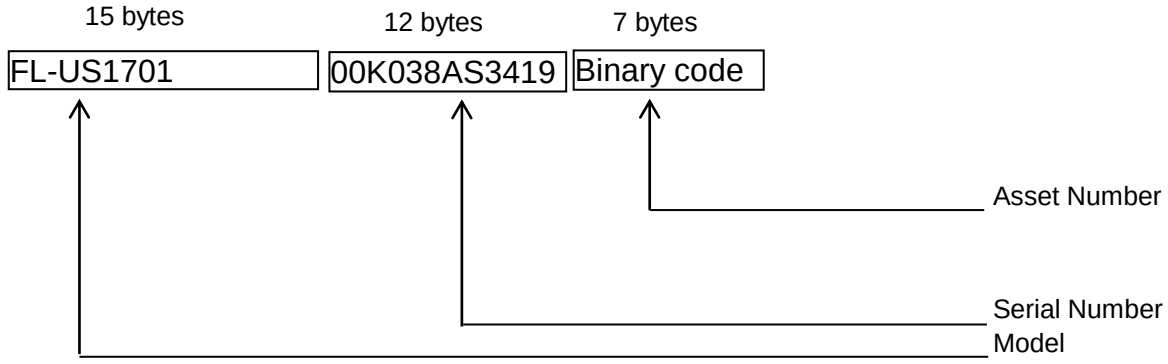
NOTE*:* After a RETURN command the Bill Validator should respond to a POLL command with the BILL RETURNED message within 30 seconds. If this command is sent when the Bill Validator is not in ESCROW state the ILLEGAL COMMAND message is returned.

IDENTIFICATION

Controller Command	Code	Bill-to-Bill unit Response Data
IDENTIFICATION	37H	34 bytes: **Z1 – Z34**

Bytes	Description
Z1-Z15	Part Number – 15 bytes, ASCII characters
Z16-Z27	Serial Number – 12 bytes Factory assigned serial number, ASCII characters
Z28-Z34	Asset Number – 7 bytes, unique to every Bill Validator, binary data

Bytes Z1-Z27 must be sent as ASCII Characters. Zero (30H) and Blank (20H) are acceptable. Asset Number must be sent as binary code.



This command is valid in the following states only: Power up, Initialize, one of the Failure states (41H-47H) or Unit Disabled.

HOLD

Controller Command	Code	Controller Data
HOLD	38H	No data bytes

This command allows the controller to hold Bill Validator in Escrow during 10 s. After this time the Controller should send the STACK or RETURN command. For continued holding in an Escrow state it is necessary to resend this command. Otherwise the Bill Validator will execute return of a bill.

SET BARCODE PARAMETERS

Controller Command	Code	Controller Data
SET BARCODE PARAMETERS	39H	2 bytes: **Y1-Y2**

Used to set the barcode format and number of characters.

Y1 - bar code format. 01H = interleaved 2 of 5.
Y2 - number of characters (min 6, max 18).

EXTRACT BARCODE DATA

Controller Command	Code	Bill Validator Response Data
EXTRACT BARCODE DATA	3AH	n bytes: **Z1-Zn**

Z1-Zn - n bytes ASCII of barcode data, n is equal min 6 bytes, max 18 bytes. Data is sent most significant byte first. Parameter n is assigned by command SET BARCODE PARAMETERS.

This command may be sent at any time after the Bill Validator responds to the Poll command by event 80H, 81H or 82H and the bill type indicates barcode token presence (23). Barcode data of a successful reading is preserved until next bill will be inserted. Otherwise an ILLEGAL COMMAND response will be returned.

GET BILL TABLE

Controller Command	Code	Bill Validator Response Data
GET BILL TABLE	41H	120 bytes: **Z1-Z120**

Command for request bill type description.

Z1-Z120 The 120 - byte string consists from 24 five-byte words.
Byte 1 of word – hex representation most significant digit of the denomination.
Bytes 2-4 of word – country code in ASCII characters.
Byte 5 of word – this byte used to determine decimal placement or proceeding zeros. If bit D7 is 0, the bits D0-D6 indicate the number of proceeding zeros. If bit D7 is 1, the bits D0-D6 indicate the decimal point position starting from the right and moving left.

A five-byte position in the 120-bytes string indicates bill type description for the particular bill type. For example, first five byte correspond bill type=0, second five byte correspond bill type=1 and so on.

Example:

Bill Type	Denomination Code First Byte	Country Code 3 bytes	Denomination Code Second Byte	Denomination	
0	0x1	ITL	0x3	1,000	Lira
1	0x2	ITL	0x3	2,000	Lira
2	0x5	ITL	0x3	5,000	Lira
3	0x5	NLC	0x82	.05	Crown
4	0x19	NLC	0x82	.25	Crown
5	0x19	NLC	0x81	2.5	Crown
6	0x19	NLC	0x0	25	Crown
7	0x19	NLC	0x2	2500	Crown
8	0x1	USA	0x0	1	Dollar
9	0x5	USA	0x0	5	Dollar
10	0x1	USA	0x1	10	Dollar
11	0x2	USA	0x1	20	Dollar

Unsent bytes are assumed to be zero.

This command is valid in the following states only: Power up, Initialize, one of the Failure states (41H-47H) or Unit Disabled.

GET CRC32 OF THE CODE

Controller Command	Code	Bill Validator Response Data
GET CRC32 OF THE CODE	51H	4 bytes: **Z1-Z4**

Z1-Z4 - 4 bytes of CRC, MSB first.

This command is valid in the following states: Power up, Initialize, one of the Failure states (41H-47H) or Unit Disabled.

DOWNLOAD

Controller Command	Code	Controller Data
DOWNLOAD	50H	NONE

Command for transition Bill Validator to download mode.

Upon receipt of this command and issuing of confirmation ACK the Bill Validator transits to download mode. In this mode the Bill Validator receives DOWNLOAD command with sub-commands only. All other commands will be ignored.

Issuing of confirmation ACK by Bill Validator to the DOWNLOAD command is delayed for up to the time of command execution (200 ms).

This command is valid in the following states: Power up, Initialize, one of the Failure states (41H-47H) or Unit Disabled.

Controller Command	Code	Sub-Command	Bill Validator Response Data
DOWNLOAD	50H	00H	1 byte: **Z1**

(Switch to DOWNLOAD/Request Status)

Command for polling boot loader program status. This command should be send for transition to download mode and after issuing operation command.

Z1 If Z1= 00H boot loader program is ready to execute download commands;
If Z1= E0H boot loader program has executed previous operation command successfully;
If Z1= E1H boot loader program has executed previous operation command with error.

Controller Command	Code	Sub-Command	Bill Validator Response Data
DOWNLOAD	50H	01H	1 byte: **Z1**

(Request block size)

Request for memory block size. Further programming with “DOWNLOAD – Write To Memory” must be performed with blocks of this size. No defaults are defined for data block size, and no assumptions should be made – valid response to block size request is the only way of evaluating the actual block size for the session given.

Z1 DATA block size index. Actual block size is 2^Z1.
Ex: if DATA block size is 512 bytes then size index is 9 (512=2^9).

Controller Command	Code	Sub-Command	Controller Data	Bill Validator Data
DOWNLOAD	50H	02H	(n+5) bytes: Y1-Y(n+5)	1byte:**Z1**

(Write to memory)

Command for writing data to memory.

Y1-Y2 Length of message;
Y3-Y5 AdrLow, AdrMid, AdrHigh are the address first byte of n Data bytes;
Y6-Y(n+5) n Data bytes (following requested block size).
Z1 if Z1=E0H – data received successfully
If Z1=E1H – data received with error

Controller Command	Code	Sub-Command	Bill Validator Response Data
DOWNLOAD	50H	03H	1 byte: **Z1**

(Exit)

Command for exiting to normal operation.

Z1 If Z1= E0H - CRC of freshly written code matched the one supplied with code itself;
If Z1= E2H - CRC of freshly written code did not match the one supplied with code itself;

REQUEST STATISTICS

Controller Command	Code	Bill Validator Response Data
REQUEST STATISTICS	60H	4358bytes: **Y1-Y4358**

Command for retrieving statistical data

To describe the Bill Validator performance characteristics more meaningfully all cumulative data are grouped into three aggregates of counters. The first aggregate is formed by 16 sets of 96 counters each. The second aggregate is formed by 20 sets of 46 counters each. The third aggregate is formed by 20 sets of one counter each. Each aggregate keeps track of Bill Validator events on a different statistical basis.
In the first aggregate sets are averaged by 10000 inserted bills, in the second aggregate sets are averaged by 500 inserted bills, and in the third aggregate – by 50 inserted bills.
The first aggregate counters are populated after insertion of every 10000 bills is completed, second - after insertion of 500 bills, third - after insertion of 50 bills. Aggregates contain circular buffers, i.e. one set of counters in every aggregate always contains current results. For every aggregate counter sets are output the most recent set last, therefore if circular buffer is not full zeroes will precede normal statistic data.
Such organization of the statistical data allows analyzing change of the characteristics of Bill Validator depending on number of the inserted bills for various scales with point's 10000 bills, 500 bills and 50 bills. Below as an example the diagrams of change of some characteristics for these three scales are given.

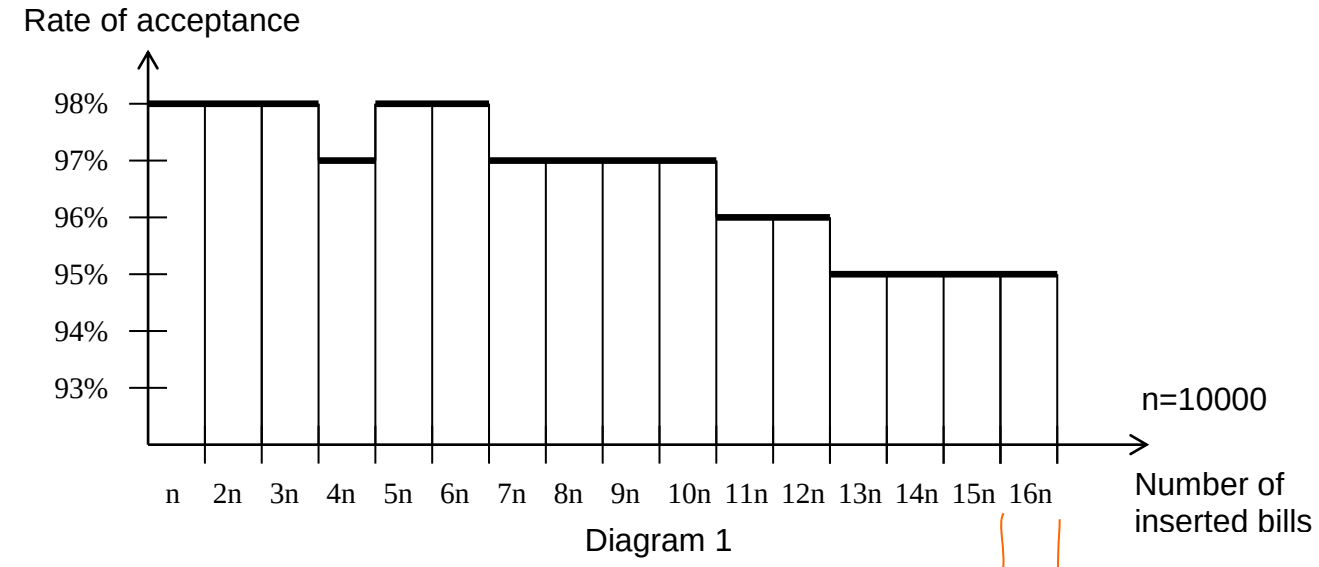


Diagram 1

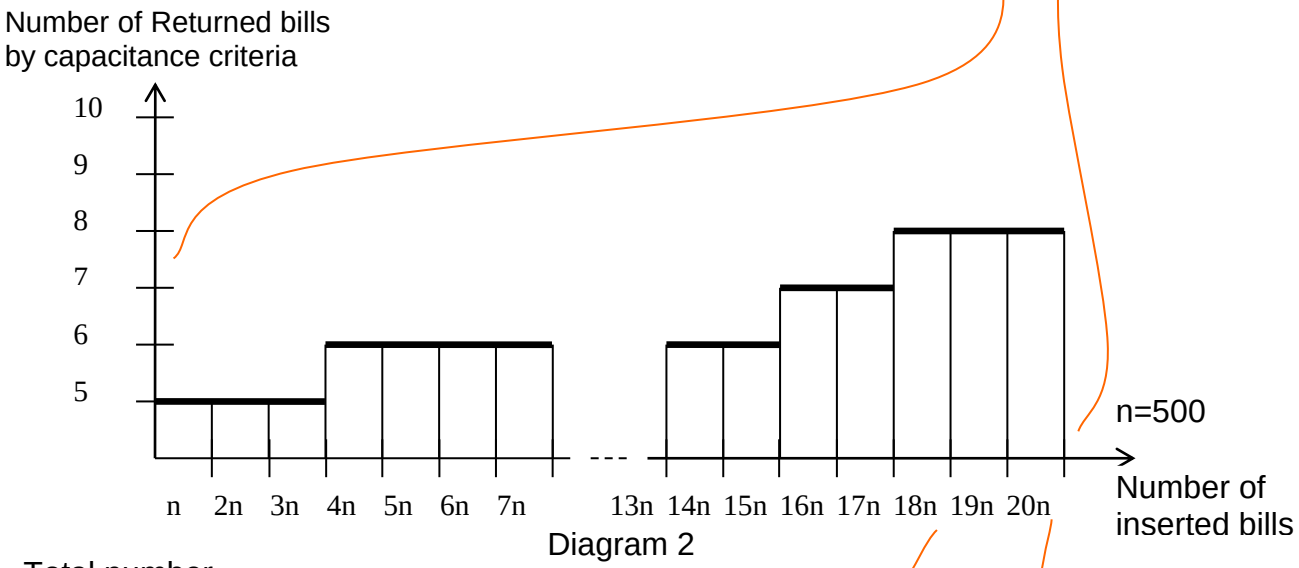


Diagram 2

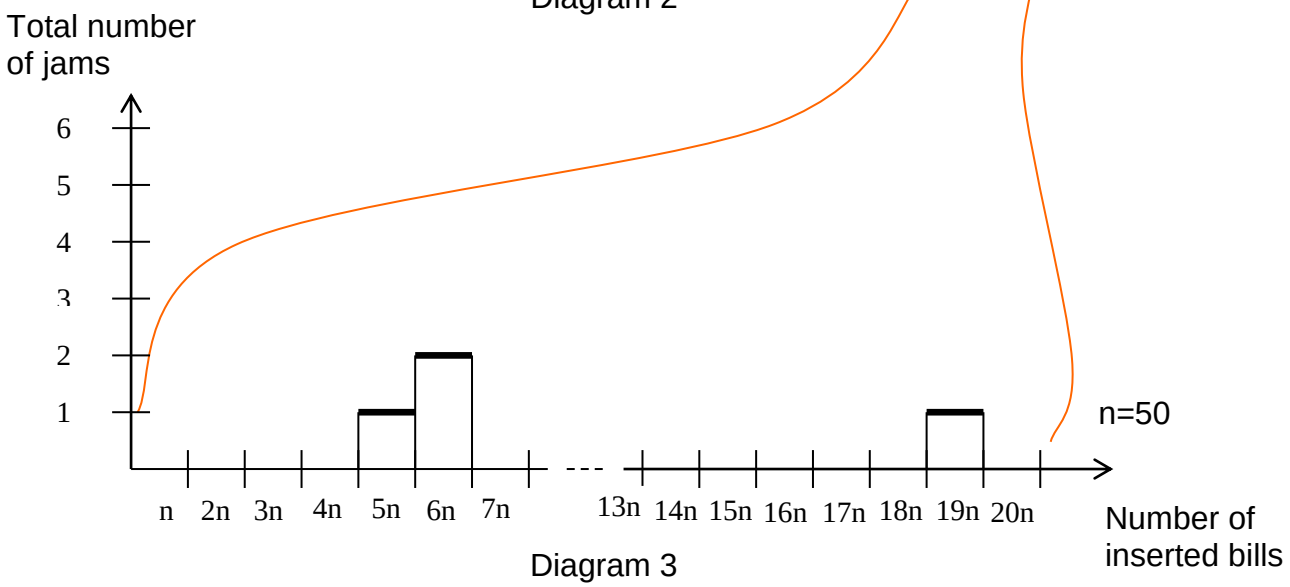


Diagram 3

In response to a REQUEST STATISTICS command Bill Validator sends Y1-Y4358 bytes. Y1-Y2 is length of message (see 2.3). Y3-Y4358 corresponds to the following table.

	Type of Data	Numbers of bytes
1.	Identification data	32 (Y3-Y34)
2.	16 sets of first aggregate	16x179=2864 (Y35-Y2898)
3.	20 sets of second aggregate	20x66=1320 (Y2899-Y4218)
4.	20 sets of third aggregate	20x1=20 (Y4219-Y4238)
5.	Table of denominations	100 (Y4239-Y4338)
6.	Currency names block	20 (Y4339-Y4358)

Format of identification data block:

Bytes	Description
Z1-Z3	Country Code – 3 bytes, ASCII characters
Z4	Denomination count – 1 byte (number of denominations accepted), binary
Z5	Data format version – 1 byte, RESERVED FOR FACTORY USE
Z6-Z12	Last statistics reset Date and Time – 8 bytes, format shown below: - LSB of year (ex: for year 2002, it's 210 decimal), 1 byte (Z6); - MSB of year (ex. for year 2002, it's 7 decimal), 1 byte (Z7); - Month, 1 byte (Z8); - Day, 1 byte (Z9); - Hour (24-hour based), 1 byte (Z10); - Minutes, 1 byte (Z11); Seconds, 1 byte (Z12).
Z13	Reserved
Z14-Z25	Serial Number – 12 bytes, ASCII
Z26-Z32	Reserved

Format of statistical data for one set of 1/10000 aggregate:

Z1-Z2	Number Inserted bills
Z3-Z4	Number of Accepted bills for 1st denomination
Z5-Z6	Number of Accepted bills for 2nd denomination
…	**…**
Z41-Z42	Number of Accepted bills for 20th denomination
Z43-Z44	Number of Returned bills by optical criteria for 1st denomination
Z45-Z46	Number of Returned bills by optical criteria for 2nd denomination
…	**…**
Z81-Z82	Number of Returned bills by optical criteria for 20th denomination
Z83-Z84	Number of Returned bills by magnetic criteria for 1st denomination
Z85-Z86	Number of Returned bills by magnetic criteria for 2nd denomination
…	**…**
Z121-Z122	Number of Returned bills by magnetic criteria for 20th denomination
Z123-Z124	Number of Returned bills by capacitance criteria for 1st denomination
Z125_Z126	Number of Returned bills by capacitance criteria for 2nd denomination
…	**…**
Z161-Z162	Number of Returned bills by capacitance criteria for 20th denomination
Z163-Z164	Number of Unrecognized bills

Z165-Z166	Number of Returned bills by length
Z167	Number of Failures by magnetic
Z168	Number of Failures by optics
Z169	Number of Failures of transport mechanism
Z170	Number of Failures of stacking mechanism
Z171	Number of Failures of aligning mechanism
Z172	Number of Jams in drop cassette
Z173	Number of Jams at entrance sensor
Z174	Number of Jams at both entrance and aligning sensors
Z175	Number of Jams at aligning sensor
Z176	Number of Jams at aligning and optical sensors
Z177	Number of Jams at optical sensor
Z178	Number of Jams at optical and exit sensors
Z179	Number of Jams at exit sensor

*** - Number of denominations accepted includes both old and new denominations, if they're present. Order of denominations is listed in table of denominations**

Format of statistical data for one set of 1/500 aggregate:

Z1-Z2	Number of Accepted bills for 1st denomination
Z3-Z4	Number of Accepted bills for 2nd denomination
…	**…**
Z39-Z40	Number of Accepted bills for 20th denomination
Z41	Number of Returned bills by verification for 1st denomination
Z42	Number of Returned bills by verification for 2nd denomination
…	**…**
Z60	Number of Returned bills by verification for 20th denomination
Z61	Number of Returned bills by optical criteria
Z62	Number of Returned bills by magnetic criteria
Z63	Number of Returned bills by capacitance criteria
Z64	Number of Jams at bezel
Z65	Number of Jams at transport path
Z66	Total number of Jams

Format of statistical data for one set of 1/50 aggregate:

Z1	Total number of Jams

Format of denominations table:

Z1-Z100 The 100 byte string consisting of 20 five-byte words.
Two most significant bits of first byte in every word – denomination type (old or new).

Coding of a denomination type:

7th bit	**6th bit**	**denomination type**
0	0	Old denomination
0	1	New denomination

Other bits of first byte – hex representation of the most significant digit of the denomination.

Bytes 2-4 of word – country code in ASCII characters.
Byte 5 of word –used to determine decimal placement or proceeding zeros. If bit D7 is 0, the bits D0-D6 indicate the number of proceeding zeros. If bit D7 is 1, the bits D0-D6 indicate the decimal point position starting from the right and moving left.

Example:

Denomination Code First Byte	Country Code 3 bytes	Denomination Code Second Byte	Denomination
0x1	USA	0x0	1 Dollar
0x5	USA	0x0	5 Dollars
0x1	USA	0x1	10 Old Dollars
0x41	USA	0x1	10 New Dollars
0x2	USA	0x1	20 Old Dollars
0x42	USA	0x1	20 New Dollars
0x5	USA	0x1	50 Old Dollars
0x45	USA	0x1	50 New Dollars
0x1	USA	0x2	100 Old Dollars
0x41	USA	0x2	100 New Dollars
0x1	MXN	0x1	10 Pesos
0x2	MXN	0x1	20 Old Pesos
0x42	MXN	0x1	20 New Pesos
0x5	MXN	0x1	50 Old Pesos
0x45	MXN	0x1	50 New Pesos
0x1	MXN	0x2	100 Pesos
0x2	MXN	0x2	200 Pesos
0x5	MXN	0x2	500 Pesos

Format of currency names block:

Names of currencies comprise a formatted ASCII data string where currency names appear in a sequence corresponding to the table of denominations. Currency names are separated by slash. String is terminated with dot.

Example:
Dollar/Peso.

Command REQUEST STATISTICS is valid in the following states: Power up, Initialize, one of the Failure states (41H-47H) or Unit Disabled.

If statistic data is not correct Bill Validator sends an error response with data field set to 31H (INVALID STATISTIC DATA).

6 CONTROLLER/ Card Reader Communication Specification

6.1 Introduction

This section defines the communications bytes sent and received between Card Reader and the Controller. The Card Reader ‘s address is 04H.

Unless stated otherwise, all information is assumed to be in hexadecimal format. The numbers will be sent most significant byte first.

6.2 Card Reader States

Card Readers may be viewed as state machines. These states are as follows:

1) Inactive
2) Disabled
3) Idling
4) Ready for Transaction
5) Vending
6) Busy
7) Vend OK/Vend Failed

6.2.1 Inactive

This is the state of the card reader at power up or after a reset. All cards except for stored value cards (for balance inquiry after internal initialization completes) will not be accepted. The card reader cannot leave this state until all SETUP information is received from the Controller.

6.2.2 Disabled

The card reader automatically enters this state from the Inactive state when it has received all SETUP information from the Controller and completes its internal initialization. It will also enter the Disabled state from the Idling state when it receives the READER/DISABLE command. While in the Disabled state, stored value cards will be accepted (for balance inquiries), but no vending requests will be granted.

6.2.3 Idling

In this state, cards may be used for transactions. The card reader will remain in this state until a valid card is read (when it will enter the Ready for Transaction state), a READER/DISABLE command is received (when it will return to the Disabled state) or a RESET is received (when it will enter the Inactive state).

6.2.4 Ready for Transaction

In the Idling state, when a valid card is processed, the card reader will enter the Ready for Transaction state. This indicates that the card reader is available for vending activities. The only structured exits from the Ready for Transaction state are:

- Through the VEND/SESSION COMPLETE subcommand from the Controller (for a no-value cards, ex. debit and credit cards; for stored value cards this command makes no sense but will not generate an error);
- Through card removal (for stored value cards only).

Other VEND subcommands will cause the card reader to leave the Ready for Transaction state and enter the Vending state when products are purchased.

6.2.5 Vending

This state is entered from the Ready for Transaction state upon reception of a VEND/VEND REQUEST command from the Controller.

6.2.6 Busy

This state is entered when Card Reader starts performing internal operations and exited upon completion of internal operations. Exit is done to the state present before entering Busy, except for error conditions appeared while in Busy.

6.2.7 Vend OK/Vend Failed

This state is entered after processing Vend Request command. When all transactions with card and/or bank are finalized this state is returned to POLL command and cleared upon successful readout or kept until successful readout in case of communication errors.

6.3 Command Protocol

The card reader will provide an informational response immediately with the requested data.

6.4 Controller Commands

COMMAND	HEX CODE	DESCRIPTION
RESET	*30H*	Command for Card Reader to self-reset.
SETUP	*11H*	Send/Request Card Reader setup status.
POLL	*33H*	Request for Card Reader activity status.
VEND	*13H*	Vend state control.
ENABLE/DISABLE	*14H*	Disabled/Enabled state control.
IDENTIFICATION	*15H*	Request for Model, Serial Number, Software Version, Localization ISO code.
DOWNLOAD	*50H*	Command for transition to download mode.

6.5 Controller Command Format

RESET

Controller Command	Code	Controller Data
RESET	30H	No data bytes

This command is the vehicle that the Controller must use to tell the Card Reader that it must return to the Inactive state. With the exception of the ACK response, it must abort all communication, terminate any ongoing transaction (with a refund, if appropriate), eject the card (if applicable), and go to the Inactive state until otherwise instructed by the Controller.

The Controller must follow the RESET command with the SETUP and ENABLE/DISABLE commands to enable vending transactions. RESET command is not valid for Vending, Busy and Vend OK/Vend Failed states. If received with any of these states active, a COMMAND INVALID response is issued.

SETUP

Controller Command	Code/Subcommand	Controller Data	Response Data
SETUP	**11H**	**Y1**	**Z1 – Z2**

Y1 = Controller Capabilities Level
Indicates the highest capabilities level the Controller supports. Currently, this byte is set to 01.

RESPONSE - CARD READER CONFIGURATION:

Indicates the Card Reader is responding to a SETUP request from the Controller. This response includes the following data:

Z1 = Card Reader capabilities level
Capabilities level of the Card Reader. Currently the highest capabilities level is 01.

Z2 = Miscellaneous options

b0 Fund Restoring capable
- 0 = the card reader is NOT capable of restoring funds to the user's card or account. Do not request refunds.
- 1 = the card reader is capable of restoring funds to the user's card or account. Refunds may be requested.

b1 Continuous Payment capable
- 0 = the card reader is NOT capable of collecting funds from multiple cards for single payment. Money withdraw request will be denied if funds are insufficient in authorized card.
- 1 = the card reader is capable of collecting funds from multiple cards for single payment. Money withdraw request will be approved even if funds are insufficient in authorized card

Other bits are ignored (under card reader capabilities level 01) and may be set to any value.

POLL

Controller Command	Code	Controller Data	Response Data
POLL	33H	No data	**Z1 up to Z2**

The POLL command is used by the Controller to obtain information from the Card Reader. This information may include user actions, hardware malfunctions, software malfunctions or information explicitly requested by the controller.
Controller may receive the following POLL responses from the Card Reader:

Response data bytes		Description	
Z1	**Z2**		
01H		Inactive	Indicates the Card Reader has been reset/re-powered.
05H		Disabled	Indicates the Card Reader has received all SETUP information from the Controller. Card Reader will also enter the Disabled state from the Enabled state when it receives the ENABLE/DISABLE command.
06H		Idling	In this state cards may be used for transactions. Card Reader will enter the Enabled state from the Disabled state when it receives the ENABLE/DISABLE command.
07H		Ready for Transaction	Indicates the Card Reader is available for vending activity.
08H		Vending	This state is entered from the Idling state upon reception of a VEND REQUEST command from the Controller.
12H		Vend OK	Transaction successfully completed.
13H		Vend Failed	Transaction failed for a reason not relevant to the Controller.
09H	**Z2**	Malfunction/Error	The Card Reader is reporting a malfunction or error.
10H		Delayed	Indicates that a full response to a command is not available right now. This may occur, for example, while card reader is dialing out to the bank.
11H		Busy	Card Reader is busy with internal operations

Z1 = 01H **INACTIVE:**
Indicates the Card Reader has been reset, due to either an external RESET or an internally detected condition.

Z1 = 05H **DISABLED:**
Indicates the Card Reader has received all SETUP information from the Controller and completed its internal initialization. Card Reader will also enter the Disabled state from the Idling state when it receives the ENABLE/DISABLE command.

Z1 = 06H **IDLING:**
In this state cards may be used for transactions. Card Reader will enter the Idling state from the Disabled state when it receives the ENABLE/DISABLE command.

Z1 = 07H **READY FOR TRANSACTION:**
Indicates the Card Reader is available for vending activity.

Z1 = 08H **VENDING:**
This state is entered from the Idling state upon reception of a VEND REQUEST command from the Controller.

Z1 = 12/13H **VEND OK/VEND FAILED:**

This state is entered from the Vending state upon completion of a transaction.

Z1 = 09H **MALFUNCTION/ERROR:**

The Card Reader is reporting a malfunction or error. This response includes the following information:

Z2 = Error Code = xxxxyyyy

xxxx =

0010: Card Error (e.g. stored value card removed while transaction is in progress, or a no-value card read incomplete)[1] or [2]
0100: Communications Error (checksum error/data frame inconsistent)[2]
1000: Reader Failure[3]

Other values not defined under card reader capabilities level 01.

[1]Transient error Reported once.

[2]Non-transient error reported every POLL until cleared. Card Reader functional after error cleared.

[3]Non-transient error reported every POLL until cleared. Card Reader not presently functional.
yyyy = Manufacturer defined sub-code

VEND/BEGIN SESSION

Controller Command	Code/Subcommand	Controller Data	Response Data
VEND/BEGIN SESSION	13H	**Y1**	**none**

Y1 = 10H **VEND BEGIN SESSION - subcommand.**
Indicates that VMC is requesting card authentication from the Card Reader. This command may only be issued in Idling state, otherwise a COMMAND INVALID will be returned. If a stored value card is already authenticated, the Ready for Transactions state will be entered immediately. For the no stored value card an invitation is issued for the user to insert/swipe a card, and after completed, Ready for Transactions state is entered. It's at Controllers discretion to terminate this condition if a time-out occurs without a card being authenticated, as this state is never left on Card Reader's initiative for no stored value cards and left on Card Reader's initiative when card is removed for stored-value cards.

VEND/GET FUNDS

Controller Command	Code/Subcommand	Controller Data	Response Data
VEND/GET FUNDS	13H	**Y1**	**Z1-Zn**

Y1 = 00H **VEND GET FUNDS - subcommand.**

Indicates that VMC is requesting a balance from the card. This command may only be issued in Ready for Transactions state, otherwise a COMMAND INVALID will be returned.

Z1 = Flag indicating that a card contains stored value:
E0H…E7H – the card is a stored value card;
E8H…EFH - the card is not a stored value card**;**

All subsequent data exists for stored value cards only and consists of N 8-byte blocks, one block per each "wallet" with different currency in the card. Wallets are numbered sequentially as they arrive in response to VEND/GET FUNDS, starting from 0.

Z2-Z4 = Currency code, three ASCII characters, following ISO 4217;

Z5-Z8 = Wallet balance in "composite" format, i.e. binary unsigned integer value; Z9 least significant bits compose the fractional part – to obtain the exact value Z5…Z8 should be treated as a 32-bit binary value, then divided by 10^Z9; the division result is the integer part of funds value, the remainder stays for the fractional part.

Z9 = Number of decimal places.

VEND/VEND REQUEST

Controller Command	Code/Subcommand	Controller Data	Response Data
VEND/ VEND REQUEST	13H	**Y1 – Y10**	**none**

Y1 = 01H **VEND REQUEST - subcommand.**
Indicates the customer has made a selection. The Controller is requesting vend approval from the Card Reader before dispensing the product. The Card Reader enters state Vending after reception of this command and keeps it until transaction is finalized (state Vend OK is entered) or error occurs (state Vend Fail is entered). The state is preserved until successful read-out.

Y1 = 02H **VEND REQUEST REFUND - subcommand.**
Indicates the controller is requesting a refund for the given wallet with amount given. The Card Reader enters state Vending after reception of this command and keeps it until transaction is finalized (state Vend OK is entered) or error occurs (state Vend Fail is entered). The state is preserved until successful read-out.

VEND REQUEST and **VEND REQUEST REFUND** commands are valid only in Ready For Transaction state, otherwise COMMAND INVALID is issued.

Y2 = Wallet number. Wallets are numbered serially starting from 0 as returned by VEND/GET FUNDS command. If card is not a stored value card Y2 may be assigned any value.

Y3 – Y6 = Amount to draw in "composite" format, same as for VEND/GET FUNDS command.

Y7 – Y10 = Accumulated amount in "composite" format, same as for VEND/GET FUNDS command. This field represents amount, accumulated for the given card session (i.e. it is set 0 prior to issuing VEND/BEGIN SESSION and incremented prior to sending every VEND/VEND REQUEST with requested value). This value is used by Card Reader to trace retransmissions of VEND/VEND REQUEST issued by Controller.

Separate accumulated amounts are maintained for every wallet reported by the Card Reader.

VEND/VEND CANCEL

Controller Command	Code/Subcommand	Controller Data	Response Data
VEND/ VEND CANCEL	13H	**Y1**	**Z1**

Y1 = 03H **VEND CANCEL - subcommand.**
This command can be issued by the Controller to cancel a VEND REQUEST command before the Card Reader has sent a VEND APPROVED/DENIED. The Card Reader will respond to VEND CANCEL with a VEND DENIED or VEND CANCEL FAILED and return to the Idling state. State preserved until successful read-out. This command is valid only in Vending state, otherwise COMMAND INVALID response is issued.

Z1 = 06H **VEND DENIED:**
Approval denied for the customer's selection. Do not dispense any products/credit services.

Z1 = 07H **VEND CANCEL FAILED:**
Funds are already withdrawn and may not be returned. Products/services should be dispensed.

VEND/SESSION COMPLETE

Controller Command	Code/Subcommand	Controller Data
VEND/ SESSION COMPLETE	13H/04H	**Y1**

Y1 = 04H **SESSION COMPLETE – subcommand.**

This tells the Card Reader that the session is complete and to return to Idling state.

ENABLE/DISABLE

Controller Command	Code/Subcommand	Controller Data
READER/DISABLE	14H	**Y1**

Any transaction in progress will not be affected and must continue to its normal completion.

Y1 = 00H **DISABLE - subcommand.**

This informs the Card Reader that it has been disabled, i.e. it must no longer accept a customer's card for the purpose of vending. Vending activities may be re-enabled using the READER ENABLE command. The Card Reader must retain all SETUP information.

Y1 = 01H **ENABLE - subcommand.**

This informs the Card Reader that it has been enabled, i.e. it must be ready for accepting cards.

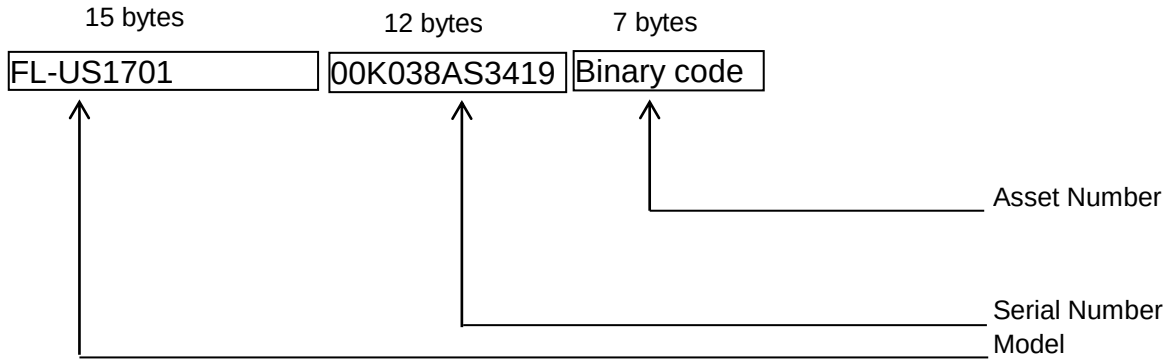
Reader may be enabled from Disabled state only and disabled from Idling state only, otherwise a COMMAND INVALID response will be issued.

IDENTIFICATION

Controller Command	Code	Card Reader Response Data
IDENTIFICATION	15H	**Z1 – Z34**

Bytes	Description
Z1-Z15	Part Number – 15 bytes, ASCII characters
Z16-Z27	Serial Number – 12 bytes Factory assigned serial number, ASCII characters
Z28-Z34	Asset Number – 7 bytes, unique to every Card Reader, binary data

Bytes Z1-Z27 must be sent as ASCII Characters. Zero (30H) and Blank (20H) are acceptable. Asset Number must be sent as binary code.



DOWNLOAD

Controller Command	Code	Controller Data
DOWNLOAD	50H	No data bytes

Command for transition Card Reader to download mode. Currently not defined and returns COMMAND INVALID response.

6.6 Non-Response Time

The maximum non-response time for a Card Reader is 5 seconds. This is the maximum time for which a Card Reader will not respond to a command with ACK, NAK or a data message.

7 APPENDIX

Example CCNET Message Sequences

7.1 Power Up & Reset sequence

Power Up & Reset sequence			
Controller		**Bill-to-Bill unit**	**Comments**
POLL	------>		
	<------	POWER UP	Power is switched on
ACK	------>		
RESET	------>		Reset peripheral
	<------	ACK	
POLL	------>		
	<------	INITIALIZE	Bill-to-Bill unit is initializing
ACK	------>		
GET STATUS	------>		Collect operational parameters
	<------	BILL-TO-BILL UNIT CONFIG.	
ACK	------>		
SET SECURITY	------>		Update bill security levels
	<------	ACK	
IDENTIFICATION	------>		Collect asset inf.
	<------	BILL-TO-BILL UNIT ID	
ACK	------>		
POLL	------>		
	<------	INITIALIZE	Bill-to-Bill unit is initializing
ACK	------>		
POLL	------>		
	<------	UNIT DISABLE	All bill types are disabled
ACK	------>		

7.2 Enable sequence

Enable sequence			
Controller		**Bill-to-Bill unit**	**Comments**
ENABLE BILL TYPES	------>		Enable appropriate bill types
	<------	ACK	
POLL	------>		
	<------	IDLING	Bill-to-Bill unit is waiting accepting of bill
ACK	------>		

7.3 Disable sequence

Disable sequence			
Controller		**Bill-to-Bill unit**	**Comments**
ENABLE BILL TYPES	------>		Disable all bill types
	<------	ACK	

7.4 Bill Accept sequence (Bill stacked).

Bill Accept sequence - Bill stacked -			
Controller		**Bill-to-Bill unit**	**Comments**
POLL	------>		
	<------	ACCEPTING	Bill is accepting
ACK	------>		
POLL	------>		
	<------	ESCROW POSITION	Bill type in escrow position
ACK	------>		
	. . .		
POLL	------>		
	<------	ESCROW POSITION	Bill type in escrow position
ACK	------>		
STACK	------>		Send bill to drop cassette or one of the recycling cassette
	<------	ACK	
POLL	------>		
	<------	STACKING	Bill is stacking
ACK	------>		
	. . .		
POLL	------>		
	<------	BILL STACKED	Bill has been stacked
ACK	------>		
CASSETTE STATUS	------>		Collect operational parameters
	<------	BILL-TO-BILL UNIT CASSETTE STATUS	
ACK	------>		
ENABLE BILL TYPES	------>		Enable appropriate bill types
	<------	ACK	
POLL	------>		
	<------	IDLING	Bill-to-Bill unit is waiting accepting of bill
ACK	------>		

7.5 Bill Accept sequence (Bill returned)

Bill Accept sequence - Bill returned -			
Controller		**Bill-to-Bill unit**	**Comments**
POLL	------>		
	<------	ACCEPTING	Bill is accepting
POLL	------>		
	<------	ESCROW POSITION	Bill type in escrow position
	. . .		
POLL	------>		
	<------	ESCROW POSITION	Bill type in escrow position
ACK	------>		
RETURN	------>		Return bill to consumer
	<------	ACK	
POLL	------>		
	<------	RETURNING	Bill is returning
ACK	------>		
	. . .		
POLL	------>		
	<------	BILL RETURNED	Bill has been returned
ACK	------>		
ENABLE BILL TYPES	------>		Enable appropriate bill types
	<------	ACK	
POLL	------>		
	<------	IDLING	Bill-to-Bill unit is waiting acceptance of bill
ACK	------>		

7.6 Bill Dispense sequence (Bill dispensed).

Bill Dispense sequence - Bill returned -			
Controller		**Bill-to-Bill unit**	**Comments**
DISPENSE	------>		Dispensed bills to customer
	<------	ACK	
POLL	------>		
	<------	BILL DISPENSING	Bill transporting to dispenser
ACK	------>		
	. . .		
POLL	------>		
	<------	BILL DISPENSED	Dispense is completed. Bills pay to customer.
ACK	------>		
CASSETTE STATUS	------>		Collect operational parameters
	<------	BILL-TO-BILL UNIT CASSETTE STATUS	
ACK	------>		
POLL	------>		
	<------	IDLING	Bill-to-Bill unit is waiting accepting of bill
ACK	------>		

7.7 Bill Unload sequence (Bill unloaded).

Bill Unload sequence - Bill unloaded -			
Controller		**Bill-to-Bill unit**	**Comments**
UNLOAD	------>		Stacking all bills from Recycling Cassettes to drop cassette. (Unload Bill-to-Bill unit)
	<------	ACK	
POLL	------>		
	<------	BILL UNLOADING	Bill transporting to drop cassette.
ACK	------>		
	. . .		
POLL	------>		
	<------	BILL UNLOADED	Unload is completed. All Recycling Cassettes are empty. (All banknotes stacked in drop cassette.
ACK	------>		
CASSETTE STATUS	------>		Collect operational parameters
	<------	BILL-TO-BILL UNIT CASSETTE STATUS	
ACK	------>		
POLL	------>		
	<------	IDLING	Bill-to-Bill unit is waiting acceptance of bill
ACK	------>		

7.8 Set cassette type sequence

Set cassette type sequence			
Controller		**Bill-to-Bill unit**	**Comments**
CASSETTE STATUS	------>		Collect operational parameters
	<------	BILL-TO-BILL UNIT CASSETTE STATUS	Cassette necessary for setting is not empty.
ACK	------>		
SET CASSETTE TYPE	------>		Assigning cassette to bill type
	<------	ACK	
POLL	------>		
	<------	SETTING CASSETTE TYPE	Bills transporting to drop cassette.
ACK	------>		
	. . .		
POLL	------>		
	<------	SET CASSETTE TYPE	Setting recycling cassette type is completed.
ACK	------>		
CASSETTE STATUS	------>		Collect operational parameters
	<------	BILL-TO-BILL UNIT CASSETTE STATUS	Necessary recycling cassette is empty. It is assigned to new bill type.
ACK	------>		
POLL	------>		
	<------	IDLING	Bill-to-Bill unit is waiting acceptance of bill
ACK	------>		

7.9 Power Up with Bill in Stacker sequence for Bill Validator

Power Up & Reset sequence			
Controller		**Bill Validator**	**Comments**
POLL	------>		
	<------	POWER UP WITH BILL IN STACKER	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 exit sensor and before the response to a poll command with the Bill Stacked event is sent.
ACK	------>		

RESET	------>		Reset peripheral
	<------	ACK	
POLL	------>		
	<------	INITIALIZE	Bill Validator is initializing
ACK	------>		
GET STATUS	------>		Collect operational parameters
	<------	BILL VALIDATOR CONFIG.	
ACK	------>		
SET SECURITY	------>		Update bill security levels
	<------	ACK	
IDENTIFICATION	------>		Collect asset inf.
	<------	BILL VALIDATOR ID	
ACK	------>		
POLL	------>		
	<------	INITIALIZE	Bill Validator is initializing
ACK	------>		
POLL	------>		
	<------	BILL STACKED	Bill Validator confirmation signal
ACK	------>		
POLL	------>		
	<------	UNIT DISABLE	All bill types are disabled
ACK	------>		