# CashCode® NET

## Document 1

## Common Specifications

December 2014

## Document revision history

| Rev. | Date | Author | Description |
|------|------|--------|-------------|
| 2.6.0 | Dec. 3, 2014 | CashCode Engineering | - Corrected footer and header<br>- Sequence of commands section 3.4/5.4 aligned according to section 3.3/5.3<br>- Restored command GET CRC16 OF THE CODE in command list for Bill-to-Bill unit (pt.3.3)<br>- Removed command WRITE / READ TABLE OF SHIFTS FOR DISPENSING (ONLY FOR CC-INSIDE USE) from Bill-to-Bill section<br>- For IDENTIFICATION command the set of applicable states was extended (pt.5.3)<br>- Removed table Validation result from description POOL command in Bill Validator section<br>- Technical correction |
| 2.5.7 | Sep. 8, 2014 | E.Alekseev | - Added LED OFF and HOLD BILL options to commands GET/SET OPTIONS for Bill-to-Bill and Bill Validator units.. |
| 2.5.7 | May 9, 2012 | CashCode Engineering | - Added for Bill Validator unit setTable48 option to GET/SETOPTIONS commands<br>- Changed commands GET/SET OPTIONS, ENABLE BILL TYPES, SET SECURITY, GET STATUS, GET BILL TABLE for Bill Validator unit.. |
| 2.5.6 | Mar. 22, 2012 | K. Zakharenkov | - Added response data 84h, 90h to POLL command. for Bill Validator section to indicate counterfeit and suspected bills<br>- Added Set/Get options commands to Bill Validator section for turning on/off Article 6 mode |
| 2.5.5 | Feb. 29, 2012 | E. Alekseev | - Added failure codes to POLL command for Bill Validator unit |
| 2.5.4 | Mar. 10, 2010 | CashCode Engineering | - Added description commands GET CRC32 OF THE CODE WITH PARAMETRS, GET CRC16 OF THE CODE WITH PARAMETRS to Bill Validator section<br>- Added description command SET ESCROW TIME OUTS to Bill Validator section |
| 2.5.3 | Aug. 18, 2009 | E. Alekseev | - Added description command GET CRC16 OF THE CODE to .Bill-To-Bill and Bill Validator sections. |

| 2.5.2 | Mar. 08, 2009 | CashCode Engineering | - Changed note to Response data 47h(Code1) of POLL command for Bill-to-Bill unit<br>- Command Protocol section changed/clarified for Controller/Bill Validator and Bill-to-Bill sections<br>- MODULE DOWNLOAD, VALIDATION MODULE IDENTIFICATION commands added to Controller/Bill Validator section<br>- SET/GET CAPACITY LIMIT OF CASSETTE commands added to Bill-to-Bill section<br>- Added 10sec timeout in PAUSE state for Bill Validator section<br>- Added note to ENABLE BILL TYPE command for Controller/Bill Validator and Bill-to-Bill sections<br>- Added note to EXTRACT BARCODE DATA command for Controller/Bill Validator and Bill-to-Bill sections<br>- Added Timeout control to SET OPTIONS command for Bill-to-Bill unit<br>- Added response data 11h,12h to POLL command for POWER UP with bill in Validator and Chassis for Bill-to-Bill Communication Specification<br>- Response data 1Dh/01h(DISPENSING) to POLL command changed/clarified for Bill-to-Bill unit |
|---|---|---|---|
| 2.5.1 | Nov. 01, 2007 | CashCode Engineering | - Manufacturer-specific command added to the protocol description for Bill Validator and Coin Changer units. All set of manufacturer-specific command for Bill-To-Bill and Bill Validator units moved to Document 7 |
| 2.5.0 | Sep. 10, 2007 | CashCode Engineering | - Manufacturer-specific command added to the protocol description for Bill-To-Bill unit |
| 2.4.5 | Jul. 07, 2006 | CashCode Engineering | - VALIDATION SOFTWARE IDENTIFICATION command added to Bill-To-Bill section |
| 2.4.4 | Jun. 22, 2006 | CashCode Engineering | - POWER RECOVERY command description and examples corrected |
| 2.4.3 | Apr. 28, 2006 | CashCode Engineering | - POWER RECOVERY commands added to Bill-To-Bill section<br>- SET OPTIONS and GET OPTIONS added to Bill-To-Bill section<br>- MODULE IDENTIFICATION REQUEST command added to Bill-To-Bill section<br>- Several timeout exceptions described |
| 2.4.2 | Jul. 18, 2005 | CashCode Engineering | - Escrow cassette functionality description changed. |
| 2.4.1 | Jun. 21, 2005 | CashCode Engineering | - Power recovery states added to section 3.4<br>- EMPTY DISPENSER command added to section 3.4<br>- SET OPTIONS and GET OPTIONS commands added to section 3.4<br>- Description of Bill-To-Bill TIME&DATE commands added to section 3.4 |

| 2.4 | Apr. 11, 2005 | CashCode engineering | - Version numbering revised to reflect internal/external numbering;<br>- Section 7.9 revised to reflect it's irrelevance to Bill-To-Bill operation;<br>- EXTENDED IDENTIFICATION command for the Bill-To-Bill – removed data sample from Bill-To-Bill devices software version definition;<br>- For DOWNLOAD command the set of applicable states was extended (pt.3.3).<br>- List of states added for SET BARCODE PARAMETERS command for Bill-To-Bill unit<br>- Failure state reporting format for Bill-To-Bill units changed (response 0x47 to the POLL command)<br>- List of states added to Bill Validator command list<br>- Command GET CRC32 OF THE CODE (code 0x51) added to Bill-To-Bill command set<br>- MODULE DOWNLOAD command (code 0x52) introduced to provide update of internal modules of multi-controller devices for Bill-To-Bill unit<br>- Revision number in the footer corrected<br>- Barcode related commands (0x39, 0x3A) added for Bill-To-Bill devices<br>- The EXTENDED IDENTIFICATION code changed to 0x3E to avoid conflicts<br>- IDENTIFICATION request (code 0x37) format for Bill-To-Bill has been changed<br>- EXTENDED IDENTIFICATION request (code 0x39) for Bill-To-Bill units has been added<br>- Revision format changed to comply with the company versioning standard |

**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 |

$$P(X) = X^{16} + X^{12} + X^5 + I$$

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

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

```
const CCNET_CRC_POLY = $08408

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

Sent in PERIPHERAL to CONTROLLER direction to confirm a command correctly received.
Sent in CONTROLLER to PEROPHERAL direction to confirm a data response correctly received.

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

| 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 |
|---------|------------|
| **00**H | *Forbidden* |
| **01**H | *Bill-to-Bill unit* |
| **02**H | *Coin Changer* |
| **03**H | *Bill Validator* |
| **04**H | *Card Reader* |
| **05**H | Reserved for Future Standard Peripherals |
| . | . |
| . | . |
| . | . |
| **0D**H | Reserved for Future Standard Peripherals |
| **0E**H | *Reserved for Future Broadcast Transmissions* |
| **0F**H | Reserved for Future Standard Peripherals |

## 2.5 Software Operational Rules

- Within 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 msec.
- 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 msec. It 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 dependent 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 msec 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 msec time-out (t-response) described in the Timing section of this document is the equivalent of a NAK message. **A Bill-to-Bill unit has some exceptions when it responds to IDENTIFICATION and RECYCLING CASSETTE STATUS commands. Refer to section 2.8 for details.**

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 msec of Bus silence, but for reliable operation of future multi-device buses the recommended value of $t_{free}$ is 20 msec.

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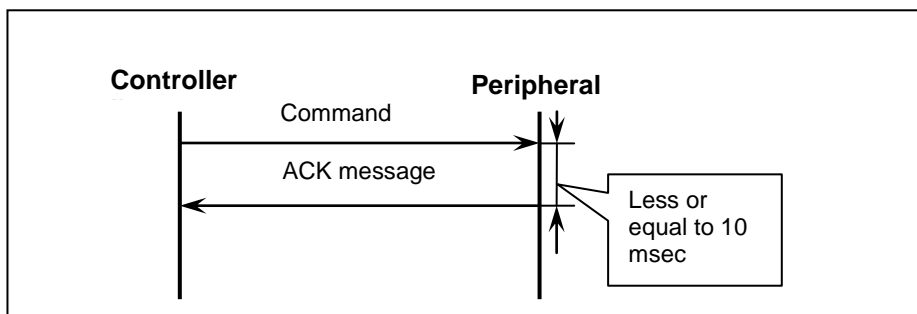


2.6.2    The diagram below represents a typical transmission when PERIPHERAL has data to return.
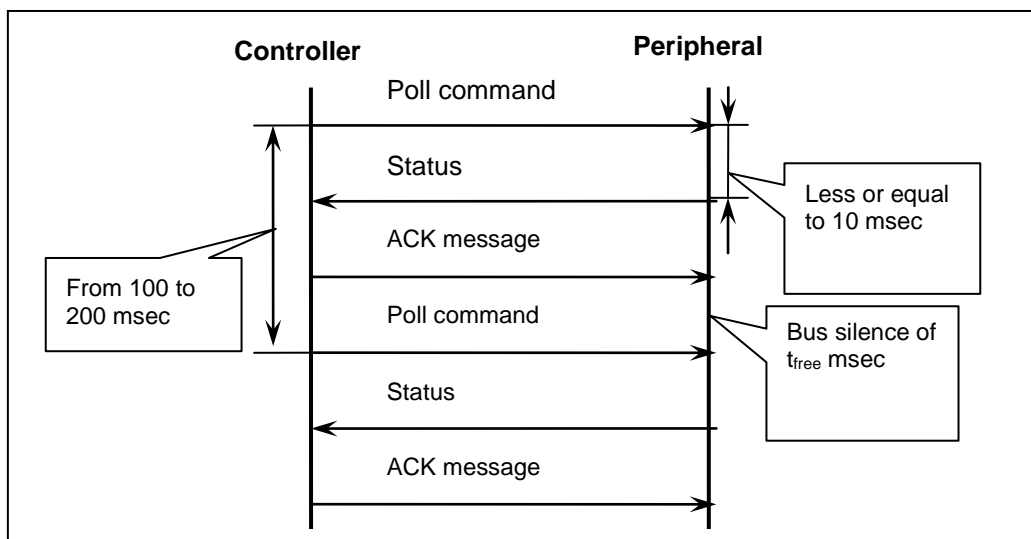
2.6.3    The diagram below represents a typical transmission when the Controller determines a CRC is not correct.
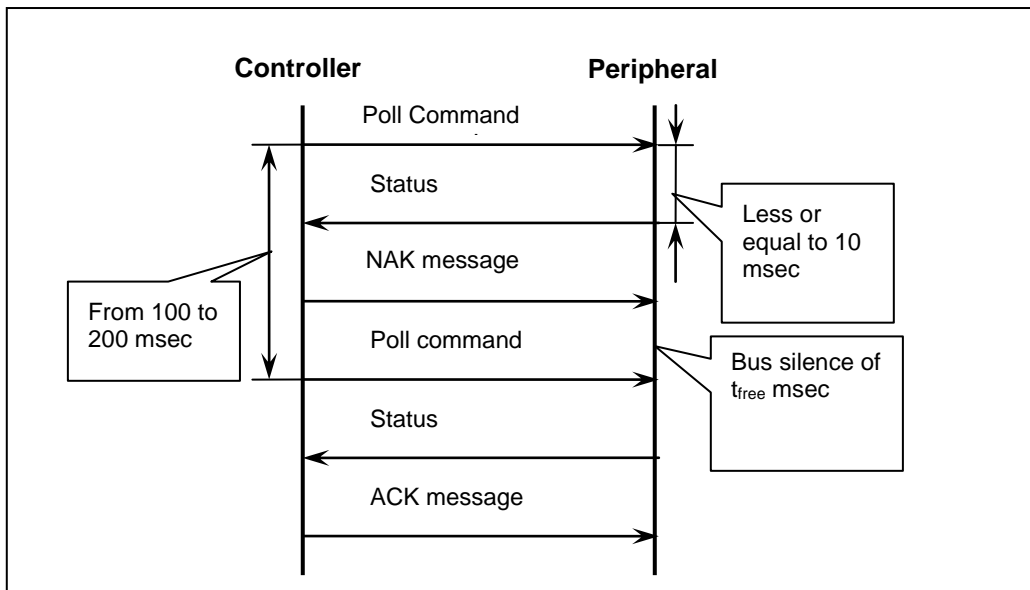


2.6.4    The diagram below represents a typical transmission when Peripheral determines a CRC is not correct. The Peripheral responds by sending a NAK message to the Controller to indicate that the information was not received correctly.
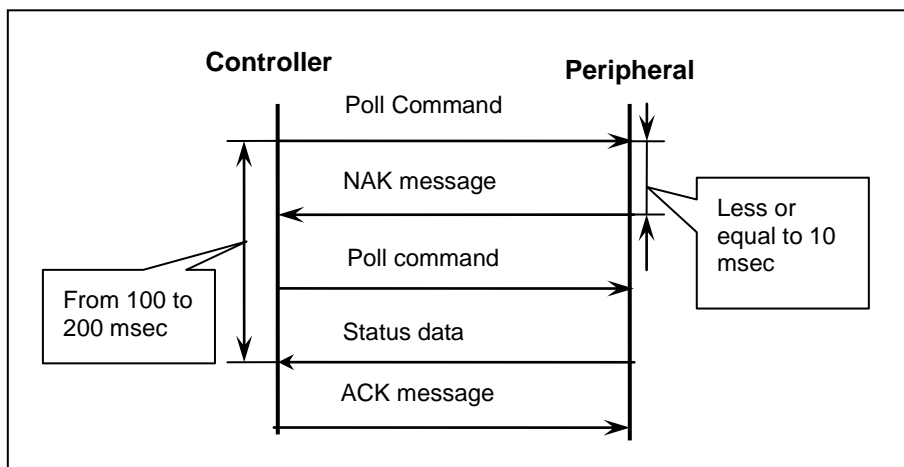
## 2.7 Timing Definitions

| Baud rate | The rate of bit transfer per second |
|---|---|
| $t_{inter-byte(max.)}$ | 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 |
| $t_{non-response\ (max.)}$ | The maximum non-response time |
| $T_{poll(min.)}$ | The minimum interval of time between two commands Poll |
| $T_{free(min.)}$ | The minimum 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 |
|---|---|
| $t_{inter-byte(max.)}$ | 5.0 msec |
| $t_{response(max.)}$ ** | 10.0 msec |
| $t_{bus\ reset(min.)}$ | 100 msec |
| $T_{non-response\ (max.)}$ | 5.0 S |
| $T_{poll(min.)}$ | 100msec (recommended 200 msec or more) |
| $T_{free(min.)}$ | 10 msec (recommended 20 msec or more) |

**

Bill-to-Bill may take as much as 200 msec to respond to **IDENTIFICATION** and **RECYCLING CASSETTE STATUS** commands. A 2 s timeout is required for **POWER RECOVERY**, **EXTENDED IDENTIFICATION** and **EXTENDED CASSETTE STATUS** commands.

# 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 other Bill Validators with Recycling capability 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.
Basic terminology used in the following subsections is as follows:
- "Bill type" – an entry in the bill table; bill types numbering starts with 0 and spans up to 23; not all the bill types may be active for any given Bill-to-Bill configuration (country dependant) and first bill is not mandated to have a bill type 0; bill table is not guaranteed to be continuous, empty entries may be present in between valid bill types;
- Bill type 24 – pseudo bill type for *escrow* cassette; does not represent any real bill type, but used to handle bills in Escrow Cassette instead;
- "Cassette number" – unique number of a recycling cassette; for 3-cassette configuration 1st cassette has number of 1, 2nd has number of 2 and so on; this sequence is not required to be continuous – i.e. if 2nd cassette is missing, there will be only cassettes with numbers 1 and 3 present;
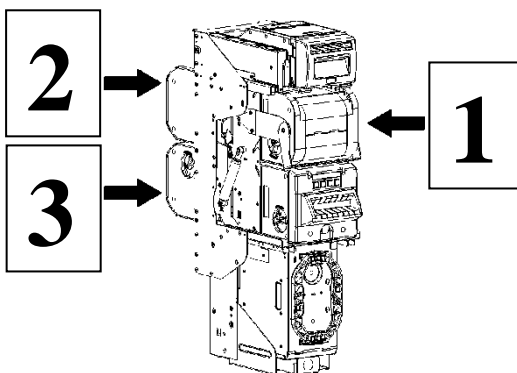


,
## 3.2 Command Protocol

If a Controller command is received by the Bill-to-Bill, which is not applicable to its current state, the Bill-to-Bill issues ILLEGAL COMMAND message. Other products may not support some commands. Please, refer to product manual or CashCode technical support for CCNET implementation details.

## 3.3 Controller Commands

| Command | HEX Code | Description | Applicable States |
|---|---|---|---|
| RESET | 30H | Command for Bill-to-Bill unit to self-reset | ALL |
| GET STATUS | 31H | Request for Bill-to-Bill unit set-up status | IDLING, DISABLED |
| SET SECURITY | 32H | Sets Bill-to-Bill unit Security Mode. Command is followed by set-up data. See command format | INITIALIZE, DISABLED |
| POLL | 33H | Request for Bill-to-Bill unit activity Status | ALL |
| ENABLE BILL TYPES | 34H | Indicates Bill Type enable or disable. Command is followed by set-up data. See command format | IDLING, ACCEPTING, REJECTING, ESCROW, HOLDING, STACKING, RETURNING, DISABLED |
| STACK | 35H | Sent by Controller to stack a bill in escrow to drop cassette or to one of the recycling cassettes | ESCROW, HOLDING |
| RETURN | 36H | Sent by Controller to return a bill in escrow | ESCROW, HOLDING |
| IDENTIFICATION | 37H | Request for Model, Serial Number, Country ISO code, Asset Number | POWER UP, INITIALISE, DISABLED, IDLING |
| HOLD | 38H | Command for holding a bill in Escrow state | ESCROW, HOLDING |

| Command | HEX Code | Description | Applicable States |
|---------|----------|-------------|-------------------|
| **SET BARCODE PARAMETERS** | *39H* | Command for settings the barcode format and number of characters | **INITIALIZE, DISABLED** |
| **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. | **ESCROW, PACKED, IDLING, DISABLED** |
| **RECYCLING CASSETTE STATUS** | *3BH* | Request for Bill-to-Bill unit recycling cassette status | **IDLING, DISABLED, ESCROW** |
| **DISPENSE** | *3CH* | Command to dispense bill(s) | **DISABLED** |
| **UNLOAD** | *3DH* | Command to unload bills from recycling cassette(s) to drop cassette | **DISABLED** |
| **EXTENDED IDENTIFICATION** | *3EH* | Request for Model, Serial Number, Software Version of Bill-to-Bill unit and its subunits, Country ISO code, Asset Number | **DISABLED, IDLING** |
| **SET RECYCLING CASSETTE TYPE** | *40H* | Assigns recycling cassettes to bill type | **DISABLED** |
| **GET BILL TABLE** | *41H* | Request for bill type description | **IDLING, DISABLED** |
| **GET CAPACITY LIMIT OF CASSETTE** | *43H* | Request for amount of bills which can load to correspond cassettes | **IDLING, DISABLED, ESCROW** |
| **SET CAPACITY LIMIT OF CASSETTE** | *44H* | Assigns amount of bills which can load to correspond cassettes | **DISABLED** |
| **DOWNLOAD** | *50H* | Command for transition to download mode. Please refer to CCNET Document 2 for details. | **DISABLED, FAILURE, DROP CASSETTE REMOVED** |
| **GET CRC32 OF THE CODE** | *51H* | Request for Bill Validator's firmware CRC32. | **POWER UP, INITIALIZE, DISABLED, FAILURE** |
| **MODULE DOWNLOAD** | *52H* | Command to enter an internal module update mode. Please refer to CCNET Document 2 for details. | **DISABLED** |
| **MODULE IDENTIFICATION REQUEST** | *53H* | Request serial numbers of all intelligent modules | **DISABLED, IDLING** |
| **VALIDATION MODULE IDENTIFICATION** | *54H* | Request identification information from banknote validation software module | **DISABLED, IDLING** |
| **GET CRC16 OF THE CODE** | *56H* | Request for Bill Validator's firmware CRC16. | **INITIALIZE, DISABLED, FAILURE** |
| **REQUEST STATISTICS** | *60H* | Command for retrieving full information about acceptance performance. Please refer to CCNET Document 3 for details. | **DISABLED** |
| **REQUEST/SET TIME&DATE** | *62H* | Read or initialize internal Real-Time Clock. | **DISABLED** |
| **POWER RECOVERY** | *66H* | Request whether there was a power cut and perform credit recovery | **DISABLED** |
| **EMPTY DISPENSER** | *67H* | Dispense all bills remaining in the dispenser after power cut. | **POWER CUT WHILE DISPENSING** |
| **SET OPTIONS** | *68H* | Set various Bill-To-Bill options | **INITIALIZE, DISABLED** |
| **GET OPTIONS** | *69H* | Set various Bill-To-Bill options | **INITIALIZE, DISABLED** |
| **EXTENDED CASSETTE STATUS** | *70H* | Extended recycling cassette status request | **DISABLED** |
| **DIAGNOSTIC/SETTING** | *F0H* | Device/Manufacturer specific command set. Refer to CCNET Document 7 for more details. | **DISABLED, FAILURE, DROP CASSETTE REMOVED** |

.

## 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 recycling 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. Low security levels provide higher acceptance and vice versa.

# POLL

| Controller Command | Code | Bill-to-Bill unit Response Data |
|---|---|---|
| **POLL** | 33H | *n* data bytes: **Z1- Z*n*** |

Indicates state of the Bill-to-Bill unit and its activity. The Bill-to-Bill unit sends 3 bytes of data in most cases (unless stated otherwise), but the package length should be determined according to the length of the frame (refer to the 2.2 for the message format).

The following data can be received from Bill-To-Bill unit in response to the POOL command:

| Z1 | Z2 | Z3 … Zn | Description |
|---|---|---|---|
| 10H | N/A | N/A | *POWER UP* – The state of the Bill-to-Bill after a power up. |
| 11H | N/A | N/A | *POWER UP with Bill in Validator* - Power up with bill in the Bill Validator head/upper chassis channel. The bill(s) will be returned back to the bezel during initialization phase. |
| 12H | N/A | N/A | *POWER UP with bill in Chassis* – Power up with bill in switch area, stacker area or in dispenser. The bill will be either returned to the bezel or stacked/dispensed with the credit message reported if necessary. |
| 13H | N/A | N/A | *INITIALIZE* – The state in which the Bill-to-Bill unit initializes itself after a RESET command from the Controller. |
| 14H | N/A | N/A | *IDLING* – The state in which the Bill-to-Bill is ready accept bills. |
| 15H | N/A | N/A | *ACCEPTING* – In this state the Bill-to-Bill unit continues to validate a bill and determine its denomination. |
| 17H | N/A | N/A | *STACKING* – In this state, the Bill-to-Bill unit transports a bill from Escrow position to the recycling cassette or to the drop cassette and remains in this state until the bill is stacked or returned if jammed. |
| 18H | N/A | N/A | *RETURNING* – In this state the Bill-to-Bill unit transports a bill from Escrow position to front bezel and remains in this state until the bill is removed by customer or returned if jammed. |
| 19H | N/A | N/A | *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 | N/A | N/A | *HOLDING* – The state, in which the bill is held in Escrow position after the HOLD command from the Controller. |
| 1BH | N/A | N/A | *BUSY* - The state in which the Bill-to-Bill unit is unable to act on any command. |
| 1CH | 60H | N/A | *REJECTING* - Rejecting due to Insertion. Insertion error |
|  | 61H | N/A | *REJECTING* - Rejecting due to Magnetic. Magnetic error |
|  | 62H | N/A | *REJECTING* - Rejecting due to bill Remaining in the head. Bill remains in the head, and new bill is rejected. |
|  | 63H | N/A | *REJECTING* - Rejecting due to Multiplying. Compensation error/multiplying factor error. |
|  | 64H | N/A | *REJECTING* - Rejecting due to Conveying. Conveying error. |
|  | 65H | N/A | *REJECTING* - Rejecting due to Identification1. Identification error. |
|  | 66H | N/A | *REJECTING* - Rejecting due to Verification. Verification error. |
|  | 67H | N/A | *REJECTING* - Rejecting due to Optic. Optic error. |
|  | 68H | N/A | *REJECTING* - Rejecting due to Inhibit. Returning by inhibit denomination error. |
|  | 69H | N/A | *REJECTING* - Rejecting due to Capacity. Capacitance error. |
|  | 6AH | N/A | *REJECTING* - Rejecting due to Operation. Operation error. |
|  | 6CH | N/A | *REJECTING* - Rejecting due to Length. Length error. |
|  | 6DH | N/A | *REJECTING* - Rejecting due to UV optic. Banknote UV properties do not meet the predefined criteria. |
|  | 92H | N/A | *REJECTING* - Rejecting due to unrecognised barcode. Bill taken was treated as a barcode but no reliable data can be read from it. |
|  | 93H | N/A | *REJECTING* - Rejecting due to incorrect number of characters in barcode. Barcode data was read (at list partially) but is inconsistent. |
|  | 94H | N/A | *REJECTING* - Rejecting due to unknown barcode start sequence. Barcode was not read as no synchronization was established. |
|  | 95H | N/A | *REJECTING* - Rejecting due to unknown barcode stop sequence. Barcode was read but trailing data is corrupt. |

| Z1 | Z2 | Z3 … Zn | Description |
|---|---|---|---|
| 1DH | 00H | N/A | *DISPENSING* - The Bill-to-Bill moves the bill(s) from recycling cassette to dispenser. |
| | 01H | N/A | *DISPENSING* - The Bill-to-Bill unit transitions into this state when dispensed bills become accessible to customer.B2B remains in this state until customer takes the bill(s) from dispenser. |
| 1EH | 00H | N/A | *UNLOADING* – The Bill-to-Bill unit is moving the bill(s) from recycling cassette to drop cassette. |
| | 01H | N/A | *UNLOADING* – The Bill-to-Bill unit is moving the bill(s) from recycling cassette to drop cassette. Number of bills requested is more than the number of bills in the cassette. |
| 21H | N/A | N/A | *SETTING TYPE CASSETTE* – The unloading of the recycling cassette is carried out, and if it is necessary, reprogramming EEPROM. |
| 25H* | N/A | N/A | *DISPENSED* – Dispensing is completed. |
| 26H* | Number of Bills | N/A | *UNLOADED* – Unloading is completed. |
| 28H | N/A | N/A | *INVALID BILL NUMBER* – Required number of bills is incorrect. |
| 29H | N/A | N/A | *SET CASSETTE TYPE* – Setting recycling cassette type is completed. |
| 30H | N/A | N/A | *INVALID COMMAND* – Command from the Controller is not valid. |
| 41H | N/A | N/A | *DROP CASSETTE FULL* – Drop Cassette full condition. |
| 42H | N/A | N/A | *DROP CASSETTE REMOVED* – The Bill-to-Bill unit has detected the drop cassette to be open or removed. |
| 43H | N/A | N/A | *JAM IN ACCEPTOR* – A bill has jammed in the bill path. |
| 44H | N/A | N/A | *JAM IN STACKER* – · A bill has jammed in drop cassette. |
| 45H* | N/A | N/A | *CHEATED* – The Bill-to-Bill unit detected attempts by to user to cheat. |
| 47H | Code1 | N/A | Generic Bill-to-Bill ERROR codes. Followed by failure description bytes. |
| 80H | Bill Type | N/A | *ESCROW*. |
| 81H* | Bill Type | **1 byte destination:** **0 –** drop cassette **1… 16 –** cassette 1…16 correspondingly | *PACKED, STACKED*. |
| 82H* | Bill Type | N/A | *RETURNED*. |
| 83H | 01H | N/A | *WAITING OF DECISION* (01h - the Bill-to-Bill is waiting till Controller enable or disable a determinate function). |

- **N/A** Not applicable to the current version (always = 00h). Reserved for future use.
- **Bill Type** – 0…24.
- **Code1** see Document 6 for the Bill-to-Bill or refer to product manual for particular error codes.

**\*** - State will be reported until it's read by Controller (a valid ACK received by Bill-To-Bill unit).

# 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.**

Escrow mode cannot be disabled for barcode coupons.

# STACK

| Controller Command | Code | Controller Data |
|---|---|---|
| **STACK** | 35H | No data bytes |

This command causes the Bill-to-Bill unit to send the "bill in escrow" position 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.

# RETURN

| Controller Command | Code | Controller Data |
|---|---|---|
| **RETURN** | 36H | No data bytes |

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

# 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-to-Bill, binary data |

Bytes Z1-Z27 must be sent as ASCII Characters. Zero (30H) and Blank (20H) is acceptable. Asset Number must be sent as binary code.
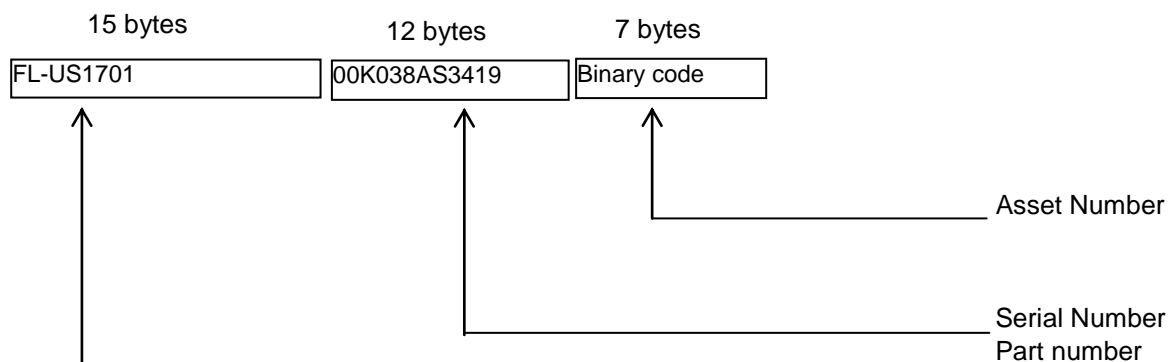
| 15 bytes | 12 bytes | 7 bytes |
|----------|----------|---------|
| FL-US1701 | 00K038AS3419 | Binary code |

Asset Number

Serial Number

Part number

# 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. To continue holding in Escrow state it is necessary to resend this command. Otherwise Bill-to-Bill unit 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-To-Bill 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-to-Bill 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.

**NOTE**：To ensure proper execution of Power Recovery routine Barcode data must be extracted when banknote is in Escrow position (event 80H).

# RECYCLING CASSETTE STATUS

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

**Z1-Z32**  Recycling Cassettes Status – up to 32 bytes.
The data returned in response to this command is a one-dimensional array of word (2-byte) entries. The number of the entries equals number of supported cassettes.

Every entry has the following format:
- ***most significant byte***:
    - bit 7 – cassette presence flag, set to 1 if cassette present;
    - bit 6 – cassette full flag;
    - bit 5 – reserved (not used);
    - bits 0 thru 4 – bill type, may take on of the following values:
        - 0 thru 23 – cassette assigned to the bill type *0* thru *23*;
        - 24 – cassette assigned to *escrow*;
        - 1FH – cassette is not assigned;
- ***least significant byte*** – number of bills in cassette.

If the cassette presence flag is not set all the other information in the entry has no meaning.

**NOTE:**  The Recycling Cassette can be programmed for **ESCROW** operation (bill type *24*). It means that all accepted bills not having a Recycling Cassette assigned to them would be directed from the Validating Head into this Recycling Cassette assigned to Escrow type. The option is useful when one wishes to return the money (in case of transaction cancellation) with exactly the same bills the customer inserted. If the customer cancels the transaction all his bills will be returned through the Dispensing Cassette. If transaction completes successfully all bills from the Escrow Recycling cassette should be reloaded into the Drop Cassette. It is the Host Controller's responsibility to perform this operation during the pause before next customer. If the next customer enters the bill the reload process should be interrupted and the Bill-to-Bill unit will serve the customer. The reload process should be continued at the next pause.
With Escrow cassette present in the Bill-to-Bill the host controller should maintain the log of bill insertions done by the customer. Otherwise in case of transaction cancellation the customer may get the money back with bills different from that one he inserted. This may happen if the customer inserted not an optimal set of bills for the amount given and the Bill-to-Bill has bills to give out an optimal combination for the amount given.

# DISPENSE

| Controller Command | Code | Controller Data |
|---|---|---|
| DISPENSE | 3CH | 32 bytes: **Y1-Y32** |

**Y1-Y32**  The odd bytes specify bill types and the even bytes contain number of bills to dispense for these types. For example, if bytes Y1-Y6 contain sequence 0, 2, 2, 1, 4, 3 it means that it is necessary to dispense 2 bills of type 0, 1 bill of type 2 and 3 bills of type 4.

**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**  The number of recycling cassette.
**Y2**  The number of bills.

# EXTENDED IDENTIFICATION

| Controller Command | Code | Bill-to-Bill unit Response Data |
|---|---|---|
| **EXTENDED IDENTIFICATION** | 3EH | 109 bytes: **Z1 – Z109** |

| Bytes | Description |
|---|---|
| **Z1-Z15** | Part Number – 15 bytes, ASCII characters |
| **Z16-Z27** | Serial Number – 12 bytes Factory assigned serial number, ASCII characters |
| **Z28-Z35** | Asset Number – 8 bytes, unique to every Bill-to-Bill, binary data |
| **Z36-Z41** | Boot Version Validator Head – 6 bytes, ASCII characters |
| **Z42-Z61** | Program Version Validator Head – 20 bytes, ASCII characters |
| **Z62-Z67** | Boot Version CPU Board – 6 bytes, ASCII characters |
| **Z68-Z73** | Program Version CPU Board – 6 bytes, ASCII characters |
| **Z74-Z79** | Boot Version Packer Board – 6 bytes, ASCII characters |
| **Z80-Z85** | Program Version Packer Board – 6 bytes, ASCII characters |
| **Z86-Z91** | Boot Version Cassette Processor1 – 6 bytes, ASCII characters |
| **Z92-Z97** | Boot Version Cassette Processor2 – 6 bytes, ASCII characters |
| **Z98-Z103** | Boot Version Cassette Processor3 – 6 bytes, ASCII characters |
| **Z104-Z109** | Program Version Cassette Processor – 6 bytes, ASCII characters |

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

Asset Number must be sent as binary code.

Example of Validator Head soft version:

6 bytes  20 bytes

| DL0317 | FL-DC7AAJ1-1.16-BB12 |
|---|---|

————— Program Version

————— Boot Version

15 bytes  12 bytes  8 bytes

| BB-US1701 | 00K038AS3419 | Binary code |
|---|---|---|

————— Asset Number

————— Serial Number

————— Part Number
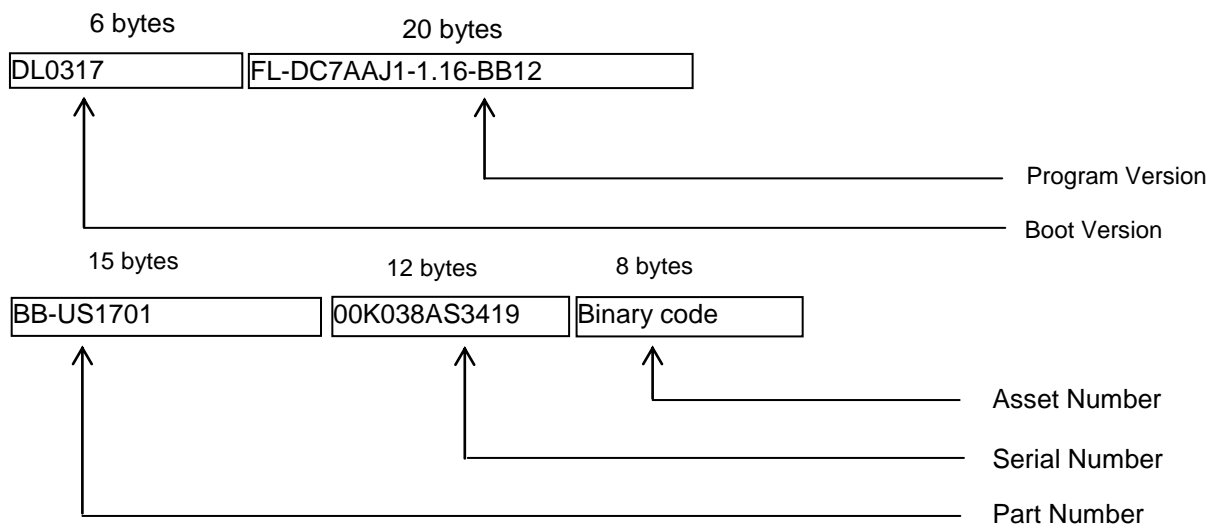
Example of Bill-to-Bill Devices software version:

6 bytes  6 bytes

| DL0703 | 203-32 |
|---|---|

————— Program Version

————— Boot Version

# 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 carries out complete unloading of the cassette.

**Y1**  Recycling cassette number.
**Y2**  Cassette assignment byte:
- Bits 0…4 **-** Bill type to assign for the cassette (1FH to make cassette "not   assigned", 24 to set cassette to Escrow type);
- Bits 5, 6, 7 – not used;

NOTE: if a cassette assign operation fails the cassette will be automatically made "not assigned"; some examples of possible fail causes are: cassette is not physically present, cassette is assigned to a bill type not supported by that specific software version (does not have a valid entry in the bill table) etc.

# GET BILL TABLE

| Controller Command | Code | Bill-to-Bill 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  – most significant digit(s) 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 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 | 1 | USA | 0x00 | 1 | Dollar |
| 1 | 5 | USA | 0x00 | 5 | Dollar |
| 2 | 1 | USA | 0x01 | 10 | Dollar |
| 3 | 2 | USA | 0x01 | 20 | Dollar |

Unsent bytes are assumed to be zero.

# GET CAPACITY LIMIT OF CASSETTE

| Controller Command | Code | Bill-to-Bill Response Data |
|---|---|---|
| **GET CAPACITY LIMIT OF CASSETTE** | 43H | up to 16 bytes**: Z1 up to Z16** |

**Z1-Z16**  Cassettes Level Status – up to 16 bytes.

The data returned in response to this command is a sequence of bytes. The number of byte equals number of supported cassettes.

Content of the byte reflects feeding level cassette (maximum amount of bills which can be loaded into cassette). If content of the byte is 0xFF - this feature is disabled.

# SET CAPACITY LIMIT OF CASSETTE

| Controller Command | Code | Controller Data |
|---|---|---|
| **SET CAPACITY LIMIT OF CASSETTE** | 44H | 2 bytes: **Y1-Y2** |

Command to assign the maximum capacity limit of a recycling cassette.
If the cassette is not empty and current number of bills in the Cassette exceed or equal new maximum level, Full Cassette flag will be set.

**Y1**        Recycling cassette number.
**Y2**        Maximum capacity limit:

Content of the **Y2** will reflect the capacity limit of corresponding cassette (maximum amount of bills which can be loaded into cassette). If **Y2** = 0xFF feature of feeding level will be disabled. If **Y2** is more than maximum actual cassette capacity (manufacturer maximum level) then maximum actual cassette capacity will be set.

# GET CRC32 OF THE CODE

| Controller Command | Code | Bill-To-Bill 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.

# MODULE IDENTIFICATION REQUEST

| Controller Command | Code | Bill-to-Bill Response Data |
|---|---|---|
| **IDENTIFICATION** | 53H | 60 bytes:    **Z1 – Z60** |

| Bytes | Description |
|---|---|
| **Z1-Z12** | Chassis' Serial Number – 12 bytes Factory assigned serial number, ASCII characters |
| **Z13-Z24** | Dispenser's Serial Number – 12 bytes Factory assigned serial number, ASCII characters |
| **Z24-Z36** | Cassette 1 Serial Number – 12 bytes Factory assigned serial number, ASCII characters |
| **Z37-Z48** | Cassette 2 Serial Number – 12 bytes Factory assigned serial number, ASCII characters |
| **Z49-Z60** | Cassette 3 Serial Number – 12 bytes Factory assigned serial number, ASCII characters |

All bytes must be sent as ASCII Characters. Zero (30H) and Blank (20H) are acceptable.

# VALIDATION MODULE IDENTIFICATION

| Controller Command | Code | Bill-to-Bill Response Data |
|---|---|---|
| **IDENTIFICATION** | 54H | 19 bytes:    **Z1 – Z19** |

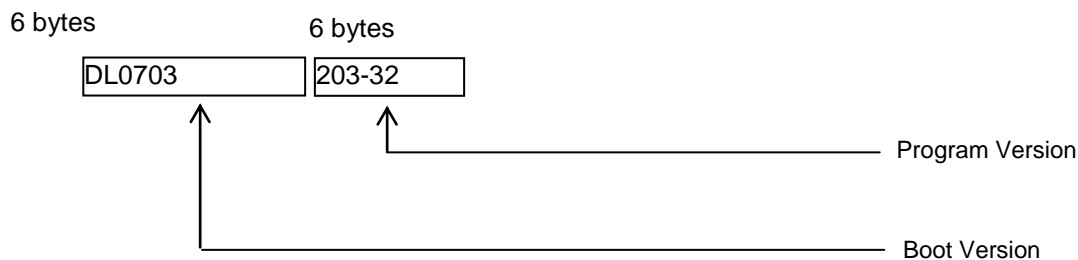| Bytes | Description |
|---|---|
| **Z1-Z15** | Part Number – 15 bytes, ASCII characters |
| **Z16-Z19** | CRC32 – 4 bytes, CRC32 of the validation software module, binary data |

Bytes Z1-Z15 must be sent as ASCII Characters. Zero (30H) and Blank (20H) are acceptable. CRC32 must be sent as binary code.

# GET CRC16 OF THE CODE

| Controller Command | Code | Controller Data | Bill-To-Bill Response Data |
|---|---|---|---|
| GET CRC16 OF THE CODE | 56H | 2 bytes: **Y1-Y2** | 2 bytes: **Z1-Z2** |

**Y1-Y2** - 2 bytes of SEED, MSB first.
**Z1-Z2** - 2 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.

# REQUEST/SET TIME&DATE

| Controller Command | Code | Controller Data | Bill-to-Bill Response Data |
|---|---|---|---|
| REQUEST TIME&DATE | 62H | - | 8 bytes: Z1 – Z8 |

| Controller Command | Code | Controller Data | Bill-to-Bill Response Data |
|---|---|---|---|
| SET TIME&DATE | 62H | 7 bytes: **Y1 – Y7** | 8 bytes: **Z1 – Z8** |

This command is used both for request and for setting time and date. If command contains controller data it is setting command otherwise it is request command. The same format data is used in these variants of a command. It corresponds to the following table. All data is presented in BCD format.

| Bit<br><br>Byte | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |
|---|---|---|---|---|---|---|---|---|---|
| 1. | CH | 10 Seconds | | | Seconds | | | | 00-59 |
| 2. | 0 | 10 Minutes | | | Minutes | | | | 00-59 |
| 3. | 0 | 0 | 10HR | | Hours | | | | 00-23 |
| 4. | 0 | 0 | 0 | 0 | 0 | Day | | | 1-7 |
| 5. | 0 | 0 | 10 date | | Date | | | | 01-28/29<br>01-30<br>01-31 |
| 6. | 0 | 0 | 0 | 10 month | Month | | | | 01-12 |
| 7. | 10 Year | | | | Year | | | | 00-99 |

**CH** – enable clock's oscillator. If this bit is cleared the clock is stopped.

# POWER RECOVERY

| Controller Command | Code | Bill-To-Bill response data |
|---|---|---|
| POWER RECOVERY | 66H | *n* data bytes: Z1-Z*n* |

This command is used to request the unit's power up status. In response to this command the Bill-To-Bill unit will report whether power was lost in the middle of a critical operation (such as packing, dispensing or unloading) and provide information for power recovery. The command should be sent by Host controller after power is restored (POWER UP state detected) and the unit finished initialization and is in DISABLED state. The unit's response could be 1 or more bytes long depending on the unit's power up status.

| Z1 (Power Up status) | Z2…Zn | Description |
|---|---|---|
| 00H | N/A | ***ACK (NO POWER CUT DETECTED)*** – all critical operations were successfully finished before power was turned OFF. |
| 17H | **2 or more bytes\* describing just packed bills** | ***POWER CUT WHILE PACKING*** - a power cut happened during the Accepting process and some bills remained in the channel when the power was restored. All bills have been packed and bytes ***Z2-Zn*** of the response contain list of packed bills. |
| 3CH | **Up to 80 bytes\*\* describing current dispenser state** | ***POWER CUT WHILE DISPENSING*** - a power cut happened during the Dispensing process and some bills remained in the dispenser. The last command and list of the bills is reported in data bytes ***Z2...Zn.*** EMPTY DISPENSER command should be issued and all bills successfully dispensed before enabling the Bill-To-Bill unit using BILL TYPE command. Otherwise the unit will report a FAILURE state. |
| 3DH | **Up to 50 bytes\*\*\* describing current dispenser state** | ***POWER CUT WHILE UNLOADING*** – a power cut happened during the Unloading process and some bills remain in the channel. The Bill-To-Bill unit has unloaded bills remaining in the channel into the drop box. The last command and list of the bills is reported in data bytes ***Z2...Zn.*** |

**N/A** – no additional bytes will be sent.


**\***- In the case of power cut while the Bill-To-Bill unit was packing a bill (either packing process had been started and bill still was in the channel or ACK to the STACKED message was not received) the Bill-To-Bill unit will pack all the bills remaining in the channel during initialization process. If the POWER RECOVERY command is sent after the initialization process the Bill-To-Bill unit will report POWER CUT WHILE PACKING state followed by list of all bills it has packed:

**Z2-Zn**     List of STACKED messages for all bills remaining in the channel upon power up. Each message is represented by 2-byte word of the following format:
- ***most significant byte*** – bill type of the bill;
- ***least significant byte*** – location where the bill was packed to:
  - **0 –** drop cassette;
  - **1… 16 –** cassette 1…16 correspondingly.


**\*\*** - In the case of power cut during the dispensing process, the Bill-To-Bill unit will likely have some bills in the dispenser when power is restored. In this case the unit will report POWER CUT WHILE DISPENSING state followed by up to 80 bytes describing the interrupted command and bills currently remaining in dispenser.

**Z2-Zn**    Operation Status Data – up to 80 bytes.
The operation status data is an array of word (2-byte) entries representing the command interrupted by power cut (up to 32 bytes) followed by an array of word (2-byte) entries representing numbers and bill types of the bills actually processed before power cut (up to 48 bytes). Note that those bills can still remain in dispenser and EMPTY DISPENSER command is required. The number of the entries equals the number of different denominations.
Every entry has the following format:
- ***most significant byte***:
- bit 7 – a bit indicating whether the value describes controller's request. If the bit is set, the word contains information supplied with the request (actual command data), otherwise it contains unit's power up state (data about the bills actually left recycling cassette before power cut).
- bits 5 … 6 – reserved (not used);
- bits 0 … 4 – bill type, may take on of the following values:
  - ***0…23*** – the least significant byte contains number of the bill type ***0…23*** correspondingly;
- ***least significant byte*** – number of the bills of a bill type specified by the most significant byte.

For example, if after power is restored the dispenser contains 5 bills of bill type 0, 3 bills of bill type 3 and 10 bills from the escrow cassette (3 of bill type 1 and 7 of bill type 5), and there was command to dispense 5 bills of bill type 0, 3 bills of bill type 3 and 12 bills from the escrow cassette the array will contain the following 2-byte entries:
      ***8005H, 8303H, 980CH, 0005H, 0103H, 0303H, 0507H***
The first 3 entries represent the command interrupted by the power cut, the last 4 – actual unit's power up state.

\*\*\* - If the unloading process was interrupted by power cut the unit will respond POWER CUT WHILE UNLOADIN
state followed by up to 50 bytes describing the interrupted command and bills actually unloaded.

**Z2-Zn**      Operation Status Data – up to 50 bytes. The operation status data is a word (2-byte) entry representing the command interrupted by power cut followed by an array of word (2-byte) entries representing numbers and bill types of the bills successfully unloaded into the drop box:

     **Z2-Z3**    **Original command data (2 bytes):**
       **Z2**    **-** Recycling cassette position (number)
       **Z3**    **-** Number of bytes to unload
     **Z4-Zn**    **Actual unit state (up to 48 bytes)**
     The actual unit's state is an array of word (2-byte) entries representing numbers and bill types of the bills successfully unloaded into the drop box. The number of the entries equals the number of different denominations.
     Every entry has the following format:
-   *most significant byte*:
-   bits 5 … 7 – reserved (not used);
-   bits 0 … 4 – bill type, may take on of the following values:
  -   *0…23* – the least significant byte contains number of the bill type *0…23* correspondingly;
-   *least significant byte* – number of the bills of a bill type specified by the most significant byte.

     For example, if power cut happened during unloading 12 bills from the escrow cassette (cassette 2), which contained 7 bills of bill type 5 and 5 bills of bill type 3, and only 10 bills (7 bills of bill type 5 and 3 bills of bill type 3) were successfully unloaded, the Operation Status Data will be:

# EMPTY DISPENSER

| Controller Command | Code | Controller Data |
|---|---|---|
| EMPTY DISPENSER | 67H | No data bytes |

This command is used to recover after power cut during dispensing process and tells the Bill-to-Bill unit to dispense all bills remaining in the dispenser at the moment. The command is valid only in DISABLED state and should be sent before enabling Bill-To-Bill unit using BILL TYPE command in the case if the unit reported POWER CUT WHILE DISPENSING in response to POWER RECOVERY command. Bill-To-Bill unit will react to this command in the same way as to DISPENSE command. Except no additional bills will be transported to the dispenser and dispenser will eject only its current content.

# SET OPTIONS

| Controller Command | Code | Controller Data |
|---|---|---|
| SET OPTIONS | 68H | 4 Bytes: **Y1 – Y4** |

The command turns ON/OFF different options implemented in the Bill-To-Bill device. Each option represents specific feature or optional behaviour supported by the device. The options are set using 4-byte bitmap with bits turning corresponding option ON/OFF. Reserved bits must be always set to 0.

| Byte Y1 bits | | | | | | | | Byte Y2 bits | | | | | | | | Byte Y3 bits | | | | | | | | Byte Y4 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 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Option is enabled if the corresponding bit is set | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| LED OFF | Hold bill | Check for tape | Turn switch | Timeout control bit 1 | Timeout control bit 0 | Disp. retracting bit1 | Disp. retracting bit0 | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved |
| Default settings on power up | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The following options are supported by Bill-To-Bill device:

- **LED OFF** – sets the state of the face lights in the DISABLED state:
  - $0$ – Red lights are turned ON;
  - $1$ – Face lights are OFF.
- **Hold bill** – defines whether or not hold bill on the input roller after ejecting:
  - $0$ – Do not hold the bill after ejecting;
  - $1$ – Hold the bill after ejecting until customer takes it out.
- **Check for tape** – defines whether or not apply advanced tape detection algorithm. The advanced algorithm better detects tapes attached to devices, but significantly increases the bill processing time.
- **Turn switch** - defines whether or not to secure bill path after packing the bill. Use this option to prevent cheating by pulling out the bill after stacking. This feature increases time required to stack the bill.
- **Timeout control –** a two-bit field used to control CCNET communication timeout. If no POLL command is received within the timeout, all bills are disabled and, if the device is in IDLING state, it is forced into DISABLED state. The bit-field is mapped using the following table:

| Bit 0 | Bit 1 | Timeout |
|---|---|---|
| 0 | 0 | Indefinite |
| 1 | 0 | 1 second |
| 0 | 1 | 2 seconds |
| 1 | 1 | 5 seconds |

- Dispensers retracting control - a two-bit field used to control bills retracting back to dispenser if customer forget take of the bills. If this feature enabled Bill-to-Bill retract bills after defined time interval and go to error mode ( CAh ). The bit-field is mapped using the following table:

| Bit 0 | Bit 1 | Timeout |
|---|---|---|
| 0 | 0 | Feature is disabled |
| 1 | 0 | Low time interval |
| 0 | 1 | Middle time interval |
| 1 | 1 | Big time interval |

Device BB XE300 has presetted values for this feature: low time interval – 20s., middle – 40s. and big time interval is 60s..

# GET OPTIONS

| Controller Command | Code | Bill-To-Bill response Data |
|---|---|---|
| GET OPTIONS | 69H | 4 Bytes: **Y1 – Y4** |

The command requests currently enabled options implemented in the Bill-To-Bill device. The options are retuned in the form of 4-byte bitmap with bits corresponding to specific options. The bitmap format is described in **SET OPTIONS** command description.

# EXTENDED CASSETTE STATUS

| Controller Command | Code | Controller Data | Bill-To-Bill response Data |
|---|---|---|---|
| EXTENDED CASSETTE STATUS | 70H | One byte **X1** | Up to 129 Bytes: **Z1 – Z129** |

The command requests extended information about the contents of the recycling cassette. In response to the command the Bill-To-Bill unit will report total number of the bills in the cassette as well as bill type for each bill. This command is useful if one needs to know exactly which bills are stored in the escrow cassette. Note, that Bill-To-Bill unit stores bills in the recycling cassettes according to last-in-first-out principle. Therefore the response will reflect the actual image of the cassette contents (actual denominations and their positions in the cassette).

**Controller data:**

  **X1 –** Cassette number to report information about

**Bill-To-Bill unit response Data:**

  **Z1**   Total number of bills in the cassette. This value determines length of the following cassette contents array.

  **Z2-Zn**  Cassette Contents Data – Z1 bytes. The cassette contents data is an array containing bill types of the bills in the cassette.  If the recycling cassette is assigned to a denomination (not escrow cassette) the array will contain Z1 identical entries equal to the bill type associated with the cassette. In the case of escrow cassette the array will reflect the bills inside of the cassette.

# 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 |
| DIAGNOSTIC/SETTING | F0H | Device/Manufacturer specific command set. Refer to CCNET Document 7 for more details. |

## 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 acceptances until otherwise instructed by the Controller.

# GET STATUS

| Controller Command | Code | Coin Changer Response Data |
|---|---|---|
| GET STATUS | 09H | 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  –  most significant digit(s) 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. Some devices may not support the entire CCNET command set. Please, refer to product manual or CashCode technical support for CCNET implementation details.

## 5.3 Controller Commands

| Command | HEX Code | Description | Applicable States |
|---|---|---|---|
| RESET | 30H | Command for Bill Validator to self-reset | ALL |
| GET STATUS | 31H | Request for Bill Validator set-up status | ALL |
| SET SECURITY | 32H | Sets Bill Validator Security Mode. Command is followed by set-up data. See command format | ALL |
| POLL | 33H | Request for Bill Validator activity Status | ALL |
| ENABLE BILL TYPES | 34H | Indicates Bill Type enable or disable. Command is followed by set-up data. See command format | ALL |
| STACK | 35H | Sent by Controller to send a bill in escrow to the drop cassette | ESCROW, HOLDING |
| RETURN | 36H | Sent by Controller to return a bill in escrow | ESROW, HOLDING |
| IDENTIFICATION | 37H | Request for Software Part Number, Serial Number, Asset Number | POWER UP, INITIALIZE, DISABLED, FAILURE |
| HOLD | 38H | Command for holding of Bill Validator in Escrow state | ESROW, HOLDING |
| SET BARCODE PARAMETERS | 39H | Command for settings the barcode format and number of characters | ALL |
| 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. | ESCROW, PACKED, DISABLED, IDLING |
| GET BILL TABLE | 41H | Request for bill type description | POWER UP, INITIALIZE, DISABLED, FAILURE |
| DOWNLOAD | 50H | Command for transition to download mode. Please refer to CCNET Document 2 for details. | POWER UP, INITIALIZE, DISABLED, FAILURE |
| GET CRC32 OF THE CODE | 51H | Request for Bill Validator's firmware CRC32. | INITIALIZE, DISABLED, FAILURE |
| MODULE DOWNLOAD | 52H | Command to enter an internal module update mode. Please refer to CCNET Document 2 for details. | DISABLED |
| VALIDATION MODULE IDENTIFICATION | 54H | Request identification information from banknote validation software module | DISABLED, IDLING |
| GET CRC16 OF THE CODE | 56H | Request for Bill Validator's firmware CRC16. | INITIALIZE, DISABLED, FAILURE |
| GET CRC32 OF THE CODE WITH PARAMETRS | 57H | Request for Bill Validator's firmware CRC32. | POWER UP, INITIALIZE, DISABLED, FAILURE |
| GET CRC16 OF THE CODE WITH PARAMETRS | 58H | Request for Bill Validator's firmware CRC16. | POWER UP, INITIALIZE, DISABLED, FAILURE |

| Command | HEX Code | Description | Applicable States |
|---|---|---|---|
| **SET ESCROW TIME OUTS** | *5AH* | Set escrow time-outs | **IDLING, ACCEPTING, REJECTING, ESCROW, HOLDING, STACKING, RETURNING, DISABLED** |
| **GET COUNTER OF STACKED BILL** | *5BH* | Get counter of stacked bills | **IDLING, POWER UP, INITIALIZE, DISABLED, FAILURE** |
| **GET DATE OF THE RELEASE** | *5CH* | Get date of the firmware release | **IDLING, POWER UP, INITIALIZE, DISABLED, FAILURE** |
| **REQUEST STATISTICS** | *60H* | Command for retrieving full information about acceptance performance. Please refer to CCNET Document 3 for details. | **POWER UP, INITIALIZE, DISABLED, FAILURE** |
| **SET OPTIONS** | *68H* | Set various BV options | **INITIALIZE, DISABLED** |
| **GET OPTIONS** | *69H* | Set various BV options | **INITIALIZE, DISABLED** |
| | | | |
| **DIAGNOSTIC/ SETTING** | *F0H* | Device/Manufacturer specific command set. Refer to CCNET Document 7 for more details. | **DISABLED, FAILURE, DROP CASSETTE REMOVED** |

## 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 | Bill 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 |

\* If *BillTable48* option is set table size doubles up to 48 denominations, bar code moves to the end of table.
Response data are Y1- Y12.

# 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.

\* If *BillTable48* option is set table size doubled up to 48 denominations, bar code moves to the end of table.
Response data are Y1- Y6.

## 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 1 or 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 Conveying | Bill transport error |
| **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 |
| **1CH** | **92H** | Rejecting due to unrecognised barcode | Bill taken was treated as a barcode but no reliable data can be read from it. |
| **1CH** | **6DH** | Rejecting due to UV | Banknote UV properties do not meet the predefined criteria |
| **1CH** | **93H** | Rejecting due to incorrect number of characters in barcode | Barcode data was read (at list partially) but is inconsistent. |

| Response data bytes | | Description | |
|---|---|---|---|
| **Z1** | **Z2** | | |
| **1CH** | **94H** | Rejecting due to unknown barcode start sequence | Barcode was not read as no synchronization was established. |
| **1CH** | **95H** | Rejecting due to unknown barcode stop sequence | Barcode was read but trailing data is corrupt. |
| **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 first bill until the second bill is removed.* |
| **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 Chanel Failure | One of the optic sensors has failed to provide its response. |
| **47H** | **56H** | Magnetic Chanel Failure | Inductive sensor failed to respond |
| **47H** | **57H** | Failure of the OS | |
| **47H** | **58H** | Failure of the Memory Stick | |
| **47H** | **59H** | Failure of the DPLINK | |
| **47H** | **5AH** | Failure of interface I2C | |
| **47H** | **5BH** | Failure of interface SPI | |
| **47H** | **5CH** | Failure of interface 52 | Only for FLS - error exchange with Second CPU. |
| **47H** | **5FH** | Capacitance Canal Failure | Capacitance sensor failed to respond |
| | | Events with credit. | |
| **80H** | **YH** | Escrow position | $Y$ = bill type (0 to 23). If *BillTable48* option is set Y is (0 to 47), bar code moves to the end of table. 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). If *BillTable48* option is set Y is (0 to 47), bar code moves to the end of table. |
| **82H** | **YH** | Bill returned | $Y$ = bill type (0 to 23). If *BillTable48* option is set Y is (0 to 47), bar code moves to the end of table. |
| **90H** | **YH** | Counterfeit bill is held in BV | $Y$ =validation results |
| **84H** | **YH** | Suspected bill detected | $Y$ =validation results |

* Bill validator will be in "**Pause**" state for **10 sec**. After that the validator will try to reject both bills.
- **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.**

Escrow mode can not be disabled for barcode coupons.

\* If *BillTable48* option is set table size doubles up to 48 denominations, bar code moves to the end of table.
Response data are Y1- Y12.

# 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 Validator 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.
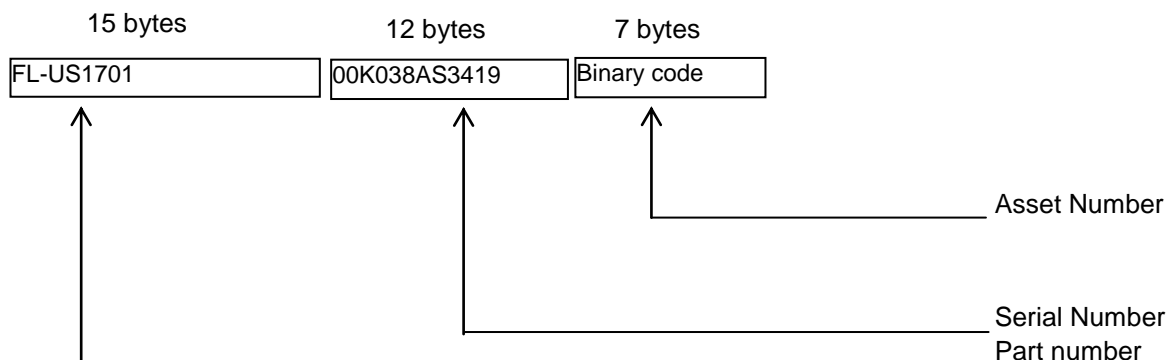


This command is valid in the following states only: Power up, Initialize, one of the Failure states (41H-47H) or Unit Disabled.

# 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.

**NOTE**：To ensure proper execution of Power Recovery routine Barcode data must be extracted when banknote is in Escrow position (event 80H).

# 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  – most significant digit(s) 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 | 1 | USA | 0x00 | 1 | Dollar |
| 1 | 5 | USA | 0x00 | 5 | Dollar |
| 2 | 1 | USA | 0x01 | 10 | Dollar |
| 3 | 2 | USA | 0x01 | 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.

\* If *BillTable48* option is set table size doubles up to 48 denominations, bar code moves to the end of table.
        Response data are Y1- Y240.

# 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.

# VALIDATION MODULE IDENTIFICATION

| Controller Command | Code | Bill Validator Response Data |
|---|---|---|
| IDENTIFICATION | 54H | 19 bytes: **Z1 – Z19** |

| Bytes | Description |
|---|---|
| Z1-Z15 | Part Number – 15 bytes, ASCII characters |
| Z16-Z19 | CRC32 – 4 bytes, CRC32 of the validation software module, binary data |

Bytes Z1-Z15 must be sent as ASCII Characters. Zero (30H) and Blank (20H) are acceptable. CRC32 must be sent as binary code.

# GET CRC16 OF THE CODE

| Controller Command | Code | Controller Data | Bill Validator Response Data |
|---|---|---|---|
| GET CRC16 OF THE CODE | 56H | 2 bytes: **Y1-Y2** | 2 bytes: **Z1-Z2** |

**Y1-Y2** - 2 bytes of Seed, MSB first.
**Z1-Z2** - 2 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.

# GET CRC32 OF THE CODE WITH SEED, START ADDRESS

| Controller Command | Code | Controller Data | Bill Validator Response Data |
|---|---|---|---|
| GET CRC32 OF THE CODE | 57H | 2 bytes: **Y1-Y8** | 4 bytes: **Z1-Z4** |

**Y1-Y4** - 4 bytes of Seed, MSB first.
**Y5-Y8** - 4 bytes of Start Address, MSB first.
**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.

# GET CRC16 OF THE CODE WITH SEED, START ADDRESS

| Controller Command | Code | Controller Data | Bill Validator Response Data |
|---|---|---|---|
| GET CRC16 OF THE CODE | 58H | 2 bytes: **Y1-Y6** | 2 bytes: **Z1-Z2** |

**Y1-Y2** - 2 bytes of Seed, MSB first.
**Y3-Y6** - 4 bytes of Start Address, MSB first.
**Z1-Z2** - 2 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.

# SET ESCROW TIME-OUTS

| Controller Command | Code | Controller Data |
|---|---|---|
| SET ESCROW TIME-OUTS | 5AH | 8 bytes: **Y1-Y8** |

**Y1-Y4** - 4 bytes of Bill time-out in msec, MSB first.
**Y5-Y8** - 4 bytes of Voucher time-out in msec, MSB first.

This command is valid in all states.

# GET COUNTER OF STACKED BILLS

| Controller Command | Code | Bill Validator Response Data |
|---|---|---|
| GET COUNTER OF STACKED BILLS | 5BH | 2 bytes: **Z1-Z2** |

**Y1-Y2** - 2 bytes of counter, MSB first.

This command is valid in all states.

# GET DATE OF THE RELEASE

| Controller Command | Code | Bill Validator Response Data |
|---|---|---|
| GET DATE OF THE RELEASE | 5CH | 8 bytes: **Z1-Z8** |

**Z1-Z8** - 8 bytes of Date of the firmware release, MSB first.

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

# SET OPTIONS

| Controller Command | Code | Controller Data |
|---|---|---|
| SET OPTIONS | 68H | 4 Bytes: **Y1 – Y4** |

The command turns ON/OFF different options implemented in the BV. Each option represents specific feature or optional behaviour supported by the device. The options are set using 4-byte bitmap with bits turning corresponding option ON/OFF. Reserved bits must be always set to 0.

| Byte Y1 bits | | | | | | | | Byte Y2 bits | | | | | | | | Byte Y3 bits | | | | | | | | Byte Y4 bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LED OFF | Hold bill | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Bill Table 48 | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Clear A6 Result | Enable Article 6 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Option is enabled if the corresponding bit is set | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Default settings on power up | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | **0** |

The following options are supported by the device:

- **LED OFF** – sets the state of the face lights in the DISABLED state:
  - **0** – Red lights are turned ON;
  - **1** – Face lights are OFF.
- **HOLD BILL** – defines whether or not hold bill on the input roller after ejecting:
  - **0** – Do not hold the bill after ejecting;
  - **1** – Hold the bill after ejecting until customer takes it out.


- *Byte Y4 bit* 0 –Enable/disable Article6 mode. Article6 mode is turn on if this bit is set.  Article6 mode is turns off if this bit is reset.  After BV reset this bit cleared by default for previous hosts software version support.
- *Byte Y4 bit* 1 –Clear counterfeit detection result. Counterfeit bill will be ejected from BV if this bit is set.  This bit will be cleared right after counterfeit bill leaves BV.
- *ByteY2 bit* 7 – Extends Bill Table up to 48 denominations. Commands **GET OPTIONS**, **SET OPTIONS**, **ENABLE BILL TYPES**, **SET SECURITY**, **GET STATUS**, **GET BILL TABLE** affected.


# GET OPTIONS

| Controller Command | Code | Bill Validator Response Data |
|---|---|---|
| **GET OPTIONS** | 69H | 4 Bytes: **Y1 – Y4** |

The command requests currently enabled options implemented in the BV. The options are retuned in the form of 4-byte bitmap with bits corresponding to specific options.

- *Byte Y4 bit* 0 –Enable/disable Article6 mode. Article6 mode is turn on if this bit is set.  Article6 mode is turns off if this bit is reset.
- *Byte Y4 bit* 1 –That bit should be always read as 0.

# 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 | 31H | 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. |
| SPECIAL COMMANDS | 61H | Special commands are Get/Set Time, Get Active Payment Systems List, Get Error Log, Get/Change Settings and Get/Modify Language Tables. Please refer to CCNET Document 4 for details. |

## 6.5    Controller Command Format

# RESET

| Controller Command | Code | Controller Data |
|---|---|---|
| RESET | 31H | 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-Y4 | Z1 – Z2 |

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

**Y2-Y4** =     Default Currency (3 bytes of ASCII ISO currency code, MSB first)
Indicates the default operating currency of the card reader. If a card is configured to store amounts in different currencies and the default currency in one of them this amount will be shown as card balance. If there is no matching currency in the card the balance to display is left to the discretion of the reader.

SETUP command is valid for Inactive and Disabled states only. If received with any other states active, a COMMAND INVALID response is issued.

          **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 |
| 14H | | Vend cancelled | Vend process successfully cancelled |
| 15H | | Vend cancel failed | Vend process may not be cancelled and should be completed |
| 16H | Z2 | Level Updated | Indicates the Reader has internally updated its capabilities level. Z2 contains new capabilities level |
| 17H | Z2 | Options Updated | Indicates the Reader has internally updated its options. Z2 contains new options byte where format is same as for options byte returned in response to setup command (see blow). |

**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 | none |

**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 an ACK message and will initiate cancellation sequence. Upon completion the result will be reflected in the reader status (returned in response to only one successful Poll command) and return to the Idling state. This command is valid only in Vending state, otherwise COMMAND INVALID response is issued.

## 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.

| 15 bytes | 12 bytes | 7 bytes |
|---|---|---|
| FL-US1701 | 00K038AS3419 | Binary code |

                                        Asset Number

                                        Serial Number
                                        Model
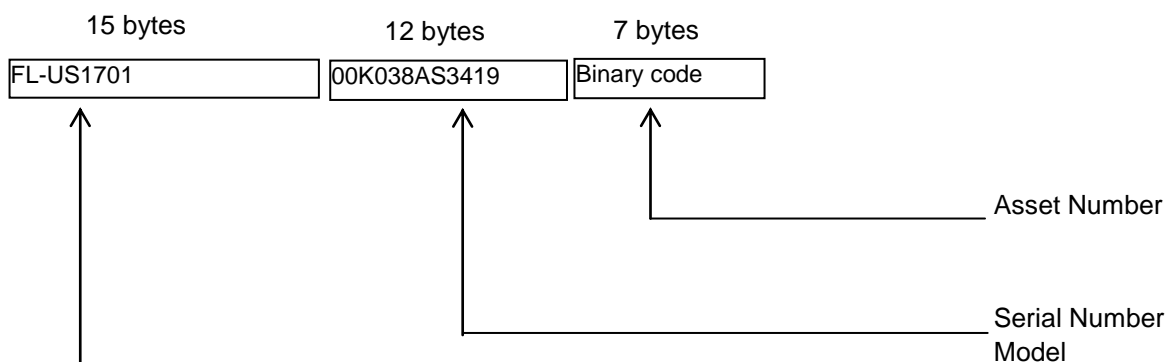
## DOWNLOAD

| 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

CCNET Message Sequences

Examples

(Bill-to-Bill unit)

## 7.1 Power Up & Reset sequence (No previous power cut)

| 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** | ------> | | |
| | | | |
| **SET OPTIONS*** | ------> | | Enabling/disabling optional features |
| | <------- | ACK | |
| | | | |
| **POLL** | ------> | | |
| | <------ | INITIALIZE | Bill-to-Bill unit is initializing |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | UNIT DISABLED | All bill types are disabled |
| **ACK** | ------> | | |
| | | | |
| **GET STATUS** | ------> | | Collect operational parameters |
| | <------ | BILL-TO-BILL UNIT CONFIG. | |
| **ACK** | ------> | | |
| | | | |
| **GET BILL TABLE** | ------> | | |
| | <------ | BILL TABLE | |
| **ACK** | ------> | | |
| | | | |
| **SET SECURITY** | ------> | | Update bill security levels |
| | <------ | ACK | |
| | | | |
| **IDENTIFICATION** | ------> | | Collect Information |
| | <------ | BILL-TO-BILL UNIT ID | |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | INITIALIZE | Bill-to-Bill unit is initializing |
| **ACK** | ------> | | |
| | | . | |
| | | . | |
| | | . | |
| **POLL** | ------> | | |
| | <------ | INITIALIZE | Bill-to-Bill unit is initializing |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | DISABLED | Bill-to-Bill unit completed initialization process and is ready for work |
| **ACK** | ------> | | |
| | | | |
| **POWER RECOVERY**** | ------> | | Report Power Up status |
| | <------ | NO POWER CUT DETECTED | |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | DISABLED | Bill-to-Bill unit completed initialization |

| | | | |
|---|---|---|---|
| | | | process and is ready for work |
| **ACK** | ------> | | |
| | | | |

\*- The SET OPTIONS command is optional and should be sent if it is planned to use feature not enabled by default.

\*\* - The POWER RECOVERY command is optional and can be skipped.

## 7.2 Enable sequence

| Controller | | Bill-to-Bill unit | Comments |
|---|---|---|---|
| **ENABLE BILL TYPES** | ------> | | Enable appropriate bill types |
| | <------ | ACK | |
| **POLL** | ------> | | |
| | <------ | IDLING | Ready to accept bills |
| **ACK** | ------> | | |

## 7.3 Disable sequence.

| Controller | | Bill-to-Bill unit | Comments |
|---|---|---|---|
| **ENABLE BILL TYPES** | ------> | | Disable all bill types |
| | <------ | ACK | |
| **POLL** | ------> | | |
| | <------ | DISABLED | Unit is not ready to accept bills |
| **ACK** | ------> | | |

## 7.4 Bill Accepting sequence (Bill stacked).

| Controller | | Bill-to-Bill unit | Comments |
|---|---|---|---|
| **POLL** | ------> | | |
| | <------ | ACCEPTING | Accepting bill |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | ESCROW POSITION | Bill in escrow position |
| **ACK** | ------> | | |
| | . | | |
| | . | | |
| | . | | |
| **POLL** | ------> | | |
| | <------ | ESCROW POSITION | Bill in escrow position |
| **ACK** | ------> | | |
| | | | |
| **STACK** | ------> | | Send bill to drop cassette or one of the recycling cassettes. |
| | <------ | ACK | |
| | | | |
| **POLL** | ------> | | |
| | <------ | STACKING | Stacking bill. |
| **ACK** | ------> | | |
| | . | | |
| | . | | |
| | . | | |
| **POLL** | ------> | | |
| | <------ | BILL STACKED | Bill has been stacked |
| **ACK** | ------> | | |
| | | | |
| **CASSETTE STATUS** | ------> | | Collect operational parameters |
| | <------ | BILL-TO-BILL UNIT CASSETTE STATUS | |
| **ACK** | ------> | | |
| | | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | IDLING | Ready to accept bills |
| | | | |
| **ACK** | ------> | | |

## 7.5 Bill Accepting sequence (Bill returned)

| Controller | | Bill-to-Bill unit | Comments |
|---|---|---|---|
| POLL | ------> | | |
| | <------ | ACCEPTING | Accepting bill |
| | | | |
| POLL | ------> | | |
| | <------ | ESCROW POSITION | Bill in escrow position |
| | | | |
| | . | | |
| | .. | | |
| POLL | ------> | | |
| | <------ | ESCROW POSITION | Bill in escrow position |
| ACK | ------> | | |
| | | | |
| RETURN | ------> | | Return bill to user |
| | <------ | ACK | |
| | | | |
| POLL | ------> | | |
| | <------ | RETURNING | Returning bill |
| ACK | ------> | | |
| | | | |
| | . | | |
| | .. | | |
| POLL | ------> | | |
| | <------ | BILL RETURNED | Bill has been returned |
| ACK | ------> | | |
| | | | |
| | | | |
| POLL | ------> | | |
| | <------ | IDLING | Ready to accept bills |
| | | | |
| ACK | ------> | | |

## 7.6 Bill Dispensing sequence.

| Controller | | Bill-to-Bill unit | Comments |
|---|---|---|---|
| **ENABLE BILL TYPES** | ------> | | Disable bill types |
| | <------ | ACK | |
| | | | |
| **DISPENSE** | ------> | | Dispense bills to customer |
| | <------ | ACK | |
| | | | |
| **POLL** | ------> | | |
| | <------ | BILL DISPENSING | Transporting bills to dispenser |
| **ACK** | ------> | | |
| | . | | |
| | . | | |
| | . | | |
| **POLL** | ------> | | |
| | <------ | BILL DISPENSED | Dispense is completed. Bills removed from the dispenser. |
| **ACK** | ------> | | |
| | | | |
| **CASSETTE STATUS** | ------> | | Collect operational parameters |
| | <------ | BILL-TO-BILL UNIT CASSETTE STATUS | |
| **ACK** | ------> | | |
| | | | |
| **ENABLE BILL TYPES** | ------> | | Enable bill types |
| | <------ | ACK | |
| | | | |
| **POLL** | ------> | | |
| | <------ | IDLING | Ready to accept bills |
| | | | |
| **ACK** | ------> | | |

## 7.7    Bill Unloading sequence.

| Controller | | Bill-to-Bill unit | Comments |
|---|---|---|---|
| **ENABLE BILL TYPES** | ------> | | Disable bill types |
| | <------ | ACK | |
| **UNLOAD** | ------> | | Unload bills from recycling cassette to drop cassette. |
| | <------ | ACK | |
| | | | |
| **POLL** | ------> | | |
| | <------ | BILL UNLOADING | Bill transporting to drop cassette. |
| **ACK** | ------> | | |
| | . . . | | |
| **POLL** | ------> | | |
| | <------ | BILL UNLOADED | Unloading is completed. |
| **ACK** | ------> | | |
| | | | |
| **CASSETTE STATUS** | ------> | | Collect operational parameters |
| | <------ | BILL-TO-BILL UNIT CASSETTE STATUS | |
| **ACK** | ------> | | |
| | | | |
| **ENABLE BILL TYPES** | ------> | | Enable bill types |
| | <------ | ACK | |
| **POLL** | ------> | | |
| | <------ | IDLING | Ready to accept bills |
| **ACK** | ------> | | |

## 7.8   Set cassette type sequence

| Controller | | Bill-to-Bill unit | Comments |
|---|---|---|---|
| **CASSETTE STATUS** | ------> | | Collect operational parameters |
| | <------ | BILL-TO-BILL UNIT CASSETTE STATUS | Cassette to be SET contains bills. |
| **ACK** | ------> | | |
| | | | |
| **SET CASSETTE TYPE** | ------> | | Assigning cassette to bill type |
| | <------ | ACK | |
| | | | |
| **POLL** | ------> | | |
| | <------ | SETTING CASSETTE TYPE | Unloading bills. |
| **ACK** | ------> | | |
| | . | | |
| | . | | |
| | . | | |
| **POLL** | ------> | | |
| | <------ | SET CASSETTE TYPE | Cassette is set to new bill type. |
| **ACK** | ------> | | |
| | | | |
| **CASSETTE STATUS** | ------> | | Collect operational parameters |
| | <------ | BILL-TO-BILL UNIT CASSETTE STATUS | Cassette is assigned new bill type. It contains no bills. |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | DISABLED | Unit is disabled. |
| **ACK** | ------> | | |

### 7.9 Power Up sequence after power cut during stacking (Power recovery command was sent)

| Controller | | Bill-to-Bill Unit | Comments |
|---|---|---|---|
| **POLL** | ------> | | |
| | <------ | POWER UP | Power |
| **ACK** | ------> | | |
| | | | |
| **RESET** | ------> | | Reset peripheral |
| | <------ | ACK | |
| | | | |
| **POLL** | ------> | | |
| | <------ | INITIALIZE | Bill Validator is initializing |
| **ACK** | ------> | | |
| | | | |
| **SET OPTIONS*** | ------> | | Enabling miscellaneous options |
| | <------- | ACK | |
| | | | |
| **GET STATUS** | ------> | | Collect operational parameters |
| | <------ | BILL VALIDATOR CONFIG. | |
| **ACK** | ------> | | |
| | | | |
| **SET SECURITY** | ------> | | Update bill security levels. |
| | <------ | ACK | |
| | | | |
| **IDENTIFICATION** | ------> | | Collect asset information. |
| | <------ | BILL VALIDATOR ID | |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | INITIALIZE | Bill-to-Bill unit is initializing |
| **ACK** | ------> | | |
| | | . | |
| | | . | |
| | | . | |
| **POLL** | ------> | | |
| | <------ | INITIALIZE | Bill-to-Bill unit is initializing |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | DISABLED | Bill-to-Bill unit completed initialization process and is ready for work |
| **ACK** | ------> | | |
| | | | |
| **POWER RECOVERY**** | ------> | | Report Power Up status and credit information |
| | <------ | POWER CUT WHILE PACKING | |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | UNIT DISABLED | All bill types are disabled. |
| **ACK** | ------> | | |

*- The SET OPTIONS command is optional and should be sent if it is planned to use feature not enabled by default.
** - On 'Power up' the validator will check if there was a bill in process just before the power down. A bill is considered to be in process if:
  1. It was in the stacking process (bill successfully recognized and verified and decision to stack the bill had been made).
  2. It was stacked but the event was not reported or a response to stacked message was not received.
The bill (or several bills), which were in process will be packed and credits will be reported for each bill in the response to POWER RECOVERY command. The POWER RECOVERY command is optional and can be skipped. If the Bill-To-Bill unit is enabled by BILL TYPE command before issuing POWER RECOVERY command the credits will be lost.

### 7.10 Power Up sequence after power cut during dispensing (Power recovery command was sent)

| Controller | | Bill-to-Bill Unit | Comments |
|---|---|---|---|
| **POLL** | ------> | | |
| | <------ | POWER UP | Power |
| **ACK** | ------> | | |
| | | | |
| **RESET** | ------> | | Reset peripheral |
| | <------ | ACK | |
| | | | |
| **POLL** | ------> | | |
| | <------ | INITIALIZE | Bill Validator is initializing |
| **ACK** | ------> | | |
| | | | |
| **SET OPTIONS*** | ------> | | Enabling miscellaneous options |
| | <------ | ACK | |
| | | | |
| **GET STATUS** | ------> | | Collect operational parameters |
| | <------ | BILL VALIDATOR CONFIG. | |
| **ACK** | ------> | | |
| | | | |
| **SET SECURITY** | ------> | | Update bill security levels. |
| | <------ | ACK | |
| | | | |
| **IDENTIFICATION** | ------> | | Collect asset information. |
| | <------ | BILL VALIDATOR ID | |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | INITIALIZE | Bill Validator is initializing. |
| **ACK** | ------> | | |
| | | . | |
| | | . | |
| | | . | |
| **POLL** | ------> | | |
| | <------ | INITIALIZE | Bill-to-Bill unit is initializing |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | DISABLED | Bill-to-Bill unit completed initialization process and is ready for work |
| **ACK** | ------> | | |
| | | | |
| **POWER RECOVERY**** | ------> | | Report Power Up status and dispenser status |
| | <------ | POWER CUT WHILE DISPENSING | |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | DISABLED | All bill types are disabled. |
| **ACK** | ------> | | |
| | | | |
| **EMPTY DISPENSER** | ------> | | |
| | <------ | ACK | All bills are ejected from dispenser. |
| | | | |
| **POLL** | ------> | | |
| | <------ | DISPENSING | Transporting bills to dispenser |
| | | | |
| **ACK** | ------> | | |
| | | . | |
| | | . | |
| **POLL** | ------> | | |

| | | | |
|---|---|---|---|
| | <------ | DISPENSED | Dispense is completed. Bills removed from the dispenser. |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | DISABLED | All bill types are disabled. |
| **ACK** | ------> | | |

\*- The SET OPTIONS command is optional and should be sent if it is planned to use feature not enabled by default.
\*\* - On 'Power up' the validator will check if there were bills in dispensing process just before the power down. A bill is considered to be in dispensing process if:

1. It was in the channel.
2. It was inside of dispenser.

The bill (or several bills), which were in process will be moved either back to the cassette or into dispenser and dispenser state will be reported in the response to POWER RECOVERY command. The POWER RECOVERY command is optional and can be skipped. If the Bill-To-Bill unit is enabled by BILL TYPE command before issuing POWER RECOVERY command it will report a FAILURE state.

## 7.11 Power Up sequence after power cut during dispensing (Power recovery command was not sent)

| Controller | | Bill-to-Bill Unit | Comments |
|---|---|---|---|
| **POLL** | ------> | | |
| | <------ | POWER UP | Power |
| **ACK** | ------> | | |
| | | | |
| **RESET** | ------> | | Reset peripheral |
| | <------ | ACK | |
| | | | |
| **POLL** | ------> | | |
| | <------ | INITIALIZE | Bill Validator is initializing |
| **ACK** | ------> | | |
| | | | |
| **SET OPTIONS\*** | ------> | | Enabling miscellaneous options |
| | <------ | ACK | |
| | | | |
| **GET STATUS** | ------> | | Collect operational parameters |
| | <------ | BILL VALIDATOR CONFIG. | |
| **ACK** | ------> | | |
| | | | |
| **SET SECURITY** | ------> | | Update bill security levels. |
| | <------ | ACK | |
| | | | |
| **IDENTIFICATION** | ------> | | Collect asset information. |
| | <------ | BILL VALIDATOR ID | |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | INITIALIZE | Bill Validator is initializing. |
| **ACK** | ------> | | |
| | | . | |
| | | . | |
| | | . | |
| **POLL** | ------> | | |
| | <------ | INITIALIZE | Bill-to-Bill unit is initializing |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | DISABLED | Bill-to-Bill unit completed initialization process and is ready for work |
| **ACK** | ------> | | |
| | | | |
| **ENABLE BILL TYPES** | ------> | | Enable appropriate bill types |
| | <------ | ACK | |
| | | | |
| **POLL** | ------> | | |
| | <------ | FAILURE | Dispenser failure. |
| **ACK** | ------> | | |

*- The SET OPTIONS command is optional and should be sent if it is planned to use feature not enabled by default.
** - On 'Power up' the validator will check if there were bills in dispensing process just before the power down. A bill is considered to be in dispensing process if:
      1. It was in the channel.
      2. It was inside of dispenser.
The bill (or several bills), which were in process will be moved either back to the cassette or into dispenser and dispenser state will be reported in the response to POWER RECOVERY command. The POWER RECOVERY command is optional and can be skipped. If the Bill-To-Bill unit is enabled by BILL TYPE command before issuing POWER RECOVERY command it will report a FAILURE state.

### 7.12 Power Up sequence after power cut during unloading (Power recovery command was sent)

| Controller | | Bill-to-Bill Unit | Comments |
|---|---|---|---|
| **POLL** | ------> | | |
| | <------ | POWER UP | Power |
| **ACK** | ------> | | |
| | | | |
| **RESET** | ------> | | Reset peripheral |
| | <------ | ACK | |
| | | | |
| **POLL** | ------> | | |
| | <------ | INITIALIZE | Bill Validator is initializing |
| **ACK** | ------> | | |
| | | | |
| **SET OPTIONS*** | ------> | | Enabling Power Recovery option |
| | <------ | ACK | |
| | | | |
| **GET STATUS** | ------> | | Collect operational parameters |
| | <------ | BILL VALIDATOR CONFIG. | |
| **ACK** | ------> | | |
| | | | |
| **SET SECURITY** | ------> | | Update bill security levels. |
| | <------ | ACK | |
| | | | |
| **IDENTIFICATION** | ------> | | Collect asset information. |
| | <------ | BILL VALIDATOR ID | |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | INITIALIZE | Bill Validator is initializing. |
| **ACK** | ------> | | |
| | | | |
| | | . | |
| | | . | |
| | | . | |
| **POLL** | ------> | | |
| | <------ | INITIALIZE | Bill-to-Bill unit is initializing |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | DISABLED | Bill-to-Bill unit completed initialization process and is ready for work |
| **ACK** | ------> | | |
| | | | |
| **POWER RECOVERY**** | ------> | | Report Power Up status |
| | <------ | POWER CUT WHILE UNLOADING | |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | DISABLED | All bill types are disabled. |
| **ACK** | ------> | | |
| | | | |

*- The SET OPTIONS command is optional and should be sent if it is planned to use feature not enabled by default.
** - On 'Power up' the validator will check if there were bills in unloading process just before the power down. A bill is considered to be in unloading process if:
     1. It was in the channel.
     2. It was inside the drop box but was not properly stacked.
     3. ACK to the UNLOADED message was not received.

The bill (or several bills), which were in process will be moved into drop box and stacked. The POWER CUT WHILE UNLOADING will be reported in the response to POWER RECOVERY command. The POWER RECOVERY command is optional and can be skipped.

### 7.13 Drop Cassette Removal and Replacement sequence

| Controller | | Bill-to-Bill Unit | Comments |
|---|---|---|---|
| **POLL** | ------> | | |
| | <------ | DROP CASSETTE OUT OF POSITION | Drop cassette is removed |
| **ACK** | ------> | | |
| | | | |
| | | **DROP CASSETTE IS ATTACHED AT THIS POINT** | |
| | | | |
| **POLL** | ------> | | |
| | <------ | INITIALIZE | Bill-to-Bill is initializing |
| **ACK** | ------> | | |
| | | | |
| **GET STATUS** | ------> | | Collect operational parameters |
| | <------ | BILL-TO-BILL UNIT CONFIG. | |
| **ACK** | ------> | | |
| | | | |
| **SET SECURITY** | ------> | | Update bill security levels |
| | <------ | ACK | |
| | | | |
| **IDENTIFICATION** | ------> | | Collect asset information. |
| | <------ | BV UNIT ID | |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | INITIALIZE | Bill-to-Bill unit  is initializing |
| **ACK** | ------> | | |
| | | | |
| **POLL** | ------> | | |
| | <------ | UNIT DISABLE | All bill types are disabled |
| **ACK** | ------> | | |
| | | | |
| | | | |
| | | | |