Interface MDB master – RS232
(DIRECT VERSION)
v19.04.2017
Quick reference

Table of Contents

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 to communicate with peripherals using two methods:

a. A low level communication method that can be used to send binary MDB messages over the serial port

b. A high level communication method that simplifies the user interface development, offering a language independent support.

A. Low level mode

In low level mode, the user application is responsible of all VMC logic and payment systems manipulation along with multiplexer handling.

B. High level mode

In high level mode, the user's app is connecting using sockets to the 5127 TCP port on localhost and sends some standard messages, described below.

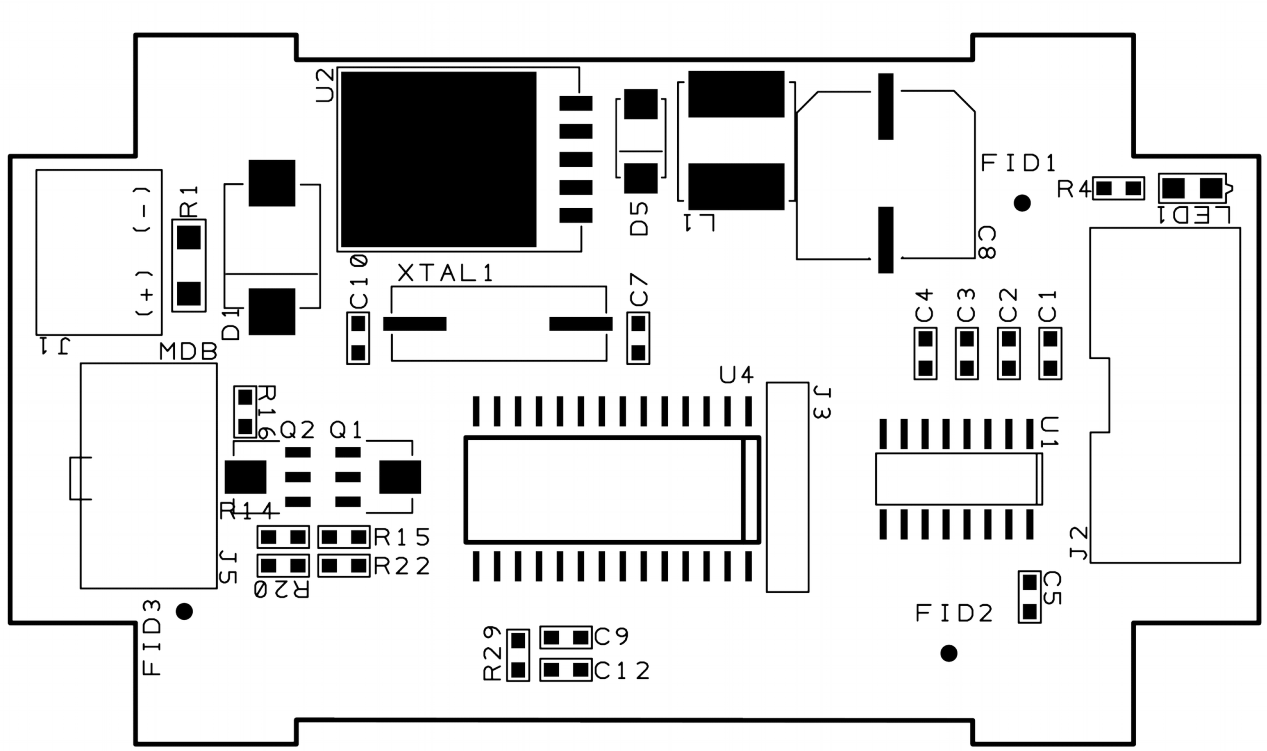
3. Communication parameters

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	115200
data bits	8
parity	NONE
hardware flow	YES (RTS/CTS)
software flow	NO

Table 1: 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. Low level mode

Using this mode, your application must handle all the low level MDB features, by sending binary MDB messages directly to interface. In this mode, your application must send to the serial port, all needed commands required by the desired MDB payment system that is connected. The interface accepts any MDB command (see NAMA MDB manual for this). Also, your application must calculate the correct CRC, according to the MDB protocol specifications. You can use our Python demo application to see the implementation model.

IV. High level mode with service daemon

Using this mode, the development becomes much easier. The communication with the board and the peripherals is managed by a small application. The application is available for download on product's page.

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.

1. Configuring Python 3

To use this mode, you need the following:

- install Python 3 on your computer (for Ubuntu/Debian "sudo apt-get install python3");
- install pip3 (for Ubuntu/Debian "sudo apt-get install python3-pip");
- install PySerial ("sudo pip3 install pyserial==3.0.1");
- download and run the Python script from our website.
- open a new console and run telnet on localhost, port 5126("sudo telnet localhost 5127")
- in the telnet window start sending commands to the device.

2. BillReset

GUI command	
Command	**Parameters/Comments**
BillReset	[none]
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBBIIIReset": 0} or {"MDBBIIIReset": -1}	This command will send the reset command to the bill validator. If the returned value is "0", the command were successfully executed and the validator responded with ACK. If the returned value is "-1", there was an error reseting the bill validator and the user's application should retry few times before aborting the operation. When you have a connected bill validator, this is the first command to send in the initialization process

3. BillInit

GUI command	
Command	**Parameters/Comments**
BillInit	[none]
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBBIllReset": 0} or {"MDBBIllReset": -1}	This command will perform the initialization procedure on the bill validator. The answers could be “0” - success or “-1” - failed. When you have a connected bill validator, this is the second command to send in the initialization process

4. BillSettings

GUI command	
Command	**Parameters/Comments**
BillSettings	[none]
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBBillSettings": "Current", "Level": 2, "CountryCode": 1642, "ScalingFactor": 100, "StackerCappacity": 300, "EscrowAvailable": true, "BillValues":[1,5,10,0,0,0,0,0,0,0,0,0,0,0,0,0], "Manufacturer": "ITL", "SerialNumber": "000000271269", "Model": "BV0100 000", "SoftwareVersion": "0414", "RecyclingAvaliable": false}	This command will return all validator’s settings When you have a connected bill validator, this is the third command to send in the initialization process. This command is mandatory, otherwise the application will not be able to perform some calculations (for example, bill values) because it has not enough informations (for example, scaling factor or decimal places, plus the accepted bills values)

5. BillStacker

GUI command	
Command	**Parameters/Comments**
BillStacker	[none]
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBBIllStacker": 38,"StackerFull": false} or {"MDBBIllStacker": -1} if not succeeded	This command will return the number of the bills in stacker (if the bill validator has a stacker and it will return “true” if the stacker is full or “false” if the stacker is not full, yet. Please note that this function depends on the bill validator and some of them may always return 0 and false.

6. BillEnable

GUI command	
Command	**Parameters/Comments**
BillEnable	[none]
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBBIIIEnable": 0} – for success or {"MDBBIIIEnable": -1} – for failure	This command will enable the attached MDB bill validator. The bill validator will accept all denominations that are supported by it's internal firmware.

7. BillDisable

GUI command	
Command	**Parameters/Comments**
BillDisable	[none]
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBBIIIDisable": 0} – for success or {"MDBBIIIDisable": -1} – for failure	This command will disable the attached MDB bill validator. It will no longer accept any of the denominations that are supported by it's firmware

8. BillAccept

GUI command	
Command	**Parameters/Comments**
BillAccept	[none]
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBBIIIAcceptBillInEscrow": 0} – response if the command was accepted by the bill validator. After this response, the bill will process the bill and try to stack it. When the bill is correctly stacked, the interface will send a second message (unsolicited message): {"BillStacked": 0,"BillValue": 100} that will show the bill is safely deposited in stacker or inside the vending machine (if the bill validator is stackerless). This is the message that should be used to increment the current credit. {"MDBBIIIAcceptBillInEscrow": -1} - if the command failed to reach the bill validator	This command can be used only when the bill validator has escrow capability (it gets the bill, recognizes it and keep it in escrow position sending a message about this) The interface will send a message like: {"BillInEscrow": 0,"BillValue": 100} to notify the bill number and the bill value (according to the scaling factor) waiting to accept or reject.

9. BillReject

GUI command	
Command	**Parameters/Comments**
BillReject	[none]
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBBIllRejectBillInEscrow": 0} – response if the command is accepted by the bill validator. When the bill is turned to the customer, the interface will send a second message (unsolicited message): {"BillReturned": 0,"BillValue": 100} that will show the bill is correctly returned to the customer. {"MDBBIllRejectBillInEscrow": -1} - if the command failed to reach the bill validator	This command can be used only when the bill validator has escrow capability (it gets the bill, recognizes it and keep it in escrow position sending a message about this) The interface will send a message like: {"BillInEscrow": 0,"BillValue": 100} to notify the bill number and the bill value (according to the scaling factor) waiting to accept or reject.

10. CoinReset

GUI command	
Command	**Parameters/Comments**
CoinReset	
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCoinReset": 0} – if success or {"MDBCoinReset": -1} – if failed	This command will send the reset command to the coin acceptor. If the returned value is “0”, the command were successfully executed and the coin acceptor responded with ACK. If the returned value is “-1”, there was an error reseting the coin acceptor and the user’s application should retry few times before aborting the operation. When you have a connected coin acceptor, this is the first command to send in the initialization process

11. CoinInit

GUI command	
Command	**Parameters/Comments**
CoinInit	
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCoinReset": 0} – if success or {"MDBCoinReset": -1} – if failed	This command will perform the initialization procedure on the coin acceptor. The answers could be “0” - success or “-1” - failed. When you have a connected bill validator, this is the second command to send in the initialization process

12. CoinSettings

GUI command	
Command	**Parameters/Comments**
CoinSettings	
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCoinSettings": "Current", "Level": 3, "CountryCode": 1642, "ScalingFactor": 5, "DecimalPlaces": 2, "CoinRoutingChannel": [0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0], "CoinValues": [1,2,10,255,0,0,0,0,0,0,0,0,0,0,0,0], "Manufacturer": "MEI", "SerialNumber": "2378G802863 ", "Model": "CF7900MDB ", "SoftwareVersion": "0118", "AlternativePayout": true}	This command will return all coin acceptor’s settings. When you have a connected coin acceptor, this is the third command to send in the initialization process. This command is mandatory, otherwise the application will not be able to perform some calculations (for example, coin values) because it has not enough informations (for example, scaling factor or decimal places, plus the accepted coin values)

13. CoinEnable

GUI command	
Command	**Parameters/Comments**
CoinEnable	[none]
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCoinEnable": 0} – for success or {"MDBCoinEnable": -1} – for failure	This command will enable the attached MDB coin acceptor. The coin aceptor will accept all denominations that are supported by it’s internal firmware.

14. CoinDisable

GUI command	
Command	**Parameters/Comments**
CoinDisable	[none]
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCoinEnable": 0} – for success or {"MDBCoinEnable": -1} – for failure	This command will disable the attached MDB coin aceptor. It will no longer accept any of the denominations that are supported by it's firmware.

15. CoinTubeStatus

GUI command	
Command	**Parameters/Comments**
CoinTubeStatus	[none]
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCoinTubeStatus": 13240} –for success (in this example, the total available change is EUR 132.40 or {"MDBCoinTubeStatus": -1} – for failure	This comand will return scaled value of the total change available in tubes, that can be used to return change after transactions

16. CoinChange(NNN)

GUI command	
Command	**Parameters/Comments**
CoinChange(NNN)	"NNN" is the total value of the change that should be returned to the customer. For example, CoinChange(130) will return EUR 1.30
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCoinChange": 0} – if success This response may be followed by one or more unsolicited messages (depending on the coin acceptor MDB implementation). For example, it can notify that the changer is busy returning change, by: {"CoinStatus": "ChangerPayoutBusy", "CoinStatusCode" : 2} followed by: {"CoinStatus": "OK","CoinStatusCode" : 0} when the changer finished the action.	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. During this command you have to poll the payout status by issuing the CoinPayStatus command (see details on 17.)

17. CoinPayStatus

GUI command	
Command	**Parameters/Comments**
CoinPayStatus	
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCoinChangeStatus": 130} – for success (this example means that EUR1.30 were ejected). or {"MDBCoinChangeStatus": -1} – if failure	This command will return the total value of the ejected coins until the command is issued.

18. CashlessReset(N)

GUI command	
Command	**Parameters/Comments**
CashlessReset(1)	This command will perform a reset of the cashless device number “N”, where “N” can be 1 or 2, depending on your cashless settings.
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCashlessReset": 0} – on success or {"MDBCashlessReset": -1} – on failure	

19. CashlessInit(N)

GUI command	
Command	**Parameters/Comments**
CashlessInit(1)	This command will initialize the cashless number “N” where “N” can be 1 or 2, depending on your cashless settings
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCashlessInit": 0} – on success or {"MDBCashlessInit": -1} – on failure	

20. CashlessSettings(N)

GUI command	
Command	**Parameters/Comments**
CashlessSettings(N)	“N” is the cashless number (as deschibed above)
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"CashlessLevel": 2, "CashlessCountryCode": 1978, "CashlessScalingFactor": 1, "CashlessDecimalPlaces": 2, "CashlessMaxResponseTime": 7, "CashlessCanRevalue": true, "CashlessCanMultivend": true, "CashlessHasDisplay": false, "CashlessCanCashSale": false, "CashlessManufacturer": "COM", "CashlessSerialNumber": "000000114761", "CashlessModelNumber": "NEW_EUROKEY ", "CashlessSoftwareVersion": 0201}	This command will return all cashless device settings. When you have a connected cashless device, this is the third command to send in the initialization process. This command is mandatory, otherwise the application will not be able to perform some calculations (for example, credit value) because it has not enough informations (for example, scaling factor or decimal places). You may extract some informations here to avoid sending commands that are not supported by the cashless device (for example, revalue if the cashless device does not support it).

21. CashlessEnable(N)

GUI command	
Command	**Parameters/Comments**
CashlessEnable(N)	This command will enable the cashless device number “N”, where “N” can be 1 or 2
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCashlessEnable": 0} – on success or {"MDBCashlessEnable": -1} – on failure	After the cashless is enabled, by using payment media, there could be some unsolicited messages (when a customer swipes the card or inserts the card, toke, etc) {"CashlessNumber": 1, "CashlessStatus": "ReaderBeginSession", "CashlessStatusCode": 3, "CashlessFundsAvailable": 800, "CashlessMediaPaymentId": "0x00 0x53 0x44 0x16", "CashlessPaymentType": "NormalVendCard"} In this example, an RFID kay with EUR8.00 credit was inserted in the reader. The key ID is 0x00 0x53 0x44 0x16 (every payment media has a 4 byte unique ID).

22. CashlessDisable(N)

GUI command	
Command	**Parameters/Comments**
MDBCashlessVendFailed(N)	This command will disable the cashless device number “N”, where “N” can be 1 or 2
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCashlessDisable": 0} – on success or {"MDBCashlessDisable": -1} – on failure	

23. CashlessVendRequest(AAA,BBB,CCC)

GUI command	
Command	**Parameters/Comments**
CashlessVendRequest(AAA,BBB,CCC)	This command can be used only when a session was opened by the cashless device and some credit is reported to the interface - AAA – cashless number (1 or 2); - BBB – product price – the requested product price, scalled by scaling factor) - CCC – product number (for example, 8 is product number 8 on the machine)
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCashlessVendRequest": 0} – on success. Other unsolicited messages will appear after this response: {"CashlessNumber": 1, "CashlessStatus": "VendApproved", "CashlessStatusCode": 5, "ApprovedValue": 100} – this means that the cashless device approved the vend and the VMC should dispense the product. The reader will wait for vend result (success or failure) after the transaction is finished Also, the cashless device could respond with the following unsolicited message: {"CashlessNumber": 1, "CashlessStatus": "VendDenied", "CashlessStatusCode": 6} – if the funds ar insufficient for the selected product or for other reasons. {"MDBCashlessVendRequest": -1} – if the command cannot reach the cashless device.	

24. CashlessNegativeVendRequest(AAA,BBB, CCC)

GUI command	
Command	**Parameters/Comments**
CashlessNegativeVendRequest(AAA,BBB, CCC)	This command can be used only when a session was opened by the cashless device and some credit is reported to the interface - AAA – cashless number (1 or 2); - BBB – product price – the requested product price, scalled by scaling factor) - CCC – product number (for example, 8 is product number 8 on the machine). This will add BBB value on the payment media (card, token, etc.) if the vend success is reported by the machine. Do not use this to add credit on the payment media, but only for negative vend transactions (for example, recycling vending machine) and only for cashless payment systems that are supporting negative vend. Use revalue (see below) to add credit on the payment media.
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCashlessNegativeVendRequest": 0} – on success. Other unsolicited messages will appear after this response: {"CashlessNumber": 1, "CashlessStatus": "VendApproved", "CashlessStatusCode": 5, "ApprovedValue": 100} – this means that the cashless device approved the vend and the VMC should dispense the product. The reader will wait for vend result (success or failure) after the transaction is finished Also, the cashless device could respond with the following unsolicited message: {"CashlessNumber": 1, "CashlessStatus": "VendDenied", "CashlessStatusCode": 6} – if the funds ar insufficient for the selected product or for other reasons. {"MDBCashlessNegativeVendRequest": -1} – if the command cannot reach the cashless device.	

25. CashlessVendSuccess(AAA,BBB)

GUI command	
Command	**Parameters/Comments**
CashlessVendSuccess(AAA,BBB)	This command will confirm that the transaction is successfully finished on the vending machine. Depending on it's implementation, this is the moment that the cashless device will commit the credit withdrawal. - AAA – the cashless number - BBB – the product number (selection number) dispensed to the customer
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCashlessVendSuccess": 0} – on success or {"MDBCashlessVendSuccess": -1} – on failure	

26. CashlessVendFailed(N)

GUI command	
Command	**Parameters/Comments**
CashlessVendFailed(N)	If the machine fails to dispense the product, then will send this command to the cashless device, to avoid taking the money from the customer's payment media. - "N" is the cashless number
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCashlessVendFailed": 0} – on success or {"MDBCashlessVendFailed": -1} – on failure (cashless payment system did not received the command)	

27. CashlessSessionComplete(N)

GUI command	
Command	**Parameters/Comments**
CashlessSessionComplete(N)	This command will instruct the cashless device “N” to end current session. Eventually, if the payment media was not removed before, the cashless device will open another session. It is a good option to send this command after each finished transaction, to check the cashless available credit.
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCashlessSessionComplete": 0} – on success. Depending on the cashless MDB implementation, it will also respond with an unsolicited message of status: {"CashlessNumber": 1, "CashlessStatus": "EndSession", "CashlessStatusCode": 7} or {"MDBCashlessSessionComplete": -1} – on failure	If the payment media was not removed before, some cashless payment systems will open a new session: {"CashlessNumber": 1, "CashlessStatus": "ReaderBeginSession", "CashlessStatusCode": 3, "CashlessFundsAvailable": 600, "CashlessMediaPaymentId": "0x00 0x53 0x44 0x16", "CashlessPaymentType": "NormalVendCard"}

28. CashlessRevalueLimitRequest(N)

GUI command	
Command	**Parameters/Comments**
CashlessRevalueLimitRequest(N)	This command will ask the cashless for the maximum amount that will accept for revalue command (see revalue command below for details)
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"CashlessNumber": 1, "CashlessStatus": "LimitAmount", "CashlessStatusCode": 15, "LimitValue": 900} or {"MDBCashlessRevalueLimitRequest": -1} – on failure	The LimitValue will be the maximum value accepted by the cashless device on the next revalue command. If you try to revelue it with a higher value, then it will respond with a failure.

29. CashlessRevalue(AAA,BBB)

GUI command	
Command	**Parameters/Comments**
CashlessRevalue(AAA,BBB)	This command will recharge the customer's account, stored on payment media: - AAA – cashless number; - BBB – value to recharge. BBB should be <= than the LimitValue reported by the cashless device using command CashlessRevalueLimitRequest (see 28 for details)
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"CashlessNumber": 1, "CashlessStatus": "RevalueApproved", "CashlessStatusCode": 13} or {"CashlessNumber": 1,"CashlessStatus": "RevalueDenied","CashlessStatusCode" : 14} if the revalue is higher than the revalue limit reported by the cashless device or {"MDBCashlessRevalueLimitRequest": -1} – on failure communicating with cashless device.	

30. CashlessCashSale(AAA,BBB,CCC)

GUI command	
Command	**Parameters/Comments**
CashlessCashSale(AAA,BBB,CCC)	This command will report a cash only sale to the cashless device if this supports the option (see cashless settings command). This command is for statistic purposes, if the cashless device supports audit or sales reporting: - AAA – cashless number; - BBB – product price; - CCC – product number.
INTERFACE daemon answer	
Answer	**Parameters/Comments**
{"MDBCashlessCashSale": 0} – on success or {"MDBCashlessCashSale": -1} – on faliure communicating with cashless device	

31. MDBSendRaw(A,B,C,D,…,N)

GUI command	
Command	**Parameters/Comments**
MDBSendRaw(A,B,C,D,…,N)	This command offers the possibility to the user's application to send any other desired command to the MDB bus. The last byte should be the CRC calculated by the MDB rule (see MDB manual for details). Bytes value could be sent in decimal format (0-255) or in hexadecimal format (0x00-0xFF), even mixed (decimal and hexadecimal in the same message)
INTERFACE daemon answer	
Answer	**Parameters/Comments**
MDB response	The response will vary, depending on the sent command, according to the MDB manual. The interface will only convert 8 bit to 9 bit and back to 8 bit and will ACK/NACK in the 5ms interval required, according to the MDB specifications.

32. MDBSendRawCRC(A,B,C,D,…,N)

GUI command	
Command	**Parameters/Comments**
MDBSendRaw(A,B,C,D,…,N)	This command offers the possibility to the user's application to send any other desired command to the MDB bus. The users is sending the message bytes only, and the CRC will be calculated and added by the daemon. Bytes values could be sent in decimal format (0-255) or in hexadecimal format (0x00-0xFF), even mixed (decimal and hexadecimal in the same message)
INTERFACE daemon answer	
Answer	**Parameters/Comments**
MDB response	The response will vary, depending on the sent command, according to the MDB manual. The interface will only convert 8 bit to 9 bit and back to 8 bit and will ACK/NACK in the 5ms interval required, according to the MDB specifications.

NOTES: