

**PICOBRIDGE RS232  
(MDB master mode)  
v06.05.2018**

# Table of Contents

I. General informations.....	4
1. Terms.....	4
2. Working modes.....	4
A. Low level mode.....	4
B. High level mode.....	4
3. Low level communication parameters.....	4
II. Hardware overview.....	5
1. Power supply requirements.....	5
2. Connectors description.....	5
III. Low level mode.....	7
IV. High level mode with service daemon.....	8
1. Configuring Python 3.....	8
2. BillReset.....	8
3. BillInit.....	9
4. BillSettings.....	9
5. BillStacker.....	9
6. BillEnable.....	9
7. BillDisable.....	10
8. BillAccept.....	10
9. BillReject.....	10
10. BillTimeout.....	11
11. CoinReset.....	11
12. CoinInit.....	12
13. CoinSettings.....	12
14. CoinEnable.....	12
15. CoinDisable.....	13
16. CoinTubeStatus.....	14
17. CoinChange(NNN).....	14
18. CoinPayStatus.....	14
19. CoinTimeout.....	15
20. CashlessReset(N).....	15
21. CashlessInit(N).....	15
22. CashlessSettings(N).....	16
23. CashlessEnable(N).....	16
24. CashlessDisable(N).....	17
25. CashlessVendRequest(AAA,BBB,CCC).....	17
26. CashlessNegativeVendRequest(AAA,BBB, CCC).....	18
27. CashlessVendSuccess(AAA,BBB).....	19
28. CashlessVendFailed(N).....	19
29. CashlessSessionComplete(N).....	20
30. CashlessRevalueLimitRequest(N).....	20
31. CashlessRevalue(AAA,BBB).....	21
32. CashlessCashSale(AAA,BBB,CCC).....	21
33. MDBSendRaw(A,B,C,D,...,N).....	22
34. MDBSendRawCRC(A,B,C,D,...,N).....	23
35. RTCSet.....	23
36. RTCGet.....	24



# I. General informations

## 1. Terms

- **MDB PERIPHERALS** = payment systems connected on the MDB bus.
- **HOST APPLICATION** = the Python 3 daemon
- **CLIENT APPLICATION** = the client application that will connect to the socket of the HOST APPLICATION
- **ACK** = acknowledge
- **NACK** = not-acknowledge

## 2. Working modes

The PICOBRIDGE RS232 board can be used to communicate with peripherals using two methods:

- a. A low level communication method
- 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. There are no access to multiplexer pins and only implemented and describe functions are available.

## 3. Low level communication parameters

The communication settings should meet the following specifications:

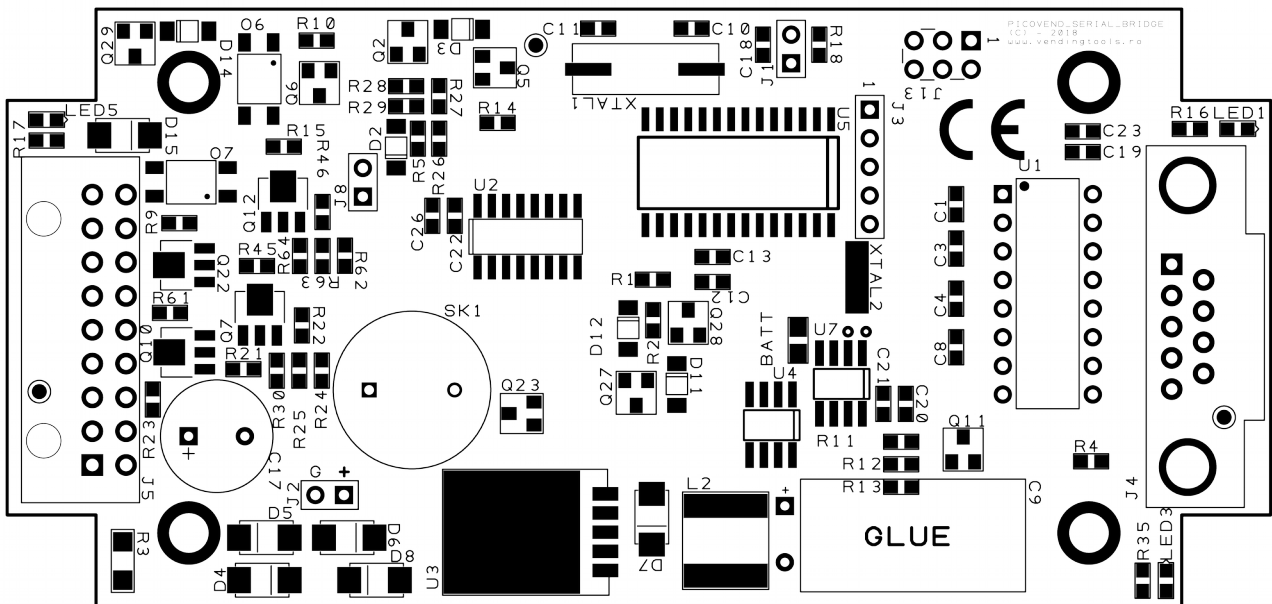
- a. For the peripherals (excepting the MDB bus), there is no restriction regarding the serial port settings you need.
- b. For then MDB communication parameters:

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

*Table 3: MDB communication parameters*

## II. Hardware overview

Picture 1: Board overview



### 1. Power supply requirements

The PICOBRIDGE RS232 can be powered with stabilized 24VDC/AC or 12VDC/AC, depending on your MDB PERIPHERALS.

**NOTE: If you apply 24AC, please make sure that your MDB peripherals can support 34VDC input. Otherwise, use a DC power supply.**

### 2. Connectors description

J5 – see table below

Pin no.	Pin description	Function details
1	AC power IN	AC power IN – only for Executive bridge and MDB master mode
2	AC power IN	AC power IN – only for Executive bridge and MDB master mode
3	24VDC power IN/OUT	MDB slave power input
4	GND	GND
5	GND	Signal GND for MDB master mode
6	Master RX	MDB master mode RX
7	GND	GND
8	Master TX	MDB master mode TX
9	GND	GND
10	Executive RX	Executive to MDB bridge mode – Executive RX
11	GND	GND
12	Executive TX	Executive to MDB bridge mode – Executive TX

Pin no.	Pin description	Function details
13	GND	GND
14	MDB slave TX	MDB slave mode TX
15	GND	GND
16	MDB slave RX	MDB slave mode RX
17	Signal GND	MDB slave mode signal GND
18	Master wake	Not implemented in this firmware version (battery mode MDB payment systems wake signal)

J1 – working mode change jumper

J8 – when installed, enables MDB sniffer

J4 – RS232 DB9 female connector. You need a straight RS232 cable or you can connect an USB to RS232 cable. Please make sure your USB to RS232 cable/converter can correctly handle hardware flow (RTS/CTS). Best results were obtained using FTDI based cables/converters and PL2303 based cables/converters.

You do not need to perform any settings on the PICOBRIDGE RS232, neither hardware or software. You only have to select master working mode if not already selected when you have ordered the product.

### III. Low level mode

To use this mode, your application must handle the following:

a. Serial port

b. For MDB handling you need, also, to manipulate MDB RTS and MDB CTS pins.

Please check our demo application (Python 3 daemon) to see the correct RTS/CTS manipulation.

The user is responsible to send any correct MDB message to the interface that will translate it to 9bit, handling the mode bit. Also, any answer from the payment systems will be translated to 8bit format and sent back to the serial port.

For settings RTC, you need to send the following message:

RTC set - <0xFE> <0x01> <ss> <mm> <hh> <dow> <dd> <MM> <yy> <CRC>

Device answer will be 0xFC 0xFC 0xFC 0xFC 0xFC 0xFC on success and timeout on failure.

RTC get - <0xFE> <0x02> <CRC>

Device answer will be in the following format:

<0xFE> <0x02> <ss> <mm> <hh> <dow> <dd> <MM> <yy> <CRC>

All parameters on RTC set and get are BCD (please check the DS1307 IC manual on page 8 to find registers' interpretation).

CRC calculation follows the standard MDB rules for both MDB messages and RTC manipulation messages.

Your application is responsible to send correct MDB messages and to parse all the answers. Also, your application is responsible to send messages in the required order for the addressed peripheral.

## 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 selective capitalization to facilitate reading.

It is a good idea for your application to retry sending the command few times if you get a "failed" answer. This answer can be returned in the event of PICOBRIDGE RS232 board communication failure. Also, it can be returned if you try to address a not connected device. Also, to successfully run the application, you have to copy it into a directory and decompress the archive, keeping the shared libraries in the same place with the main application. Also, it requires some Python 3 libraries, so it is highly recommended to install Python 3 on your computer.

### 1. Configuring Python 3

To use this mode, you need the following:

- install Python 3 on your computer;
- install pip3 on your computer;
- install PySerial;
- download and run the Python script from our website (using python3 interpreter);
- open a new console and run telnet on localhost, port 5127
- in the telnet window start sending commands to the device.

### 2. BillReset

GUI command	
Command	Parameters/Comments
BillReset	[none]
PICOBRIDGE RS232 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 resetting 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]
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
<pre>{ "MDBBillReset": 0 } or { "MDBBillReset": -1 }</pre>	<p>This command will perform the initialization procedure on the bill validator. The answers could be “0” - success or “-1” - failed.</p> <p>When you have a connected bill validator, this is the second command to send in the initialization process</p>

### 4. BillSettings

GUI command	
Command	Parameters/Comments
BillSettings	[none]
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
<pre>{ "MDBBillSettings": "Current",   "Level": 2,   "CountryCode": 1642,   "ScalingFactor": 100,   "StackerCapacity": 300,   "EscrowAvailable": true,   "BillValues": [1,5,10,0,0,0,0,0,0,0,0,0,0,0,0],   "Manufacturer": "ITL",   "SerialNumber": "000000271269",   "Model": "BV0100 000",   "SoftwareVersion": "0414",   "RecyclingAvailable": false }</pre>	<p>This command will return all validator’s settings</p> <p>When you have a connected bill validator, this is the third command to send in the initialization process.</p> <p>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)</p>

### 5. BillStacker

GUI command	
Command	Parameters/Comments
BillStacker	[none]
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
<pre>{ "MDBBillStacker": 38, "StackerFull": false } or { "MDBBillStacker": -1 } if not succeeded</pre>	<p>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.</p>

### 6. BillEnable

GUI command	
Command	Parameters/Comments
BillEnable	[none]
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
{"MDBBillEnable": 0} – for success or {"MDBBillEnable": -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]
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
{"MDBBillDisable": 0} – for success or {"MDBBillDisable": -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]
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
{"MDBBillAcceptBillInEscrow": 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. {"MDBBillAcceptBillInEscrow": -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]
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
<p>{ "MDBBillRejectBillInEscrow": 0 } – response if the command is accepted by the bill validator.</p> <p>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.</p> <p>{ "MDBBillRejectBillInEscrow": -1 } - if the command failed to reach the bill validator</p>	<p>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)</p> <p>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.</p>

## 10. BillTimeout

GUI command	
Command	Parameters/Comments
BilTimeout(N)	This sets the application timeout value (in seconds), expecting answers from bill validator. Some bill validators are non-compliant on some commands and require a bigger timeout (for example on the init phase). Default value for "N" is 0.001
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
<p>{ "MDBBillTimeout": 0 } – if success  or  { "MDBBillTimeout": -1 } – if failed</p>	If the returned value is "0", the command were successfully executed. If the returned value is "-1", there was an error

## 11. CoinReset

GUI command	
Command	Parameters/Comments
CoinReset	
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
<p>{ "MDBCoinReset": 0 } – if success  or  { "MDBCoinReset": -1 } – if failed</p>	<p>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 resetting 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</p>

## 12. CoinInit

GUI command	
Command	Parameters/Comments
CoinInit	
PICOBRIDGE RS232 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

## 13. CoinSettings

GUI command	
Command	Parameters/Comments
CoinSettings	
PICOBRIDGE RS232 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)

## 14. CoinEnable

GUI command	
Command	Parameters/Comments
CoinEnable	[none]
PICOBRIDGE RS232 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 acceptor will accept all denominations that are supported by it's internal firmware.

## 15. CoinDisable

GUI command	
Command	Parameters/Comments
CoinDisable	[none]
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
{"MDBCoinEnable": 0} – for success or {"MDBCoinEnable": -1} – for failure	This command will disable the attached MDB coin acceptor. It will no longer accept any of the denominations that are supported by it's firmware.

## 16. CoinTubeStatus

GUI command	
Command	Parameters/Comments
CoinTubeStatus	[none]
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
<pre> {"MDBCoinTubeStatus": 13240} –for success (in this example, the total available change is EUR 132.40 or {"MDBCoinTubeStatus": -1} – for failure                     </pre>	<p>This comand will return scaled value of the total change available in tubes, that can be used to return change after transactions</p> <p>Please note that most of the changers will return an approximative value (depending on it's settings and sensor status). Use this information only for low level alarm and don;t use it for exact coin stock, nor change counting.</p>

## 17. 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
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
<pre> {"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.                     </pre>	<p>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.</p> <p>During this command you have to poll the payout status by issuing the CoinPayStatus command (see details on 17.)</p>

## 18. CoinPayStatus

GUI command	
Command	Parameters/Comments
CoinPayStatus	
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
<pre> {"MDBCoinChangeStatus": 130} – for success (this example means that EUR1.30 were ejected). or {"MDBCoinChangeStatus": -1} – if failure                     </pre>	<p>This command will return the total value of the ejected coins until the command is issued.</p>

## 19. CoinTimeout

GUI command	
Command	Parameters/Comments
CoinTimeout(N)	This sets the application timeout value (in seconds), expecting answers from coin acceptor. Some coin acceptors are non-compliant on some commands and require a bigger timeout (for example on the init phase). Default value for "N" is 0.001
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
{"MDBCoinTimeout": 0} – if success or {"MDBCoinTimeout": -1} – if failed	If the returned value is "0", the command were successfully executed. If the returned value is "-1", there was an error.

## 20. 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.
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
{"MDBCashlessReset": 0} – on success or {"MDBCashlessReset": -1} – on failure	

## 21. 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
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
{"MDBCashlessInit": 0} – on success or {"MDBCashlessInit": -1} – on failure	

## 22. CashlessSettings(N)

GUI command	
Command	Parameters/Comments
CashlessSettings(N)	"N" is the cashless number (as described above)
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
<pre>{ "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} </pre>	<p>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.</p> <p>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).</p>

## 23. 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
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
<pre>{ "MDBCashlessEnable": 0} – on success or {"MDBCashlessEnable": -1} – on failure </pre>	<p>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, token, etc)</p> <pre>{ "CashlessNumber": 1, "CashlessStatus": "ReaderBeginSession", "CashlessStatusCode": 3, "CashlessFundsAvailable": 800, "CashlessMediaPaymentId": "0x00 0x53 0x44 0x16", "CashlessPaymentType": "NormalVendCard"} </pre> <p>In this example, an RFID key 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).</p>



## 24. 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
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
{ "MDBCashlessDisable": 0 } – on success or { "MDBCashlessDisable": -1 } – on failure	

## 25. 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, scaled by scaling factor) - CCC – product number (for example, 8 is product number 8 on the machine)
PICOBRIDGE RS232 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 are insufficient for the selected product or for other reasons.  { "MDBCashlessVendRequest": -1 } – if the command cannot reach the cashless device.	

## 26. CashlessNegativeVendRequest(AAA,BBB, CCC)

GUI command	
Command	Parameters/Comments
CashlessNegativeVendRequest(AAA,BBB, CCC)	<p>This command can be used only when a session was opened by the cashless device and some credit is reported to the interface</p> <ul style="list-style-type: none"> <li>- AAA – cashless number (1 or 2);</li> <li>- BBB – product price – the requested product price, scaled by scaling factor)</li> <li>- CCC – product number (for example, 8 is product number 8 on the machine).</li> </ul> <p>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.</p>
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
<p>{"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 are insufficient for the selected product or for other reasons.</p> <p>{"MDBCashlessNegativeVendRequest": -1} – if the command cannot reach the cashless device.</p>	

## 27. 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 its 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
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
{"MDBCashlessVendSuccess": 0} – on success or {"MDBCashlessVendSuccess": -1} – on failure	

## 28. 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
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
{"MDBCashlessVendFailed": 0} – on success or {"MDBCashlessVendFailed": -1} – on failure (cashless payment system did not received the command)	

## 29. 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.
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
<pre> {"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 </pre>	<p>If the payment media was not removed before, some cashless payment systems will open a new session:</p> <pre> {"CashlessNumber": 1, "CashlessStatus": "ReaderBeginSession", "CashlessStatusCode": 3, "CashlessFundsAvailable": 600, "CashlessMediaPaymentId": "0x00 0x53 0x44 0x16", "CashlessPaymentType": "NormalVendCard"} </pre>

## 30. 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)
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
<pre> {"CashlessNumber": 1, "CashlessStatus": "LimitAmount", "CashlessStatusCode": 15, "LimitValue": 900} or {"MDBCashlessRevalueLimitRequest": -1} – on failure </pre>	<p>The LimitValue will be the maximum value accepted by the cashless device on the next revalue command. If you try to revalue it with a higher value, then it will respond with a failure.</p>

### 31. 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)
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
<pre> {"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.</pre>	

### 32. 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.
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
<pre> {"MDBCashlessCashSale": 0} – on success or {"MDBCashlessCashSale": -1} – on failure communicating with cashless device</pre>	

### 33. 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)
PICOBRIDGE RS232 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.

### 34. 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)
PICOBRIDGE RS232 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.

### 35. RTCSet

GUI command	
Command	Parameters/Comments
RTCSet(hh,mm,ss,dd,MM,yy,dow)	<p>This command sets the internal RTC. Parameters are:</p> <ul style="list-style-type: none"> <li>- hh – hour (00→23) – device only supports 24h time format;</li> <li>- mm – minutes (00 → 59);</li> <li>- ss – seconds (00 → 59);</li> <li>- dd – day (00 →31);</li> <li>- MM – month (00 → 12);</li> <li>- yy – year (00 → 99);</li> <li>- dow – day of week (00 → 07 where 0 is for Sunday and 7 for Saturday)</li> </ul> <p>Example: RTCSet(11,51,00,21,3,18,3)            This commands sets the internal RTC to 11:51:00, March 21<sup>st</sup>, 2018 on a Wednesday.</p>
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
{“RTCSet”: 0} - success or {“RTCSet”: -1} - failed	

## 36. RTCGet

GUI command	
Command	Parameters/Comments
RTCGet	
PICOBRIDGE RS232 daemon answer	
Answer	Parameters/Comments
{"RTCGet": [hh,mm,ss,dd,MM,yy,dow]}	<p>This command gets the internal RTC timestamp</p> <ul style="list-style-type: none"><li>- hh – hour (00→23) – device only supports 24h time format;</li><li>- mm – minutes (00 → 59);</li><li>- ss – seconds (00 → 59);</li><li>- dd – day (00 →31);</li><li>- MM – month (00 → 12);</li><li>- yy – year (00 → 99);</li><li>- dow – day of week (00 → 07 where 0 is for Sunday and 7 for Saturday)</li></ul> <p>Example: {"RTCGet": [12,23,24,21,3,18,3]}</p> <p>This response means the time is 12:23:24 on March 21<sup>st</sup>, 2018 on a Wednesday</p>



# NOTES: