

**Interface MDB master – RS232**  
**v.21.03.2017**  
**Quick reference**

# Table of Contents

|   |    |
|---|----|
| I. General informations.....              | 4  |
| 1. Terms.....                             | 4  |
| 2. Working modes.....                     | 4  |
| A. Transparent mode.....                  | 4  |
| B. Direct mode.....                       | 5  |
| C. High level protocol.....               | 5  |
| 3. Communication parameters.....          | 5  |
| II. Hardware overview.....                | 6  |
| 1. Power supply requirements.....         | 6  |
| 2. Connector description.....             | 6  |
| III. Transparent mode protocol.....       | 7  |
| 1. MDB bill validator initialization..... | 7  |
| 2. MDB bill enable.....                   | 7  |
| 3. MDB bill disable.....                  | 7  |
| 4. MDB bill read setup vector.....        | 8  |
| 5. Coin acceptor initialization.....      | 8  |
| 6. Coin acceptor enable.....              | 8  |
| 7. Coin acceptor disable.....             | 8  |
| 8. Coin acceptor read setup vectors.....  | 9  |
| 9. Set maximum credit.....                | 9  |
| 10. Reset the current credit.....         | 9  |
| 11. Return change.....                    | 10 |
| 12. Set current cash credit.....          | 10 |
| 13. Poll credit and devices status.....   | 11 |
| 14. Reset all devices' status.....        | 14 |
| 15. Cashless INIT.....                    | 14 |
| 16. Cashless ENABLE.....                  | 15 |
| 17. Cashless DISABLE.....                 | 15 |
| 18. Cashless VEND CANCEL.....             | 15 |
| 19. Cashless VEND REQUEST.....            | 16 |
| 20. Cashless VEND SUCCESS.....            | 16 |
| 21. Cashless VEND FAIL.....               | 16 |
| 22. Cashless REVALUE.....                 | 17 |
| 23. Cashless READ VECTORS.....            | 17 |
| 24. Cashless END SESSION.....             | 17 |
| IV. High level protocol.....              | 18 |
| 1. MDBBillInit.....                       | 18 |
| 2. MDBBillEnable.....                     | 18 |
| 3. MDBBillDisable.....                    | 19 |
| 4. MDBCoinInit.....                       | 19 |
| 5. MDBCoinEnable.....                     | 19 |
| 6. MDBCoinDisable.....                    | 20 |
| 7. MDBBillSettings.....                   | 20 |
| 8. MDBCoinSettings.....                   | 21 |
| 9. MDBSetMaxCredit.....                   | 22 |
| 10. MDBSetCurrentCredit.....              | 22 |
| 11. MDBSetChange.....                     | 23 |

12. MDBCreditReset.....23  
13. MDBPoll.....24  
14. MDBResetStatus.....24  
15. MDBCashlessInit.....25  
16. MDBCashlessEnable.....25  
17. MDBCashlessDisable.....25  
18. MDBCashlessSettings.....26  
19. MDBCashlessVendRequest.....26  
20. MDBCashlessVendSuccess.....27  
21. MDBCashlessVendFailed.....27  
22. MDBCashlessRevalue.....27  
23. MDBCashlessEndSession.....28

# I. General informations

## 1. Terms

- **INTERFACE** = the MDB-RS232 interface
- **HOST** = the computer or SBC board (Raspberry PI, Banana PI, etc.) that will send command to the INTERFACE using an RS232 serial port.
- **MDB PERIPHERALS** = payment systems connected on the MDB bus.
- **ACK** = acknowledge
- **NACK** = not-acknowledge

## 2. Working modes

The INTERFACE can be used in two modes: transparent mode and direct mode. The INTERFACE automatically changes the working mode, depending on the received commands..

### A. Transparent mode

In transparent mode, there are simple commands to manage the MDB devices. The built-in firmware will handle MDB commands and it is the ideal mode where the developers don't need to learn any MDB command and response. Also there is no need to calculate the MDB checksum since this is automatically calculated by the INTERFACE and correctly sent to the MDB peripheral. When the INTERFACE will receive a transparent mode command, it automatically turn TRANSPARENT MODE ON and begin to continuously poll the MDB PERIPHERALS.

A proprietary simple message structure is available to communicate with bill validators and coin acceptors/changers. The general message format is detailed in table 1.

| <HEADER>    | <CMD>  | <SUBCMD> | <PARAMETERS>                             | <CRC>  |
|-------------|--------|----------|--|--------|
| always 0xFE | 1 byte | 1 byte   | variable length, depending on subcommand | 1 byte |

*Table 1: Transparent mode message format*

- **<HEADER>** – Is always a byte with a value of 0xFE.
- **<CMD>** - Defines the command group sent to the INTERFACE.
- **<SUBCMD>** - Defines the specific subcommand of the command group.
- **<PARAMETERS>** - Defines the command parameters that will be send to the MDB peripheral (for example, the maximum credit that the INTERFACE should accept or the change it should return from changer). Some commands are not requiring parameters. Also, the variable length depends on subcommand.
- **<CRC>** - Defines the message checksum. The <CRC> is calculated as an XOR of all message bytes, including the <HEADER>. For example, the command to enabled the bill validator is: 0xFE 0x42 0x02 and the CRC for this command is 0xBE. Transparent mode can be used for any application.

**IMPORTANT!!! - If the host (computer) is not sending any valid command within 60 seconds, then the interface will disable all payment systems to avoid cashing money when the host is not working.**

## B. Direct mode

This mode is a low level mode that can be used by the experienced developers when they want to send MDB commands directly to the peripheral and they have to calculate the MDB checksum, also. The response from the addressed MDB peripheral is sent back to the HOST. The CRC is verified on the messages received from MDB peripheral. The INTERFACE will handle the 9<sup>th</sup> bit (mode bit) and will receive and send messages back to the HOST in 8 bit format. The INTERFACE will also check CRC on messages received from MDB PERIPHERALS and will acknowledge to the MDB bus. The acknowledge action is very time critical on MDB and must be achieved in a short interval (lower than 5ms). Otherwise, the MDB peripheral will get that as a not-acknowledge message. If the INTERFACE receives a direct mode message, it will stop polling the MDB PERIPHERALS and automatically set the DIRECT MODE as a working mode.

## C. High level protocol

This protocol is used with a small daemon, written with Python 3, the daemon you can download on product page at <http://www.vendingtools.ro>

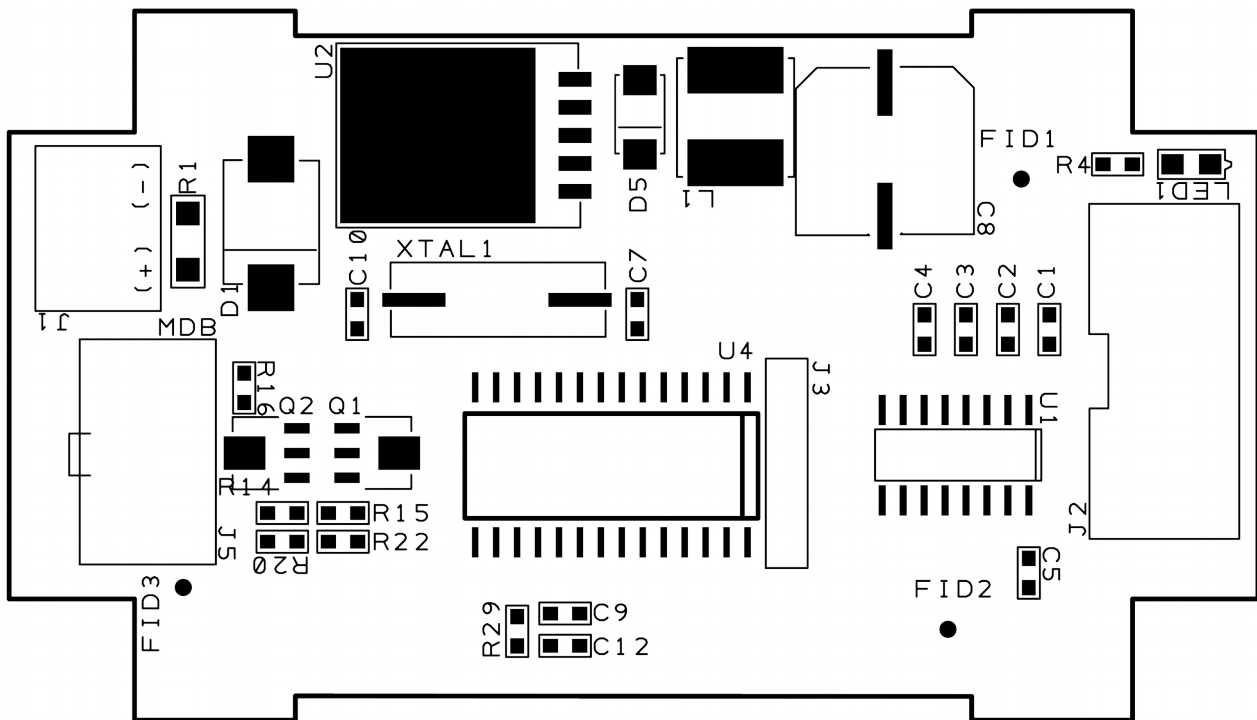
## 3. Communication parameters

The INTERFACE can be connected to any RS232 port or any USB to RS232 port. The communication settings should meet the following specifications:

| Parameter     | Value         |
|---------------|---------------|
| baud          | 57600         |
| data bits     | 8             |
| parity        | NONE          |
| hardware flow | YES (RTS/CTS) |
| software flow | NO            |

*Table 2: RS232 communication parameters*

## II. Hardware overview



Picture 1: Board overview

### 1. Power supply requirements

The INTERFACE can be powered with stabilized 24VDC or 12VDC, depending on your MDB PERIPHERALS. You must use a stabilized DC power supply with at least 2A output. It is necessary to follow the correct polarity. In the eventuality of an accidental polarity reversal, the entire board and the MDB PERIPHERALS are protected, but will not work.

### 2. Connector description

- **<J1>** – POWER connector for the INTERFACE and MDB PERIPHERALS. Use only stabilized power supplies, with a voltage rating according to your MDB PERIPHERALS. Also, be careful at the current rating, since this may vary from one MDB peripheral to another. Use your MDB peripheral manual to identify the power needs.
- **<J2>** - RS232 connector. For this port, the package includes a flat cable with all necessary connectors.
- **<MDB>** - Used to connect the MDB PERIPHERALS.

You do not need to perform any settings on the INTERFACE, neither hardware or software.

### III. Transparent mode protocol

Execution of any command in this chapter will turn the INTERFACE in transparent mode and will start the automatic MDB polling. The peripherals are polled continuously.

#### 1. MDB bill validator initialization

| <HEADER>         | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|------------------|-------|----------|---|-------|
| 0xFE             | 0x42  | 0x01     | [none]  | 0xBD  |
| INTERFACE answer |       |          |   |       |
| 0xFE             | 0x42  | 0x01     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error | CRC   |

This command will execute the initialization procedure for the MDB bill validator connected on the MDB port.

#### 2. MDB bill enable

| <HEADER>         | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|------------------|-------|----------|---|-------|
| 0xFE             | 0x42  | 0x02     | [none]  | 0xBE  |
| INTERFACE answer |       |          |   |       |
| 0xFE             | 0x42  | 0x02     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error | CRC   |

This command will enable the bill validator. This will accept all the banknotes he can recognize.

#### 3. MDB bill disable

| <HEADER>         | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|------------------|-------|----------|---|-------|
| 0xFE             | 0x42  | 0x03     | [none]  | 0xBF  |
| INTERFACE answer |       |          |   |       |
| 0xFE             | 0x42  | 0x03     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error | CRC   |

This command will disable the bill validator. This will not accept any banknote.

## 4. MDB bill read setup vector

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x42  | 0x04     | [none]  | 0xB8  |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x42  | 0x04     | - <BILL SETUP> - 27 bytes<br>- <BILL EXPANSION IDENTIFICATION> - 29 bytes | CRC   |

This command will return the settings vector for the MDB bill validator. Those vectors are read on the initialization phase. There are two vectors available and the contained data are detailed in the MDB documentation. This command is optional and is used only if you need to handle some lower informations (bill validator MDB level, software version, ISO country code, etc.).

## 5. Coin acceptor initialization

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x43  | 0x01     | [none]  | 0xBC  |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x43  | 0x01     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error | CRC   |

This command will execute the initialization procedure for coin acceptor/changer connected on the MDB port.

## 6. Coin acceptor enable

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x43  | 0x02     | [none]  | 0xBF  |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x43  | 0x02     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error | CRC   |

This command will activate the coin acceptor/changer. All recognized coins/tokens will be accepted and deposited.

## 7. Coin acceptor disable

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x43  | 0x03     | [none]  | 0xBE  |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x43  | 0x03     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error | CRC   |

This command will deactivate the coin acceptor/changer.



## 8. Coin acceptor read setup vectors

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x43  | 0x04     | [none]  | 0xB9  |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x43  | 0x04     | - <COIN SETUP> - 23 bytes<br>- <COIN EXPANSION IDENTIFICATION> - 33 bytes | CRC   |

This command will return the settings vector for the MDB coin acceptor/changer. Those vectors are read on the initialization phase. There are two vectors available and the contained data are detailed in the MDB documentation. This command is optional and is used only if you need to handle some lower informations (coin acceptor/changer MDB level, software version, ISO country code, etc.).

## 9. Set maximum credit

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>   | <CRC> |
|-------------------------|-------|----------|--|-------|
| 0xFE                    | 0x52  | 0x01     | <MAXIMUM CREDIT> - 4 bytes – MSB<br>Example: 0xFE 0x52 0x01 0x00 0x00 0x02 0x58 0xF7 – this will set the maximum credit to 600 units. In case of EUR or USD, this means 600 cents or 6.00EUR/6.00USD | CRC   |
| <b>INTERFACE answer</b> |       |          |  |       |
| 0xFE                    | 0x52  | 0x01     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error  | CRC   |

This command will set the maximum acceptable credit for bills. Any bill exceeding this value will be rejected. For coins, you should disable the MDB coin acceptor after reaching the maximum credit value.

## 10. Reset the current credit

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x52  | 0x02     | [none]  | 0xAE  |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x52  | 0x02     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error | CRC   |

The INTERFACE has an internal cash counter which is incremented for each bill or coin deposited by the MDB PERIPHERALS. This counter can be read by using a POLL command, detailed on “12. Poll credit and devices status”. For simplicity reasons, this counter can be reseted by this command. You can use this command after each transaction, or anytime you need.

## 11. Return change

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x52  | 0x03     | <CHANGE TO RETURN> - 4 bytes – MSB<br>Example: 0xFE 0x52 0x03 0x00 0x00 0x01 0x5E 0xF0 –<br>this command will return 350 change, which means, for<br>EUR and USD, 350cents or 3.50EUR/3.50USD | CRC   |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x52  | 0x03     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error   | CRC   |

This command will start returning change if the interface has a changer connected on the MDB port.

## 12. Set current cash credit

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x52  | 0x04     | <SET CURRENT CASH CREDIT> - 4 bytes – MSB<br>Example: 0xFE 0x52 0x04 0x00 0x00 0x01 0x5E 0xF7 –<br>this command will set current credit to 350, which means,<br>for EUR and USD, 350cents or 3.50EUR/3.50USD – This<br>is used to adjust credit in multivend mode and before the<br>cashless revalue command to set only remaining credit<br>for revalue. | CRC   |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x52  | 0x04     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error   | CRC   |

This command will set the current credit on the interface. It is indicated to use this command after every cash transaction finished with a successful vend.

### 13. Poll credit and devices status

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x50  | 0x01     | [none]  | 0xAF  |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x50  | 0x01     | <ul style="list-style-type: none"> <li>- &lt;CURRENT CREDIT CASH&gt; - 4 bytes MSB (for example, 0x00 0x00 0x04 0xE2 representing 1250 cents or 12.50EUR/12.50USD)</li> <li>- &lt;CURRENT CREDIT CASHLESS&gt; - 4 bytes MSB (for example, 0x00 0x00 0x04 0xE2 representing 1250 cents or 12.50EUR/12.50USD)</li> <li>- &lt;BILL validator status&gt; - 4 byte, according to MDB bill validators status (see Table 3) – this register contains last 4 bill validator status. If the register has the value 0xFFFFFFFF, then the bill validator is not initialized. The rightmost byte of this register is the last status.</li> <li>- &lt;COIN acceptor/changer status&gt; - 4 bytes, according to MDB coin acceptors/changers status (see Table 4)</li> <li>- &lt;CASHLESS #1 status&gt; - 4 bytes, according to the MDB cashless status (see Table 5)</li> <li>- &lt;CASHLESS #2 status&gt; - 4 bytes, according to the MDB cashless status (see Table 5)</li> <li>- &lt;CASHLESS MEDIA ID&gt; - 4 bytes, card/tag serial number</li> <li>- &lt;AVAILABLE CHANGE&gt; - 4 bytes – the total value of the coins available in the coin changer. This value is updated on every coin enable or coin disable command</li> </ul> | CRC   |

This command must be run periodically, at least one per second, to interrogate the payment systems status and to take all needed decisions.

| <b>Value</b> | <b>Description</b>  |
|--------------|---|
| <b>0x00</b>  | Idle  |
| <b>0x01</b>  | Defective Motor - One of the motors has failed to perform its expected assignment.  |
| <b>0x02</b>  | Sensor Problem - One of the sensors has failed to provide its response.   |
| <b>0x03</b>  | Validator Busy - The validator is busy and can not answer a detailed command right now.   |
| <b>0x04</b>  | ROM Checksum Error - The validators internal checksum does not match the calculated checksum.                                   |
| <b>0x05</b>  | Validator Jammed - A bill(s) has jammed in the acceptance path.   |
| <b>0x06</b>  | Validator was reset - The validator has been reset since the last POLL.   |
| <b>0x07</b>  | Bill Removed - A bill in the escrow position has been removed by an unknown means. A BILL RETURNED message should also be sent. |
| <b>0x08</b>  | Cash Box out of position - The validator has detected the cash box to be open or removed.                                       |
| <b>0x09</b>  | Validator Disabled - The validator has been disabled, by the VMC or because of internal conditions                              |
| <b>0x0A</b>  | Invalid Escrow request - An ESCROW command was requested for a bill not in the escrow position.                                 |
| <b>0x0B</b>  | Bill Rejected - A bill was detected, but rejected because it could not be identified.   |
| <b>0x0C</b>  | Possible Credited Bill Removal – There has been an attempt to remove a credited (stacked) bill.                                 |
| <b>0xFF</b>  | Not known or not initialized status   |

*Table 3: Bill validators status codes*

| <b>Value</b> | <b>Description</b>   |
|--------------|--|
| <b>0x00</b>  | Idle   |
| <b>0x01</b>  | Escrow request - An escrow lever activation has been detected.   |
| <b>0x02</b>  | Changer Payout Busy - The changer is busy activating payout devices.   |
| <b>0x03</b>  | No Credit - A coin was validated but did not get to the place in the system when credit is given.                |
| <b>0x04</b>  | Defective Tube Sensor - The changer has detected one of the tube sensors behaving abnormally.                    |
| <b>0x05</b>  | Double Arrival - Two coins were detected too close together to validate either one.                              |
| <b>0x06</b>  | Acceptor Unplugged - The changer has detected that the acceptor has been removed.                                |
| <b>0x07</b>  | Tube Jam - A tube payout attempt has resulted in jammed condition.   |
| <b>0x08</b>  | ROM checksum error - The changers internal checksum does not match the calculated checksum.                      |
| <b>0x09</b>  | Coin Routing Error - A coin has been validated, but did not follow the intended routing.                         |
| <b>0x0A</b>  | Changer Busy - The changer is busy and can not answer a detailed command right now.                              |
| <b>0x0B</b>  | Changer was Reset - The changer has detected an Reset condition and has returned to its power-on idle condition. |
| <b>0x0C</b>  | Coin Jam - A coin(s) has jammed in the acceptance path.  |
| <b>0x0D</b>  | Possible Credited Coin Removal – There has been an attempt to remove a credited coin.                            |
| <b>0xFF</b>  | Not known or not initialized status  |

*Table 4: Coin acceptors/changers status codes*

| Value | Description                                      |
|-------|--|
| 0x00  | Idle   |
| 0x01  | Not used in this configuration                   |
| 0x02  | Display request – not used in this configuration |
| 0x03  | Begin cashless session                           |
| 0x04  | Cancel request from cashless to VMC              |
| 0x05  | Vend approved.                                   |
| 0x06  | Vend denied.                                     |
| 0x07  | End cashless session.                            |
| 0x08  | Cancelled  |
| 0x09  | Not used in this configuration.                  |
| 0x0A  | Not used in this configuration.                  |
| 0x0B  | Command out of sequence.                         |
| 0x0C  | Not used in this configuration.                  |
| 0x0D  | Revalue approved.                                |
| 0x0E  | Revalue denied                                   |
| 0x0F  | Not used in this configuration.                  |

Table 5: Cashless status codes

## 14. Reset all devices' status

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x50  | 0x02     | [none]  | 0xAC  |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x50  | 0x02     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error | CRC   |

This command will set to 0x00 all devices' status. It is used to clear status and let the INTERFACE to update it in accordance with the new devices' status. It will not reset credits

## 15. Cashless INIT

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x53  | 0x01     | <CASHLESS NUMBER> (0x01 = cashless #1, 0x02 = cashless #2)  | CRC   |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x53  | 0x01     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error | CRC   |

This command will init the cashless device with the corresponding number

## 16. Cashless ENABLE

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x53  | 0x02     | <CASHLESS NUMBER> (0x01 = cashless #1, 0x02 = cashless #2)  | CRC   |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x53  | 0x02     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error | CRC   |

This command will enable the cashless device with the corresponding number

## 17. Cashless DISABLE

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x53  | 0x03     | <CASHLESS NUMBER> (0x01 = cashless #1, 0x02 = cashless #2)  | CRC   |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x53  | 0x03     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error | CRC   |

This command will disable the cashless device with the corresponding number

## 18. Cashless VEND CANCEL

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x53  | 0x04     | <CASHLESS NUMBER> (0x01 = cashless #1, 0x02 = cashless #2)  | CRC   |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x53  | 0x04     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error | CRC   |

This command will send “CANCEL CURRENT SESSION” command to the cashless device.

## 19. Cashless VEND REQUEST

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x53  | 0x05     | <CASHLESS NUMBER> 1 byte (0x01 = cashless #1, 0x02 = cashless #2)<br><VEND VALUE> - 4 bytes - the value of the selected product to sell - MSB (for example, 0x00 0x00 0x04 0xE2 representing 1250 cents or 12.50EUR/12.50USD)<br><PRODUCT ID> - 2 bytes – the number of the selected product. | CRC   |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x53  | 0x05     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error   | CRC   |

This command will send ask the permission to sell a product using cashless. This command requires to use POLL command to obtain the answer from the cashless device (according to Table 5).

## 20. Cashless VEND SUCCESS

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x53  | 0x06     | <CASHLESS NUMBER> (0x01 = cashless #1, 0x02 = cashless #2)<br><PRODUCT ID> - 2 bytes – the number of the product successfully dispensed to the customer | CRC   |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x53  | 0x06     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error   | CRC   |

This command must be sent after a successful vend using cashless. Following this command, the interface will subtract the product value from the cashless credit, and the cashless device will be instructed to subtract the same amount from the customer's credit.

## 21. Cashless VEND FAIL

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x53  | 0x07     | <CASHLESS NUMBER> (0x01 = cashless #1, 0x02 = cashless #2)  | CRC   |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x53  | 0x07     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error | CRC   |

This command must be sent after a failed sale situation (due to VMC error). The cashless will be instructed to refund the product price to the customer's account.



## 22. Cashless REVALUE

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x53  | 0x08     | <CASHLESS NUMBER> (0x01 = cashless #1, 0x02 = cashless #2)<br><VALUE> - 4 bytes - MSB               | CRC   |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x53  | 0x08     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error | CRC   |

This command will transform the remaining cash credit into cashless credit and will refund this amount on customer's account.

## 23. Cashless READ VECTORS

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>   | <CRC> |
|-------------------------|-------|----------|--|-------|
| 0xFE                    | 0x53  | 0x09     | <CASHLESS NUMBER> (0x01 = cashless #1, 0x02 = cashless #2)               | CRC   |
| <b>INTERFACE answer</b> |       |          |  |       |
| 0xFE                    | 0x53  | 0x09     | <CASHLESS SETUP> - 8 bytes<br><CASHLES EXPANSION INFORMATION> - 30 bytes | CRC   |

This command will return the low level settings of the cashless device, according to MDB protocol.

## 24. Cashless END SESSION

| <HEADER>                | <CMD> | <SUBCMD> | <PARAMETERS>  | <CRC> |
|-------------------------|-------|----------|---|-------|
| 0xFE                    | 0x53  | 0x0A     | <CASHLESS NUMBER> (0x01 = cashless #1, 0x02 = cashless #2)  | CRC   |
| <b>INTERFACE answer</b> |       |          |   |       |
| 0xFE                    | 0x53  | 0x0A     | 0xFB – command execution failed<br>0xFC – command successfully executed<br>0xFD – command CRC error | CRC   |

This command will return turn the session off and will force the cashless device to return to idle state.

## IV. High level protocol

Using this mode, the development becomes much easier. The communication with the board and the peripherals is managed by a small application. Commands are not case sensitive. We have used capitalization to facilitate reading.

It is a good idea for your application to retry sending the command few times if you get an “failed” answer. This answer can be returned in the event of a board communication failure. Also it can be returned if you try to address a not connected device.

### 1. MDBBillInit

| GUI command   |   |
|---|---|
| Command   | Parameters/Comments   |
| MDBBillInit   | [none]  |
| RASPIVEND daemon answer                                       |   |
| Answer  | Parameters/Comments   |
| {"MDBBillInit": "success"}<br>or<br>{"MDBBillInit": "failed"} | This command will perform all initialization tasks for the attached MDB bill validator. If something goes wrong or the MDB bill validator is not connected to the board, then the command returns “failed” message. |

### 2. MDBBillEnable

| GUI command   |  |
|---|--|
| Command   | Parameters/Comments  |
| MDBBillEnable   | [none]   |
| RASPIVEND daemon answer   |  |
| Answer  | Parameters/Comments  |
| {"MDBBillEnable": "success"}<br>or<br>{"MDBBillEnable": "failed"} | This command will activate the attached MDB bill validator. This command must be preceded by the MDBBillInit command. You cannot activate a bill validator if this one is not initialized. |

### 3. MDBBillDisable

| GUI command   |   |
|---|---|
| Command   | Parameters/Comments   |
| MDBBillDisable  | [none]  |
| RASPIVEND daemon answer   |   |
| Answer  | Parameters/Comments   |
| {"MDBBillDisable":"success"}<br>or<br>{"MDBBillDisable":"failed"} | This command will deactivate the attached MDB bill validator. |

### 4. MDBCoinInit

| GUI command   |  |
|---|--|
| Command   | Parameters/Comments  |
| MDBCoinInit   | [none]   |
| RASPIVEND daemon answer                                     |  |
| Answer  | Parameters/Comments  |
| {"MDBCoinInit":"success"}<br>or<br>{"MDBCoinInit":"failed"} | This command will initialize the attached MDB coin acceptor/changer. |

### 5. MDBCoinEnable

| GUI command   |  |
|---|--|
| Command   | Parameters/Comments  |
| MDBCoinEnable   | [none]   |
| RASPIVEND daemon answer   |  |
| Answer  | Parameters/Comments  |
| {"MDBCoinEnable":"success"}<br>or<br>{"MDBCoinEnable":"failed"} | This command will enable the attached MDB coin acceptor/changer. This command requires a previous MDBCoinInit command. You cannot enable a coin acceptor/changer if it was not previously initialized. |





## 9. MDBSetMaxCredit

| GUI command                 |   |
|-----------------------------|---|
| Command                     | Parameters/Comments   |
| MDBSetMaxCredit("NNNN")     | <p>Maximum MDB credit accepted<br/>- 32 bit positive value</p> <p>When there is a bill in escrow, where &lt;current_MDB_Credit&gt; + &lt;MDB_escrow_bill_value&gt; is bigger than NNNN, then that bill will be returned to the customer.</p> <p>For coins, your application must disable the coin acceptor/changer, once the maximum credit is reached.</p> <p>The value is multiplied by the 100 scaling factor (for example, for 1EUR you have to set this value to 100).</p> |
| RASPIVEND daemon answer     |   |
| Answer                      | Parameters/Comments   |
| {"MDBSetMaxCredit": "NNNN"} | <p>This command will return the value you have sent. For safety reasons (eliminate the communication errors), you must commit this action with MDBSetMaxCreditOK, after you verify that the returned value is the same with the sent value.</p>   |

## 10. MDBSetCurrentCredit

| GUI command                     |   |
|---------------------------------|---|
| Command                         | Parameters/Comments   |
| MDBSetCurrentCredit("NNNN")     | <p>Set the value of the MDB module current credit.<br/>- 32 bit positive value</p> <p>This value must be set after each successful vend and before issuing the "MDBCashlessRevalue" command. This is the value that the RASPIVEND board will try to use for revalue.</p> <p>The value is multiplied by the 100 scaling factor (for example, for 1EUR you have to set this value to 100).</p> <p>This command will modify the value of CurrentCreditCash. You can read this variable by issuing the MDBPoll command.</p> |
| RASPIVEND daemon answer         |   |
| Answer                          | Parameters/Comments   |
| {"MDBSetCurrentCredit": "NNNN"} | <p>This command will return the value you have sent. For safety reasons (eliminate the communication errors), you must commit this action with MDBSetCurrentCreditOK, after you verify that the returned value is the same with the sent value.</p>   |

## 11. MDBSetChange

| GUI command             |  |
|-------------------------|--|
| Command                 | Parameters/Comments  |
| MDBSetChange("NNNN")    | Set the value of the change that MDB module will return when the MDBSetChangeOK command will be issued.<br>- 32 bit positive value<br>This value must be set before any change return action<br>The value is multiplied by the 100 scaling factor (for example, for 1EUR you have to set this value to 100). |
| RASPIVEND daemon answer |  |
| Answer                  | Parameters/Comments  |
| {"MDBSetChange":"NNNN"} | This command will return the value you have sent. For safety reasons (eliminate the communication errors), you must commit this action with MDBSetChangeOK, after you verify that the returned value is the same with the sent value.  |

## 12. MDBCreditReset

| GUI command   |  |
|---|--|
| Command   | Parameters/Comments  |
| MDBCreditReset  | [none]   |
| RASPIVEND daemon answer   |  |
| Answer  | Parameters/Comments  |
| {"MDBCreditReset":"success"}<br>or<br>{"MDBCreditReset":"failed"} | This command will set the CuremntCreditCash variable to 0.<br>It is recommended to use this command before activating the payment systems on each transaction. |

## 13. MDBPoll

| GUI command  |  |
|--|--|
| Command  | Parameters/Comments  |
| MDBPoll  | [none]   |
| RASPIVEND daemon answer  |  |
| Answer   | Parameters/Comments  |
| <pre>{   "Type":"Poll",   "CurrentCreditCash":"0",   "CurrentCreditCashless":"0",   "BillStat":"0B000B00",   "CoinStat":"00000200",   "CashlessStat":"FFFFFFFF",   "CashlessID":"FFFFFFFF",   "AvailableChange","6110" }</pre> | <p>This command will return the current MDB module's information. It is recommended to issue this command at least twice a second while the transaction is opened (when the payment systems are enabled), to check the current credit and to disable the payment systems when the credit is the same or bigger than the selected product price.</p> <ul style="list-style-type: none"> <li>- &lt;CurrentCreditCash&gt; is the accumulated cash credit;</li> <li>- &lt;CurrentCreditCashless&gt; is the credit available on the customer's cashless payment media;</li> <li>- &lt;BillStat&gt; contains the last 4 hexadecimal bill validator status codes (according with the Table 2 on page 14). The leftmost value is the older one.</li> <li>- &lt;CoinStat&gt; contains the last 4 hexadecimal coin acceptor status codes (according with the Table 3 on page 15). The leftmost value is the older one.</li> <li>- &lt;CaslessStat&gt; contains the last 4 hexadecimal cashless devices status codes (according with the Table 4 on page 16). The leftmost value is the older one.</li> <li>- &lt;CashlessID&gt; contains the internal ID of the customer's media. You can use this for tracking purposes.</li> </ul> <p>It is recommended to keep an eye on this informations during the transaction. When there is no transaction open it is recommended to periodically poll this status variable to detect payment systems jam.</p> <p>If a payment system was not initialized, it's corresponding status variable will have "FFFFFFFF" value.</p> <ul style="list-style-type: none"> <li>- &lt;AvailableChange&gt; is the total value of the coins in the coing changer. This value is updated on every coin enable or coin disable command</li> </ul> |

## 14. MDBResetStatus

| GUI command  |  |
|--|--|
| Command  | Parameters/Comments  |
| MDBResetStatus   | [none]   |
| RASPIVEND daemon answer  |  |
| Answer   | Parameters/Comments  |
| <pre>{"MDBResetStatus":"success"} or {"MDBResetStatus":"failed"}</pre> | <p>This command will set to "00000000" all MDB payment system status variables.</p> <p>If a payment system was not initialized, it's corresponding status variable will have "FFFFFFFF" value.</p> |



## 15. MDBCashlessInit

| GUI command   |   |
|---|---|
| Command   | Parameters/Comments   |
| MDBCashlessInit(N)  | "N" is the address of the cashless to be initialized (1 for the first cashless device and 2 for the second cashless device)   |
| RASPIVEND daemon answer   |   |
| Answer  | Parameters/Comments   |
| {"MDBCashlessInit": "success"}<br>or<br>{"MDBCashlessInit": "failed"} | This command will perform all initialization tasks for the attached MDB cashless system. If something goes wrong or the MDB cashless system is not connected to the board, then the command returns "failed" message. |

## 16. MDBCashlessEnable

| GUI command   |  |
|---|--|
| Command   | Parameters/Comments  |
| MDBCashlessEnable(N)  | "N" is the address of the cashless to be enabled (1 for the first cashless device and 2 for the second cashless device)  |
| RASPIVEND daemon answer   |  |
| Answer  | Parameters/Comments  |
| {"MDBCashlessEnable": "success"}<br>or<br>{"MDBCashlessEnable": "failed"} | This command will activate the attached MDB cashless system. This command must be preceded by the MDBCashlessInit command. You cannot activate a cashless system if this one is not initialized. |

## 17. MDBCashlessDisable

| GUI command   |  |
|---|--|
| Command   | Parameters/Comments  |
| MDBCashlessDisable(N)   | "N" is the address of the cashless to be disabled (1 for the first cashless device and 2 for the second cashless device) |
| RASPIVEND daemon answer   |  |
| Answer  | Parameters/Comments  |
| {"MDBCashlessDisable": "success"}<br>or<br>{"MDBCashlessDisable": "failed"} | This command will deactivate the attached MDB cashless system.   |

## 18. MDBCashlessSettings

| GUI command   |   |
|---|---|
| Command   | Parameters/Comments   |
| MDBCashlessSettings(N)  | "N" is the address of the cashless to be readed (1 for the first cashless device and 2 for the second cashless device)  |
| RASPIVEND daemon answer   |   |
| Answer  | Parameters/Comments   |
| <pre>{   "Type": "Cashless",   "level": "2",   "CurrencyCode": "1978",   "ScalingFactor": "1",   "DecimalPlaces": "2",   "CanRevalue": "True",   "Manufacturer": "COM",   "SerialNumber": "000000114761",   "Model": "NEW_EUROKEY ",   "SWVersion": "513" }</pre> | This command will return the specified cashless device low level informations. You will probably don't need those informations, unless you want to track the payment systems for service/maintenance reasons. |

## 19. MDBCashlessVendRequest

| GUI command                         |  |
|-------------------------------------|--|
| Command                             | Parameters/Comments  |
| MDBCashlessVendRequest(AAA,BBB,CCC) | Asks the cashless AAA to approve a sale for the BBB value, representing the price of the product number CCC<br>For example, MDBCashlessVendRequest(1,400,12) means that we are asking the cashless device number 1 to approve a sale of 400 (EUR4.00) for the product number 12. |
| RASPIVEND daemon answer             |  |
| Answer                              | Parameters/Comments  |
| {"MDBCashlessVendRequest": "100"}   | This command will return the value you have sent. For safety reasons (eliminate the communication errors), you must commit this action with MDBCashlessVendRequestOK, after you verify that the returned value is the same with the sent value.                                  |

## 20. MDBCashlessVendSuccess

| GUI command   |  |
|---|--|
| Command   | Parameters/Comments  |
| MDBCashlessVendSuccess(AAA,BBB)   | The vending machine is reporting to the AAA cashless device that it successfully dispensed the product number BBB<br>For example, MDBCashlessVendSuccess(1,8) is a response for cashless device number 1 that the product number 8 was successfully dispensed to the customer. |
| RASPIVEND daemon answer   |  |
| Answer  | Parameters/Comments  |
| { "MDBCashlessVendSuccess": "success" }<br>or<br>{ "MDBCashlessVendSuccess": "failed" } | This command will send transaction success information to the cashless system.<br>You must use the MDBPoll command to verify the response of the cashless system according to Table 4 on page 16).   |

## 21. MDBCashlessVendFailed

| GUI command   |   |
|---|---|
| Command   | Parameters/Comments   |
| MDBCashlessVendFailed(N)  | "N" is the address of the cashless to notify that the transaction has failed (1 for the first cashless device and 2 for the second cashless device)   |
| RASPIVEND daemon answer   |   |
| Answer  | Parameters/Comments   |
| { "MDBCashlessVendFailed": "success" }<br>or<br>{ "MDBCashlessVendFailed": "failed" } | This command will send transaction failed information to the cashless system.<br>You must use the MDBPoll command to verify the response of the cashless system according to Table 4 on page 16). |

## 22. MDBCashlessRevalue

| GUI command   |  |
|---|--|
| Command   | Parameters/Comments  |
| MDBCashlessRevalue(N)   | "N" is the address of the cashless to revalue (1 for the first cashless device and 2 for the second cashless device)   |
| RASPIVEND daemon answer   |  |
| Answer  | Parameters/Comments  |
| { "MDBCashlessRevalue": "success" }<br>or<br>{ "MDBCashlessRevalue": "failed" } | This command will send load the CurentCreditCash value to the cashless.<br>You must use the MDBPoll command to verify the response of the cashless system according to Table 4 on page 16).<br>Also, your application must handle the maximum revalue settings for the cashless system. You have to set the cashless revalue only in it's aloud range. If you will not manage this, then the cashless can randomly goes to overflow. |

## 23. MDBCashlessEndSession

| GUI command   |   |
|---|---|
| Command   | Parameters/Comments   |
| MDBCashlessEndSession(N)  | "N" is the address of the cashless to close the session (1 for the first cashless device and 2 for the second cashless device)  |
| RASPIVEND daemon answer   |   |
| Answer  | Parameters/Comments   |
| {"MDBCashlessEndSession":"success"}<br>or<br>{"MDBCashlessEndSession":"failed"} | This command will force the cashless system to close the current session. If the media is not removed, most of the time, the cashless system will automatically open a new session. |

# NOTES: