MDB Coin Changer and MDB Bill Acceptor to RS-232 Interface

Model: MDB-RS232

DESCRIPTION

This Interface Unit (IU) is designed to convert the MDB Bill Acceptor (BA) and MDB Coin Changer (CC) protocol to RS-232 serial protocol. The IU functions as a slave to a master control system. The master may be any control system that supports RS-232 serial communication, for example a PC.

OPERATION

The master and slave communication is based on master requests and slave replies. The master sends a status request (or command) to the IU and IU answers by sending its current status. The IU’s typical response to the Master’s request (polling) consists of a header and data bytes. If there is no BA or CC activity, the status is one byte, the header only. When the IU receives data from the BA or CC, it sends a header followed by data bytes in the response to master’s status request. The master must always poll (query) the IU. If the IU does not receive a query from the master in the predefined time slot or IU does not recognize master messages 10 times in row, it will inhibit the BA and CC acceptance. At power up, acceptance of the BA and CC is inhibited.

SPECIFICATION

Communication format.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud rate</td>
<td>9600BPS</td>
</tr>
<tr>
<td>Start bit</td>
<td>1</td>
</tr>
<tr>
<td>Data bits</td>
<td>8</td>
</tr>
<tr>
<td>Parity</td>
<td>NO</td>
</tr>
<tr>
<td>Stop bit</td>
<td>1</td>
</tr>
</tbody>
</table>

Shanghai Wafer Microelectronics Co., Ltd.
Room 803, Building 1, 1389, DongXiu Road, Shanghai, China  Tel.: (0086)21 5187-0528  Fax: (0086)21 5045-4820
Timing Specification.

Inter-byte (max) 1ms
Response (max) 50 ms
Master polling times every 150 – 300 ms
Time IU waits for the Master’s poll before inhibiting acceptance of the BA and CC (max) 3 second

Hardware Specification.

Power
Min 20 VDC
Nominal 24...34 VDC (or 24...30VAC)
Consumes an idle current < 0.05A
0.5A for 1 sec. when accepting bills

Cable Standard computer serial cable
Dimensions approx. 80 x 50 mm
Weight approx. 100 g (0.2 lb.)

Software Specification

• Master Command codes (HEX)

Get Status (Poll) 0x01
Reset BA 0x02
Get BA Set-up Status 0x03
Enable/Disable BA 0x04
Accept Bill 0x05
Return Bill 0x06
Get Stacker Info 0x07
BA Security command 0x08
BA Expansion command 0x09
Reset CC 0x0A
Get CC Set-up Status 0x0B
Enable/Disable CC 0x0C
Get CC Tube Status 0x0D
CC Change command 0x0E
CC Expansion command 0x0F

• IU (slave) status response codes
  ACK 0x00
  NACK 0xFF

Response to the Get Status command includes one byte long Header with Status information. The Header has the following structure:
A. No BA or CC activity
Bit 7,6
0 1 – no activity
Bit 5,4
0 0 – no MDB attached
0 1 – CC attached
1 1 – BA & CC attached
Bit 3,2-reserved
Bit 1
1 – if a BA is attached and is disabled by the IU (see: OPERATION)
Bit 0
1 – if a CC is attached and is disabled by the IU (see: OPERATION)
No data bytes are following
B. Activity from BA and CC
Bit 7
1 - data message
Bit 6,5,4
0 0 0 – data from CC
0 1 0 – data from BA
Bit 3,2,1– reserved
Bit 0
If data from CC:
0 – no BA attached
1 – BA attached
If data from BA
0 – no CC attached
1 – CC attached
The byte after the Header B (activity on the lines from the BA or CC) states
the number of data bytes that are following. The last byte is a check sum of
all the sent bytes, including the Header.
In response to the master’s command that requests for data from the MDB
devices, the IU sends:
a. The first byte is the number of data bytes
b. The next byte(s) are the data byte(s)
c. The last byte is the check sum of the data bytes only.
In response to the master’s command that sends data to the MDB devices,
the IU sends ACK, however, if the IU does not recognize the master’s
command it sends back NACK. If the IU does not recognize the master’s
command, 10 times in row, it will inhibit acceptance of the BA and CC.
For data explanation see the Data Specification.
APPLICATION NOTES
The method of communication recommended is as follows:
   a. After power up the Master must start sending Get Status (poll) commands to the IU and IU must respond with its status.
   b. When communication is established, the Master can send the “Reset” command. The IU should respond with the message ACK.
   c. Then the Master sends the “Get Set-up Status” command. The IU responds with the set-up status. See Data Specification.
   d. Based on the information received in ‘c’ the Master builds bytes for the Enable/Disable command and sends it. The IU will respond with the message ACK.
   e. If above operation is OK the MDB device is ready to accept money.

DATA SPESIFICATION

<table>
<thead>
<tr>
<th>Master Command</th>
<th>Code</th>
<th>IU Response Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset BA</td>
<td>0x02h</td>
<td>ACK</td>
</tr>
</tbody>
</table>

**BA data specification**

- **Master Command**: Get BA Set-up Status
- **Code**: 0x03H
- **IU Response Data**: max 28 bytes: B1 – B27

- **B1**: Number of bytes that follow
- **B2**: Bill Validator Feature Level 1 byte
  - Indicates current feature level of the bill validator. Currently defined level one.
- **B3 – B4**: Country Code – 2 bytes
  - The International Telephone Code for the country that the Validator is set-up for. Sent in packed BCD. For example, the code for the USA is 00 01H
- **B5 – B6**: Bill Scaling Factor – 2 bytes
  - All accepted bill values must be evenly divisible by this number. For example, this could be set to 0064H for the USA.
- **B7**: Decimal Places – 1 byte
  - Indicates the number of decimal places on a credit display. For example, this could be set to 02H for the USA
- **B8 – B9**: Stacker Capacity – 2 bytes
  - Indicates the number of bills that the stacker will hold. For example, a stacker with a 400 bill capacity = 0190H.
B10 – B11  Bill Security Levels – 2 bytes
Indicates the security level for bill types 0 to 15. Since not all validators support multiple security levels, validators that do not have this feature must report a “high” security level.

B12  Escrow/No Escrow – 1 byte
Indicates the escrow capacity of the bill validator. If Z11 = 0HH, the bill validator does not have escrow capability. If Z11 = FFH, the bill validator has escrow capability.

B13 – B28  Bill Type Credit – 16 bytes
Indicates the value of the bill types 0 to 15. Values must be sent in ascending order. This number is the bill’s monetary value divided by the bill scaling factor. Unused bill types are sent as OOH. Unsent bill types are assumed to be zero. FFH bills are assumed to be vend tokens.

B29  (or last byte) Check Sum. This byte is a sum of all bytes except first byte.

Master Command  Code  IU Response Data
Enable/Disable B  0x04  ACK

The Master command is 6 bytes.

B1  command 0x04
B2-B3  Bill enable. Indicates what types of bills are accepted.
b15 b14 b13 b12 b11…b2 b1 b0
B2           B3
Bill types are 0 to 15. A bit is set to indicate acceptance of that bill type.
Sending 0000h disables the Bill Validator

B4-B5  Bill Escrow Enable
b15 b14 b13 b12 b11…b2 b1 b0
B4           B5
Bill types are 0 to 15. A bit is set to indicate enable of escrow for a bill type.

B6  Check sum of the B2..B5

Master Command  Code  IU Response Data
Accept Bill  0x05  ACK

Master Command  Code  IU Response Data
Return Bill  0x06  ACK

Master Command  Code  IU Response Data
Get Stacker Info  0x07  4bytes

B1  Number of bytes that follow
B2  “Fxxxxxxx” F=1 if the stacker is full, 0 if not
B3 “xxxxxxxx” together with B2 – the number of bills in the stacker
B4 Check Sum (except B1)

<table>
<thead>
<tr>
<th>Master Command</th>
<th>Code</th>
<th>IU Response Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>BA Security command</td>
<td>0x08</td>
<td>ACK</td>
</tr>
</tbody>
</table>

The Master command are 4 bytes:
B1 – 0x08
B2,B3 – data
B4 – check sum of B1 and B2

b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
B2 B3

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

<table>
<thead>
<tr>
<th>Master Command</th>
<th>Code</th>
<th>IU Response Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>BA Expansion command</td>
<td>0x09</td>
<td>31 bytes</td>
</tr>
</tbody>
</table>

The Master command is 3 bytes:
Y1 is the command 0x08
Y2 is 0x0
Y3 is checksum of data byte (only B2) – 0x0

Response bytes:
B1 - number of bytes that follows
B2-B4 – Manufacture Code of a BA
B5-B16 – Serial Number of the BA
B17- B28 – Model and revision of the BA
B29-B30 – Software Version of the BA

**Bill Acceptor activity data.**

<table>
<thead>
<tr>
<th>Master Command</th>
<th>Code</th>
<th>IU Response Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get Status (Poll)</td>
<td>0x01H</td>
<td>Header and possibly BA data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>May be up to 16 bytes</td>
</tr>
</tbody>
</table>

If a bill is accepted, the IU will send 4 bytes.
B1 data message header
B2 number of bytes that follow
B3 one of the following:

B3

NOTE: These responses should be used to add or subtract credit.
yyy = Bill Routing;  
000: BILL STACKED  
001: ESCROW POSITION  
010: BILL RETURNED  
011: NOT USED  
100: DISABLED BILL  
REJECTED

xxxx = Bill Type (0 to 15)

B4 = check sum

In the case of other BA activity:

The following bytes may exist:

(00000001) = Defective Motor
- One of the motors has failed to perform its expected assignment.

(00000010) = Sensor Problem
- One of the sensors has failed to provide its response.

(00000011) = Validator Busy
- The validator is busy and cannot answer a detailed command right now.

(00000100) = ROM Checksum Error
- The validator’s internal checksum does not match the calculated checksum.

(00000101) = Validator Jammed
- A bill(s) has jammed in the acceptance path.

(00000110) = Validator was reset
- The validator has been reset since the last POLL.

(00000111) = Bill Removed
- A bill in the escrow position has been removed by an unknown means. A BILL RETURNED message should also be sent.

(00001000) = Cash Box out of position
- The validator has detected the cash box to be open or removed.

(00001001) = Unit Disabled
- The validator has been disabled, by the VMC or because of internal conditions.

(00001010) = Invalid Escrow request
- An ESCROW command was requested for a bill not in the escrow position.

(00001011) = Bill Rejected
- A bill was detected, but rejected because it could not be identified.

(010xxxxx) = Number of attempts to input a bill while validator is disabled.

NOTES: The validator may send several of one type activity, up to 16 bytes total
1 Sent once each occurrence  
2 Sent once each POLL
3 Sent once each occurrence. The unit is then disabled until the condition is removed. Validator will respond with “unit disabled” until repaired or replaced.

**CC data specification**

<table>
<thead>
<tr>
<th>Master Command</th>
<th>Code</th>
<th>IU Response Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset CC</td>
<td>OxoA</td>
<td>ACK</td>
</tr>
<tr>
<td>Master Command</td>
<td>Code</td>
<td>IU Response Data</td>
</tr>
<tr>
<td>Get CC Set-up Status</td>
<td>OxoB</td>
<td>24 bytes: B1 - B24</td>
</tr>
</tbody>
</table>

B1 number of following bytes

B2 Changer Feature Level - 1 byte

Indicates the feature level of the changer. This will distinguish the changer feature level to the VMC.

Current defined levels:

- **Level 2**: Supports "core" command set. These are: RESET, STATUS, and TUBE STATUS, POLL, COIN TYPE, and DISPENSE.
- **Level 3**: Supports level two and the EXPANSION command addition changer model number, manufacturer code, turning revision, etc. See the details of EXPANSION command later in this document.

B3 – B4= Country Code - 2 bytes The International Telephone Code for the country that the changer is set-up for, is sent in packed BCD. For example, the USA code is 00 01H

B5 Coin Scaling Factor - 1 byte All accepted coin values must be evenly divisible by this number.

For example, this could be set to 05H for the USA nickel.

B6 Decimal Places - 1 byte

Indicates the number of decimal places on a credit display.

For example, this could be set to 02H in the USA.

B7 – B8 Coin Type Routing - 2 bytes

Indicates what coin types can be routed to the Changer's tubes. b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 B7 B8 Bit is set to indicate a coin type can be routed to the tube. Valid coin type is 0 to 15.

B9 – B24 Indicates the value of coin types 0 to 15. Values must be sent in ascending order. This number is the coin's monetary value divided by the coin scaling factor. Unused coin types are sent as 00H. Unsent coin types are assumed to be zero. It is not necessary to send all coin types. Coin type credits sent as FFH are assumed to be vend tokens. That is, their value is assumed to
The bytes position in the 16 byte string indicates the coin type(s). For example, the first byte sent would indicate the value of coin type 0, the second byte sent would indicate the value of coin type 1, and so on. For example, the USA coin types may be; Coin type 0 = nickel, Coin type 1 = dime, Coin type 2 = quarter, Coin type 3 = dollar.

<table>
<thead>
<tr>
<th>Master Command</th>
<th>Code</th>
<th>IU Response Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable/Disable</td>
<td>0x0C</td>
<td>ACK</td>
</tr>
</tbody>
</table>

The Master command is 6 bytes.

**B1 command 0x0C**

B2 – B3 Coin Enable - 2 bytes

<table>
<thead>
<tr>
<th>b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>B2</td>
</tr>
</tbody>
</table>

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 indicated coin type 15, and so on. To disable the changer, disable all coin types by sending a data block containing 000H. All coins are automatically disabled upon reset.

**B4 – B5= Manual Dispense Enable - 2 bytes**

<table>
<thead>
<tr>
<th>b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>B4</td>
</tr>
</tbody>
</table>

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 switches are automatically enabled upon reset.

**B6 is a check sum of the data byte (B2..B5)**

<table>
<thead>
<tr>
<th>Master Command</th>
<th>Code</th>
<th>IU Response Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get CC Tube Status</td>
<td>0x0D</td>
<td>20 bytes</td>
</tr>
</tbody>
</table>

B1 number of following bytes

B2 – B3 Tube Full Status - 2 bytes

Indicates status of coin tube for coin types 0 to 15.

<table>
<thead>
<tr>
<th>b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0</th>
</tr>
</thead>
<tbody>
<tr>
<td>B2</td>
</tr>
</tbody>
</table>

A bit is set to indicate a full tube. For example, bit 7 = set would indicate the tube for coin type 7 is full.

**B4 – B19 Tube Status - 16 bytes**

Indicates the greatest number of coins that the changer "knows" definitely are present in the coin tubes. A bytes position in the 16 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. Unsent bytes are assumed to be zero.

NOTE: If a 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.

B20 is check sum

<table>
<thead>
<tr>
<th>Master Command</th>
<th>Code</th>
<th>IU Response Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC Change command</td>
<td>0x0E</td>
<td>ACK</td>
</tr>
</tbody>
</table>

The Master command is 3 bytes.

B1 command 0x0E
B2 b7 b6 b5 b4 b3 b2 b1 b0
Bits b3, b2, b1, b0 indicate coin type to be dispensed. Valid codes are 0H to FH to indicate coin types 0 to 15.
Bits b7, b6, b5, b4 indicated the number of coins to be dispensed.
NOTE: If two coin types have the same value, the highest coin type should be paid out first.

B3 is a check sum (the same as B2)

**Coin Changer activity data.**

<table>
<thead>
<tr>
<th>Master Command</th>
<th>Code</th>
<th>IU Response Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get Status (Poll)</td>
<td>0x01H</td>
<td>Header and possibly BA data</td>
</tr>
</tbody>
</table>

B1  data message header
B2  number of bytes that are following
Then may follow up to 16 CC data bytes. Last byte always is a check sum.

CC data bytes.
Coins Dispensed Manually:

<table>
<thead>
<tr>
<th>Byte 1</th>
<th>Byte 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(lyyxxxx)</td>
<td>(zzzzzzzz)</td>
</tr>
</tbody>
</table>
yyy = The number of coins dispensed.
xxx = The coin type dispensed (0 to 15).
zzzzzzzz = The number of coins in the tube.

Coins Deposited:

<table>
<thead>
<tr>
<th>Byte 1</th>
<th>Byte 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(01yyxxxx)</td>
<td>(zzzzzzzz)</td>
</tr>
</tbody>
</table>
yy = Coin routing
00: CASH BOX
01: TUBES
10: NOT USED
11: REJECT
MDB-RS232 Interface Adapter for MDB Coin Changer and Bill Acceptor.

xxx = Coin type deposited (0 to 15).
zzzzzzz = The number of coins in the tube for the coin type accepted.

In case of other CC activity, the following bytes may be:
(00000001) = Escrow request 1 - An escrow lever activation has been detected.
(00000010) = Changer Payout Busy 2 - The changer is busy activating payout devices.
(00000011) = No Credit 1 - A coin was validated but did not get to the place in the system when credit is given.
(00000100) = Defective Tube Sensor 1 - The changer has detected one of the tube sensors behaving abnormally.
(00000101) = Double Arrival 1 - Two coins were detected too close together to validate either one.
(00000110) = Acceptor Unplugged 2 - The changer has detected that the acceptor has been removed.
(00000111) = Tube Jam 1 - A tube payout attempt has resulted in jammed condition.
(00001000) = ROM checksum error 1 - The changers internal checksum does not match the calculated checksum.
(00001001) = Coin Routing Error 1 - A coin has been validated, but did not follow the intended routing.
(00001010) = Changer Busy 2 - The changer is busy and can not answer a detailed command right now.
(00001011) = Changer was Reset 1 - The changer has detected an reset condition and has returned to its power-on idle condition.
(00001100) = Coin Jam 1 - A coin(s) has jammed in the acceptance path.
(00001101) = Not Used
(00001110) = Not Used
(00001111) = Not Used

Slug:
(001xxxxx) = xxxxx is the number of slugs since the last activity.

NOTES: The Changer may send several of one type activity, up to 16 bytes total. This will permit zeroing counters such as slug, inventory, and status.
1 Sent once each occurrence
2 Sent once each POLL

Master Command   Code   Sub-command   IU Response Data
CC Expansion command  0x0F  0x0  35 bytes
The Master command is 3 bytes.

Y0 – 0x0E
Y1-0x0
Y2 – 0x0
IU response:
   B1 number of following bytes
   B2..B30 the same as for BA
   B31..34 Optional features
   Each of the 32 bits indicate an optional features. If the bit is set the feature is available.
   b0 – Alternative pay-out method. This method allows changer designs that determine change payout.
   b1 – Extended diagnostic command supported.
   b2 – Controlled manual fill and pay-out commands
   b3..b31 for future use.

Master Command     Code   Sub-command   IU Response Data
CC Expansion command  0x0F   0x01   ACK

Features enable. This command is used to enable optional futures defined in B31...B34 above.
The Master command is 6 bytes.
B1 – 0x0E
B2 – 0x01
B3..B6 – data bytes
B7 is check sum of B2...B6

Master Command     Code   Sub-command   IU Response Data
CC Expansion command  0x0F   0x02   ACK

The Master command is 4 bytes.
B1 – 0x0E
B2 – 0x02
B3 – Value of coins to be paid out. This value is expressed as the number of coin scaling factors that would sum to the value.
B4 – is check sum of the B1 and B2

Master Command     Code   Sub-command   IU Response Data
CC Expansion command 0x0F 0x03 18bytes

The Master command is 3 bytes.
- \( Y1 = 0x0E \)
- \( Y2 = 0x03 \)
- \( Y3 = 0x03 \)

IU response:
- B1 the number of following bytes
- B2…B17 Number of each coin type paid out
- B18 – check sum

The rest of valid for IU expansion command are:

<table>
<thead>
<tr>
<th>Master Command</th>
<th>Code</th>
<th>Sub-command</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC Expansion command</td>
<td>0x0F</td>
<td>0x04</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Master Command</th>
<th>Code</th>
<th>Sub-command</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC Expansion command</td>
<td>0x0F</td>
<td>0x05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Master Command</th>
<th>Code</th>
<th>Sub-command</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC Expansion command</td>
<td>0x0F</td>
<td>0x06</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Master Command</th>
<th>Code</th>
<th>Sub-command</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC Expansion command</td>
<td>0x0F</td>
<td>0x07</td>
</tr>
</tbody>
</table>

The detail description of these commands can be found in the document “Multi-Drop Bus Communication protocol” – MDB/ICP. Knowledge of this protocol would be beneficial to developers using this product.
MDB-RS232 Interface Adapter for MDB Coin Changer and Bill Acceptor.

LED is used to show the communication status. When the data communication is working, then the LED would flash quickly.

Test software interface:
This brochure provides an overview of the products and services of WEIFU MDB-RS232 Adapter, For further information and queries kindly contact:

Email: coinco.cn@yahoo.com.cn
Web: www.waferstar.com
Tel: 0086-21-51870528

Also accessible via our online MSN service: frank_ic2001@hotmail.com