

# **PICOVEND BLUETOOTH MDB SLAVE**

**v1.0 – 15.02.2019**

# Table of Contents

I. Hardware description.....	4
II. Functionality.....	4
A. Connecting to the device using Microsoft Windows 10 operating system.....	4
1. Pairing the device over Bluetooth.....	4
2. Using the device over Bluetooth.....	5
B. Connecting to the device using Ubuntu or Raspbian (for Raspberry Pi).....	5
III. The low level protocol over Bluetooth.....	6
A. Reserved device commands.....	6
1. Get cashless status.....	6
2. Cashless internal reset.....	7
3. Cashless set country code.....	7
4. Cashless set options byte.....	7
5. Cashless set scaling factor.....	7
6. Cashless set decimal places.....	7
7. Cashless get RTC.....	8
8. Cashless set RTC.....	8
9. Cashless set poll interval.....	8
10. Cashless set MDB timings.....	8
11. Clear paired devices.....	9
B. MDB commands.....	9
1. Cashless begin session.....	10
2. Cashless cancel session request.....	10
3. Cashless vend approved.....	10
4. Cashless vend denied.....	10
5. Cashless end session.....	10
6. Cashless canceled.....	11
7. Cashless revalue approved.....	11
8. Cashless revalue denied.....	11
9. Cashless display request.....	11
C. Unsolicited messages.....	12
IV. The high level protocol over Bluetooth, using our demo application.....	13
A. Reserved device commands.....	15
1. Get cashless status.....	15
2. Cashless internal reset.....	15
3. Cashless set country code.....	16
4. Cashless set options byte.....	16
5. Cashless set scaling factor.....	16
6. Cashless set decimal places.....	16
7. Cashless get RTC.....	17
8. Cashless set RTC.....	17
9. Cashless set poll interval.....	17
10. Cashless set MDB timings.....	18
11. Clear paired devices.....	18
B. MDB commands.....	19
1. Cashless begin session.....	19
2. Cashless cancel session request.....	19
3. Cashless vend approved.....	20
4. Cashless vend denied.....	20
5. Cashless end session.....	20

6. Cashless canceled.....	20
7. Cashless revalue approved.....	21
8. Cashless revalue denied.....	21
9. Cashless display request.....	21
C. Unsolicited messages.....	22
Appendix 1 – Cashless stage codes.....	23

## I. Hardware description

The device was designed for cashless payment systems development. The main functions are:

- receiving 9bit MDB communication messages and converting them to 8bit messages;
- handling ACK/NAK in a timely manner (following MDB timings restrictions);
- handling MDB messages in a timely manner (following MDB timings restrictions);
- internally handling initialization phase.

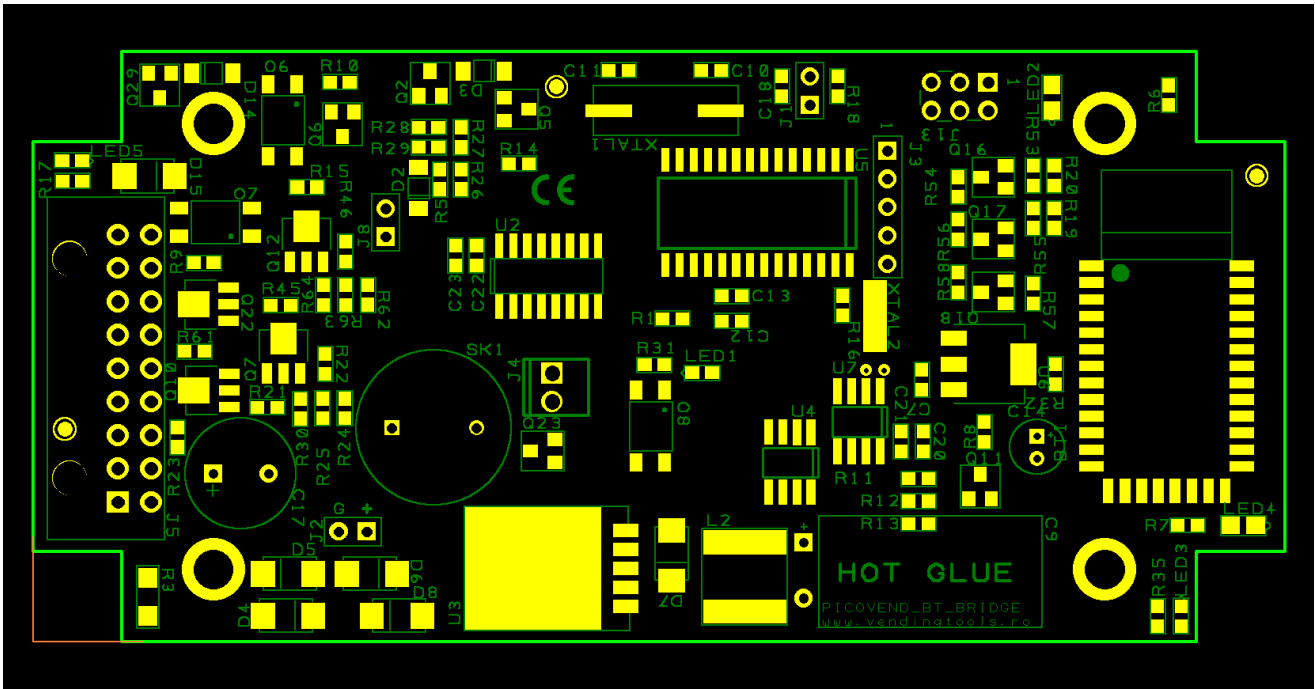


Image 1: Board overview

## II. Functionality

The device can be used to develop an MDB cashless system using a computer, a SBC (like Raspberry Pi, Orange Pi, etc.) or a phone/tablet.

This interface is emulating a Level 2 cashless device and it is enough to develop any cashless solution.

### A. Connecting to the device using Microsoft Windows 10 operating system

To connect the device on a computer using Microsoft Windows 10 operating system, you need to follow the procedure:

#### 1. Pairing the device over Bluetooth

- Power up the device, applying a voltage between 10 and 34VDC, depending on your MDB peripherals working voltage.
- Go to “Bluetooth and other devices setting”
- Choose “Add Bluetooth or other device”
- Choose “Bluetooth (mice, keyboard, pens, or audio and other kinds of Bluetooth device)”

- Wait for the the operating system to detect and identify the device. Sniffers Bluetooth IDs are respecting the following format: PVBTAAXXXX, where AA is the device family and XXXX is the device ID (both can be letters and numbers).
- Select the device, type the password on the device's label and wait for the operating system to install needed file (detection and installation are automatically performed, no software installation needed).
- After the finish message installation, check if the device is correctly installed (right click on "This PC" → click on "Manage" → click on "Device manager" → expand "Ports (COM & LPT)" and search for one or more records "Standard Serial over Bluetooth link (COMx or COMxx).

## 2. Using the device over Bluetooth

You can use some script capable serial communication application, or develop your own application, following low level device protocol handling.

## B. Connecting to the device using Ubuntu or Raspbian (for Raspberry Pi)

- Power up the device, applying a voltage between 10 and 34VDC, depending on your MDB peripherals working voltage.
- Open a terminal window and change user to root.
- Using your favorite editor, modify the file `/etc/systemd/system/dbus-org.bluez.service` as follows:
  - You need to have a line containing `ExecStart=/usr/lib/bluetooth/bluetoothd -C`
  - You need to have a line containing `ExecStartPost=/usr/bin/sdptool add SP`
- Save and reboot
- Open a terminal window and change user to root.
- Run the command `bluetoothctl`
- In `bluetoothctl` console, type the following command succession:
  - scan on
  - devices
  - agent on
  - pair XX:XX:XX:XX:XX:XX (where XX:XX:XX:XX:XX:XX is the hardware address of your device)
  - type the password you can find on device's label when asked to.
  - trust XX:XX:XX:XX:XX:XX
  - quit

In the console, type: `rfcomm bind /dev/rfcomm0 XX:XX:XX:XX:XX 1`

After that, you can use any serial terminal application to connect on `/dev/rfcomm0`

Please make sure you configure the serial port with 115200bps, 8 data bits, 1 stop bit, no parity and no hardware or software flow control. Also, please make sure you are selecting HEX data view.

### III. The low level protocol over Bluetooth

The low level protocol is the most convenient and flexible interface working mode, since it allows the developers to send virtually any command to the MDB bus.

The device is sending periodically a poll over the Bluetooth and after receiving the poll, the user application has 100ms to answer with a command for MDB bus. If the user application is not sending any message, then the device will send another poll message over the Bluetooth.

The poll message from the device is fixed length, with fixed content, as follows:  
 - 0x03 0xF0 0x50 0x53

First byte of messages received from the device represents the message length and is not included in CRC calculation procedure. Length is used to easily split concatenated messages. CRC is calculated following MDB CRC calculation rules.

Your application should wait for this message, then send any needed MDB command. There are, also some reserved commands, for device settings and registers manipulation. **ATTENTION!!!** - Any command sent later than 100ms will be ignored or will cause a NAK response from the device. You need to keep the commands prepared and send immediately after poll message received.

For this document, we will respect the following conventions:  
 - Bn = byte number n in message (starting from 0)  
 - bn = bit number n in a byte (starting from 0)

#### A. Reserved device commands

Any reserved command starts with message length, followed by 0xFE or 0xFB byte, depending on command type.

The device is answering to any command with a command ACK or NAK as follows:  
 - 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 - ACK  
 - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 - NAK

Last byte on commands and answers is the CRC and is calculated by the same procedure used for MDB.

#### 1. Get cashless status

Command from user app	
0xFB 0xFB	Get cashless status
Device answer	
0x1A 0xFB 0x08 0x01 0x02 0x00 0x28 0x09 0x02 0x00 0x02 0x10 0x02 0x01 0x30 0x31 0x38 0x45 0x34 0x33 0x35 0x43 0x37 0x45 0x30 0xE7	B0 - message length (length byte not included) B1 - always 0xFB B2 - cashless stage (described in Appendix 1) B3 - cashless scaling factor B4 - cashless decimal places B5 - cashless country code (HI byte) B6 - cashless country code (LO byte) B7 - cashless options byte B8 - cashless feature level B9 - cashless optional feature bits B10 - VMC feature level B11 - VMC columns on display B12 - VMC rows on display B13 - VMC display type info B14-B25 - cashless serial number B26 - MDB type calculated CRC from B1 to B25

## 2. Cashless internal reset

Command from user app	
0xFE 0x01 0xFF	Cashless will perform an internal reset and will report JUST RESET to the vending machine that should initialize it again.
Device answer	
- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 – ACK - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 – NAK	ACK comes after the command was successfully sent

## 3. Cashless set country code

Command from user app	
0xFE 0x02 <HI> <LO> <CRC>	Sets the cashless country code. Requires reset after value modification. <HI> - country code HI byte <LO> - country code LO byte <CRC> - MDB type calculated CRC from B1 to B25
Device answer	
- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 – ACK - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 – NAK	ACK comes after the command was successfully sent

## 4. Cashless set options byte

Command from user app	
0xFE 0x04 <OB><CRC>	Sets the cashless options byte. Requires reset after value modification. <OB> options byte according to MDB specifications: - b0 = if "1" then media reader is capable of restoring funds to media/account - b1 = if "1" then reader is multivend capable - b2 = N/A – always "0" - b3 = if "1" then reader is capable/willing to receive VEND/CASH SALE subcommand
Device answer	
- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 – ACK - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 – NAK	ACK comes after the command was successfully sent

## 5. Cashless set scaling factor

Command from user app	
0xFE 0x03 <SF><CRC>	Sets the cashless scaling factor. Requires reset after value modification. <SF> byte is the scaling factor that the cashless device will report to VMC after reset.
Device answer	
- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 – ACK - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 – NAK	ACK comes after the command was successfully sent

## 6. Cashless set decimal places

Command from user app	
0xFE 0x05 <DP><CRC>	Sets the cashless decimal places Requires reset after value modification. <DP> byte is the decimal places value that the cashless device will report to VMC after reset.
Device answer	
- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 – ACK - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 – NAK	ACK comes after the command was successfully sent

## 7. Cashless get RTC

Command from user app	
0xFE 0x06 <CRC>	Gets the date and time information from the internal RTC of the cashless device
Device answer	
0x0A 0xFE 0x0A 0x0F 0x2E 0x31 0x1E 0x0B 0x12 0x06 0xB7	B0 – message length B1 – always 0xFE B2 – always 0x0A B3 – hour B4 – minutes B5 – seconds B6 – day of month B7 – month B8 – year B9 – day of week

## 8. Cashless set RTC

Command from user app	
0xFE 0x0A <hh> <mm> <ss> <DD> <MM> <YY> <DOW> <CRC>	Sets cashless device internal RTC <hh> hour - 24h format (from 00 to 23) <mm> minutes (from 00 to 59) <ss> seconds (from 00 to 59) <DD> day of month (from 1 to 31) <MM> month (from 1 to 12) <YY> (from 00 to 99) <DOW> day of week (from 1 to 7)
Device answer	
- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 – ACK - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 – NAK	ACK comes after the command was successfully sent

## 9. Cashless set poll interval

Command from user app	
0xFE 0x0F <PI> <CRC>	Sets the number of polls from VMC to pass until the device is sending a poll on Bluetooth serial port. You may need to adjust this depending on the VMC poll speed. Some machines are respecting the MDB specifications, some are not respecting them so you may need to increase this value if the VMC is polling peripherals faster or you may lower this value if VMC is slower. If you experience a very big interval between device polls (0x03 0xF0 0x50 0x53 message), you may need to lower this value. On the other hand, if you receive many polls in a very short interval, then you may need to increase this value. <PI> - poll interval - default 15
Device answer	
- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 – ACK - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 – NAK	ACK comes after the command was successfully sent



## 10. Cashless set MDB timings

Command from user app	
0xFE 0x10 <MDB_timeout> <rec_timeout_hi> <rec_timeout_lo> <CRC>	<p>Sets the MDB timings for reading/sniffing. You may need to adjust this depending on the VMC poll speed and peripherals responses speed. Some machines are respecting the MDB specifications, some are not respecting them so, you may need to adjust those values if the VMC is polling peripherals faster or you may lower this value if VMC is slower. If you experience a frequent sniffed message loss, you may need to increase SniffSpeed. The values are in milliseconds and, at least for SniffSpeed, use with care and increase it step-by-step, since it may affect also the cashless device responses to the VMC.</p> <p>- &lt;MDB_timeout&gt; is the MDB bus timeout (default 4ms) to fulfill the 5ms MDB standard. If you experience some peripherals message loss, then you need to increase this value step by step, with 1ms resolution</p> <p>- &lt;rec_timeout_hi&gt; and &lt;rec_timeout_lo&gt; bytes are representing the MDB bus timeout for VMC messages. The default value is 300ms, to fulfill 200ms MDB standard. If you experience some VMC message loss, then you need to increase this value.</p> <p><b>ATTENTION!!!!</b> - it is recommended to be very careful with those values, since you may brick your device (completely disable the MDB communication, due to off standard settings). If you brick your device, return to default values. If it is impossible to return to default, you may need to follow the hardware reset procedure (not described in this manual). This needs to open device's enclosure and follow a strict procedure. This will void you warranty!</p>
Device answer	
- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 - ACK - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 - NAK	ACK comes after the command was successfully sent

## 11. Clear paired devices

Command from user app	
0xFE 0x12 0x10	Sometimes, especially after pairing with many devices, you cannot pair with more, because you have reached the interface maximum paired devices. Use one of already paired devices to send this command to the interface. After this, you need to pair again with all desired devices.
Device answer	
- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 - ACK - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 - NAK	ACK comes after the command was successfully sent

## B. MDB commands

Any other message your application is sending to the device as a response to the device poll, will be interpreted as a message that should be send to the MDB bus in a response of VMC cashless poll command. It is the responsibility of your application to create correct MDB message (including adding the correct CRC).

We are offering a demo application (daemon) that can be used for high level protocol (described in a different section below). The demo application is developed using B4J (<https://www.b4x.com/b4j.html>), a simple and intuitive programming language, with a completely free RAD IDE and we are providing the source code in the download archive. The demo is a console application and was tested on Windows 10, Ubuntu and Raspbian (Raspberry Pi), it is a JAR application and by creating a script you can run it as a service.

Also, you may develop a simple bridge mobile application to send all message to your backend server. This way, a simple, cheap mobile phone can become a bridge between VMC MDB bus and your server, with the entire transaction logic on the server.

### 1. Cashless begin session

Command from user app	
Example: 0x03 0x00 0xC8 0x00 0x01 0x86 0xAC 0x00 0x00 0x00 0xFE	Starting a new cashless session - B0 - always 0x03 (according to MDB specifications) - B1-B2 - funds to transfer to VMC (HI-LO) - B2-B6 - media ID - B7 - payment type (according to MDB specifications) - B8-B9 - payment data (according to MDB specifications) - B10 - MDB CRC <b>ATTENTION!!!</b> - repeatedly send this command may scramble VMC session and may turn VMC non-responsive. Always make sure you are closing the current session before beginning another one, following MDB specification (use CANCEL SESSION REQUEST and correctly answer to VMC messages that may be triggered by this command).
Device answer	
- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 - ACK - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 - NAK	ACK comes after the command was successfully sent

### 2. Cashless cancel session request

Command from user app	
Example: 0x04 0x04	Requesting a cashless cancel session to VMC. After the SESSION COMPLETE message from, you must answer with END SESSION
Device answer	
- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 - ACK - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 - NAK or - 0x03 0x13 0x04 0x17 (session complete)	ACK comes after the command was successfully sent

### 3. Cashless vend approved

Command from user app	
Example: 0x05 0x00 0x64 0x69	Approved the vend request from the VMC - B0 - 0x05 - according to MDB specification - B1 - approved amount (HI) - B2 - approved amount (LO) - B3 - MDB CRC
Device answer	
- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 - ACK - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 - NAK	ACK comes after the command was successfully sent

## 4. Cashless vend denied

Command from user app	
Example: 0x06 0x06	Reject the vend request from the VMC
Device answer	
- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 – ACK - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 – NAK	ACK comes after the command was successfully sent

## 5. Cashless end session

Command from user app	
Example: 0x07 0x07	Send END SESSION message to VMC. Required after receiving SESSION COMPLETE from VMC.
Device answer	
- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 – ACK - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 – NAK	ACK comes after the command was successfully sent

## 6. Cashless canceled

Command from user app	
Example: 0x08 0x08	Send CANCELED message to VMC. Required after receiving READER CANCEL from VMC.
Device answer	
- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 – ACK - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 – NAK	ACK comes after the command was successfully sent

## 7. Cashless revalue approved

Command from user app	
Example: 0x0D 0x0D	Send REVALUE APPROVED message to VMC. Required after receiving REVALUE REQUEST from VMC, if your application managed to add the amount to user's account
Device answer	
- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 – ACK - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 – NAK	ACK comes after the command was successfully sent

## 8. Cashless revalue denied

Command from user app	
Example: 0x0E 0x0E	Send REVALUE DENIED message to VMC. Required after receiving REVALUE REQUEST from VMC, if your application did not managed to add the amount to user's account
Device answer	
- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 – ACK - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 – NAK	ACK comes after the command was successfully sent

## 9. Cashless display request

Command from user app	
<p>Example:            0x02 0x1E 0x20 0x54 0x68 0x69 0x73 0x20 0x69            0x73 0x20 0x61 0x20 0x74 0x65 0x73 0x74 0x20            0x20 0x20 0x20 0x20 0x20 0x6D 0x65 0x73 0x73            0x61 0x67 0x65 0x20 0x20 0x20 0x20 0x5A</p>	<p>Send DISPLAY REQUEST message to VMC. Required after receiving REVALUE REQUEST from VMC, if your application did not managed to add the amount to user's account. Some machines requires an opened session for this, some are accepting this message at any time. Please check with your VMC before.</p> <p>This example will display for 3 seconds, the following message:            " This is a test "            " message "            on a VMC with 2 x 16 characters display.            Your application is required to correctly format the text (include leading spaces for a correct alignment). Use information received in GET STATUS command, regarding VMC display number of rows, number of columns and type.</p>
Device answer	
<p>- 0x05 0xFC 0xFC 0xFC 0xFC 0xF0 - ACK            - 0x05 0xFD 0xFD 0xFD 0xFD 0xF0 - NAK</p>	<p>ACK comes after the command was successfully sent</p>

## C. Unsolicited messages

Unsolicited messages may be received from the device at any time.

Unsolicited messages may be triggered by other peripherals status change (accepting a bill or a coin, returning change, enabling or disabling, some peripheral defect reporting, etc.)

You need to treat those message. Identifying another peripheral messages are easy, considering that the first byte is the message length and the next byte(s) are the VMC message.

For example:

- 0x04 0x33 0x33 0x80 0x80

- B0 - 4 is the message length

- B1 - 0x33 - bill poll (from VMC)

- B2 - 0x33 - bill poll message CRC (from VMC)

- B3 - 0x80 - there was a bill stacked - type of bill = 0 (from bill peripheral)

- B4 - 0x80 - bill stacked CRC (from peripheral)

You need to implement those parsing procedures on your application or you may simply ignore them if you don't need those information.

Other unsolicited messages category is the for the VMC messages to the cashless device (for example, VEND REQUEST, READER CANCEL, REVALUE REQUEST, VEND/CASH SALE, READER ENABLE, READER DISABLE, etc.)

You need to correctly implement those message parsing procedures in order to make sure your cashless solution is fully functional.

For this commands, you may need to study the MBD cashless section in the MDB official documentation, to understand the MDB cashless transactions and to find some information about the messages and data format.

## IV. The high level protocol over Bluetooth, using our demo application

We are offering a demo application (daemon) that can be used for high level protocol (described in a different section below). The demo application is developed using B4J (<https://www.b4x.com/b4j.html>), a simple and intuitive programming language, with a completely free RAD IDE and we are providing the source code in the download archive. The demo is a console application and was tested on Windows 10, Ubuntu and Raspbian (Raspberry Pi), it is a JAR application and by creating a script you can run it as a service.

In high level protocol, the communication is based on sending and receiving JSON messages on an open socket.

To install our demo application, make sure you have Java 8 SDK installed on your computer, then copy the demo application in a dedicated folder and launch it with the following command:

```
#java -jar pvmdbslvd.jar
```

Then connect your your application on a socket port 5132 TCP and start sending high level protocol commands. For testing purposes, you can even use TELNET or an easiest way, by installing Packet Sender (available for Windows, GNU/Linux, Mac OS, Android and iOS at <https://packetsender.com/>)

We are also providing some Packet Sender exports for download, where you need to change the target IP to match your computer/Raspberry Pi address.

If you are turning off interface's power, the demo app will disconnect from Bluetooth and this action will generate a, exception outside of Java virtual machine, that will shut down the application. This exception cannot be catch by the demo app so, it is recommended to start the application using a script that will keep relaunching it with a few seconds delay. Also, you need to monitor the socket status in your application and reopen it accordingly when it becomes available again.

JSON messages you can sent to the demo app are, of course, case sensitive. There are 2 JSON types, commands and messages. Commands are those that are modifying demo app behavior and settings on the fly, while messages are those that will be sent directly to the MDB bus. If the commands are not correctly received or interpreted by the demo app, there will be no answer.

The device it will answer with

```
- {"NumberValue":[5,252,252,252,240],  
"StringHexValue":"05FCFCFCFCF0","MessageType":"MDBMessage"} for ACK  
and with
```

```
- {"NumberValue":[5,253,253,253,240],  
"StringHexValue":"05FDFDFDFDF0","MessageType":"MDBMessage"} for NAK
```

All VMC commands that could not be parsed by the demo app will trigger a JSON response that is including the received bytes, both as number an hex string.

Example:

```
- {"NumberValue":[3,20,1,21],"StringHexValue":"03140115", "MessageType":"MDBMessage"} -  
means cashless enabled by VMC
```

The device is also sniffing and filtering bill and coin peripherals messages and is reporting the relevant ones as a JSON message.

For example:

```
{"NumberValue":[4,51,51,9,9],"StringHexValue":"0433330909","MessageType":"MDBMessage"}  
- means that on the last poll (0x33 0x33), the bill validator responded with a bill validator disabled message (0x09 0x09). Those are unsolicited messages and can come anytime. It is
```

your application responsibility to parse (or ignore them). Same message is not sent twice. Also, ACK from other peripherals is only sent when the previous peripheral message was about a status (defective, coin/bill accepted, etc.) and will come only to signal that the situation was solved. Some payment systems are responding with ACK after reporting a status so, sometimes, a received ACK does not mean that the problem was solved. If there is a quick message exchange between VMC and peripherals (including this device), sometimes, messages are concatenated. You need to identify and parse multiple messages, by evaluating the entire received packet, according to MDB specifications. First byte is always the message length. If your received packet is smaller than the length, you must ignore it (usually it may happen on device poll message over Bluetooth, due to very tight timing). If the message length is bigger than the first byte, you need to extract the exact number bytes and parse them, then continue with the next byte which is the length of the second message in the same packet, until you reach the packet end. Each sniffed message is containing both VMC and peripheral message, concatenated:

For example:

```
{"NumberValue":[4,51,51,9,9],"StringHexValue":"0433330909","MessageType":"MDBMessage"}
```

– means that on the last poll (0x33 0x33), the bill validator responded with a bill validator disabled message (0x09 0x09).

Using our demo application as a service, and tunneling it to your server, you can move the entire cashless logic in your backend. For example, you can use a cheap SBC, like Raspberry Pi, with a secure Internet connection, to get the messages from the VMC and send them to your backend and receive your server messages to send them to the VMC.

## A. Reserved device commands

All device response messages are returning a JSON with the following format:

```
{"NumberValue":[26,251,8,1,2,0,40,9,2,0,2,16,2,1,48,48,49,56,69,52,51,53,67,55,69,48,231],
"StringHexValue":"1AFB080102002809020002100201303031384534333543374530E7",
"MessageType":"MDBMessage"}
```

- NumberValue – is an array of bytes. The first byte is the message length and the last byte is the message CRC, calculated according to MDB CRC calculation specification. CRC does not include message length byte.

- StringHexValue - is a string that contains HEX representation of each byte in the received messages. The first byte is the message length and the last byte is the message CRC, calculated according to MDB CRC calculation specification. CRC does not include message length byte.

We had included both representations to make data available in multiple formats.

Last byte on commands and answers is the CRC and is calculated by the same procedure used for MDB.

### 1. Get cashless status

Command from user app	
{"Command":"GetStatus"}	Get cashless status. You may use byte array to parse the device response or you may use string HEX value, depending on your implementation. Te application
Device answer	
<pre>{"NumberValue": [26,251,8,1,2,0,40,9,2,0,2,16,2,1,48,48,49,56,69,52,51,53,67,55,69,48,231], "StringHexValue":"1AFB080102002809020002100201303031384534333543374530E7", "MessageType":"MDBMessage"}</pre>	B0 – message length (length byte not included) B1 – always 0xFB B2 – cashless stage (described in Appendix 1) B3 – cashless scaling factor B4 – cashless decimal places B5 – cashless country code (HI byte) B6 – cashless country code (LO byte) B7 – cashless options byte B8 – cashless feature level B9 – cashless optional feature bits B10 – VMC feature level B11 – VMC columns on display B12 – VMC rows on display B13 – VMC display type info B14-B25 – cashless serial number B26 – MDB type calculated CRC from B1 to B25

### 2. Cashless internal reset

Command from user app	
{"Command":"Reset"}	Cashless will perform an internal reset and will report JUST RESET to the vending machine that should initialize it again.
Device answer	
<pre>{"NumberValue":[5,252,252,252,252,240], "StringHexValue":"05FCFCFCFCF0", "MessageType":"MDBMessage"} - ACK or {"NumberValue":[5,253,253,253,253,240], "StringHexValue":"05FDFDFDFDF0", "MessageType":"MDBMessage"} - NAK</pre>	ACK comes after the command was successfully sent



### 3. Cashless set country code

Command from user app	
<pre>{"Command": "SetCountryCode", "CountryCode": 6520}</pre>	Sets the cashless country code. Requires reset after value modification. CountryCode is the decimal representation of the HEX country code, according to MDB specification
Device answer	
<pre>{"NumberValue": [5, 252, 252, 252, 252, 240], "StringHexValue": "05FCFCFCFCF0", "MessageType": "MDBMessage"} - ACK or {"NumberValue": [5, 253, 253, 253, 253, 240], "StringHexValue": "05FDFDFDFDF0", "MessageType": "MDBMessage"} - NAK</pre>	ACK comes after the command was successfully sent

### 4. Cashless set options byte

Command from user app	
<pre>{"Command": "SetOptions", "Revalue": 1, "Multivend": 1, "CashSale": 1}</pre>	Sets the cashless options byte. Requires reset after value modification. - If Revalue field is set to 1, then the cashless device will report to VMC that it can support this function - If Multivend field is set to 1, then the cashless device will report to VMC that it can support this function - If CashSale field is set to 1, then the cashless device will report to VMC that it can support this function (VEND/CASH SALE subcommand)
Device answer	
<pre>{"NumberValue": [5, 252, 252, 252, 252, 240], "StringHexValue": "05FCFCFCFCF0", "MessageType": "MDBMessage"} - ACK or {"NumberValue": [5, 253, 253, 253, 253, 240], "StringHexValue": "05FDFDFDFDF0", "MessageType": "MDBMessage"} - NAK</pre>	ACK comes after the command was successfully sent

### 5. Cashless set scaling factor

Command from user app	
<pre>{"Command": "SetScalingFactor", "ScalingFactor": 1}</pre>	Sets the cashless scaling factor. Requires reset after value modification.
Device answer	
<pre>{"NumberValue": [5, 252, 252, 252, 252, 240], "StringHexValue": "05FCFCFCFCF0", "MessageType": "MDBMessage"} - ACK or {"NumberValue": [5, 253, 253, 253, 253, 240], "StringHexValue": "05FDFDFDFDF0", "MessageType": "MDBMessage"} - NAK</pre>	ACK comes after the command was successfully sent

### 6. Cashless set decimal places

Command from user app	
<pre>{"Command": "SetDecimalPlaces", "DecimalPlaces": 2}</pre>	Sets the cashless decimal places Requires reset after value modification.
Device answer	
<pre>{"NumberValue": [5, 252, 252, 252, 252, 240], "StringHexValue": "05FCFCFCFCF0", "MessageType": "MDBMessage"} - ACK or {"NumberValue": [5, 253, 253, 253, 253, 240], "StringHexValue": "05FDFDFDFDF0", "MessageType": "MDBMessage"} - NAK</pre>	ACK comes after the command was successfully sent

## 7. Cashless get RTC

Command from user app	
<code>{"Command": "GetRTC"}</code>	Gets the date and time information from the internal RTC of the cashless device
Device answer	
<code>{"NumberValue": [10, 254, 10, 3, 11, 11, 1, 1, 0, 1, 36], "StringHexValue": "0AFE0A030B0B0101000124", "MessageType": "MDBMessage"}</code>	B0 – message length B1 – always 0xFE B2 – always 0x0A B3 – hour B4 – minutes B5 – seconds B6 – day of month B7 – month B8 – year B9 – day of week

## 8. Cashless set RTC

Command from user app	
<code>{"Command": "SetRTC", "Hour": 15, "Minutes": 45, "Seconds": 00, "Day": 30, "Month": 11, "Year": 18, "WeekDay": 6}</code>	Sets cashless device internal RTC
Device answer	
<code>{"NumberValue": [5, 252, 252, 252, 252, 240], "StringHexValue": "05FCFCFCFCF0", "MessageType": "MDBMessage"}</code> – ACK or <code>{"NumberValue": [5, 253, 253, 253, 253, 240], "StringHexValue": "05FDFDFDFDF0", "MessageType": "MDBMessage"}</code> – NAK	ACK comes after the command was successfully sent

## 9. Cashless set poll interval

Command from user app	
<code>{"Command": "SetPollInterval", "Interval": 15}</code>	Sets the number of polls from VMC to pass until the device is sending a poll on Bluetooth serial port. You may need to adjust this depending on the VMC poll speed. Some machines are respecting the MDB specifications, some are not respecting them so you may need to increase this value if the VMC is polling peripherals faster or you may lower this value if VMC is slower. If you experience a very big interval between device polls (low speed command sending to VMC), you may need to lower this value. On the other hand, if you receive many polls in a very short interval, then you may need to increase this value. <b>ATTENTION!!!</b> - wrong value may lead to an unresponsive device that may force you to proceed with a cashless interface hardware reset. Do not modify those parameters until you really need.
Device answer	
<code>- {"NumberValue": [5, 252, 252, 252, 252, 240], "StringHexValue": "05FCFCFCFCF0", "MessageType": "MDBMessage"}</code> – ACK or <code> {"NumberValue": [5, 253, 253, 253, 253, 240], "StringHexValue": "05FDFDFDFDF0", "MessageType": "MDBMessage"}</code> – NAK	ACK comes after the command was successfully sent

## 10. Cashless set MDB timings

Command from user app	
<pre>{"Command": "SetMDBTimings", "SniffSpeed": 4, "SniffListenWait": 300}</pre>	<p>Sets the MDB timings for reading/sniffing. You may need to adjust this depending on the VMC poll speed and peripherals responses speed. Some machines are respecting the MDB specifications, some are not respecting them so, you may need to adjust those values if the VMC is polling peripherals faster or you may lower this value if VMC is slower. If you experience a frequent sniffed message loss, you may need to increase SniffSpeed. The values are in milliseconds and, at least for SniffSpeed, use with care and increase it step-by-step, since it may affect also the cashless device responses to the VMC.</p> <p><b>ATTENTION!!!</b> - wrong values may lead to an unresponsive device that may force you to proceed with a cashless interface hardware reset. Do not modify those parameters until you really need.</p>
Device answer	
<pre>- {"NumberValue": [5, 252, 252, 252, 252, 240], "StringHexValue": "05FCFCFCFCF0", "MessageType": "MDBMessage"} - ACK or {"NumberValue": [5, 253, 253, 253, 253, 240], "StringHexValue": "05FDFDFDFDF0", "MessageType": "MDBMessage"} - NAK</pre>	<p>ACK comes after the command was successfully sent</p>

## 11. Clear paired devices

Command from user app	
<pre>"Command": "ClearPairedDevices"</pre>	<p>Sometimes, especially after pairing with many devices, you cannot pair with more, because you have reached the interface maximum paired devices. Use one of already paired devices to send this command to the interface. After this, you need to pair again with all desired devices.</p>
Device answer	
<pre>- {"NumberValue": [5, 252, 252, 252, 252, 240], "StringHexValue": "05FCFCFCFCF0", "MessageType": "MDBMessage"} - ACK or {"NumberValue": [5, 253, 253, 253, 253, 240], "StringHexValue": "05FDFDFDFDF0", "MessageType": "MDBMessage"} - NAK</pre>	<p>ACK comes after the command was successfully sent</p>

## B. MDB commands

Any other message your application is sending to the device as a response to the device poll, will be interpreted as a message that should be send to the MDB bus in a response of VMC cashless poll command. It is the responsibility of your application to create correct MDB message (including adding the correct CRC).

We are offering a demo application (daemon) that can be used for high level protocol (described in a different section below). The demo application is developed using B4J (<https://www.b4x.com/b4j.html>), a simple and intuitive programming language, with a completely free RAD IDE and we are providing the source code in the download archive. The demo is a console application and was tested on Windows 10, Ubuntu and Raspbian (Raspberry Pi), it is a JAR application and by creating a script you can run it as a service.

### 1. Cashless begin session

Command from user app	
Example: <pre>{"Message":"BeginSession","FundsAvailable":200,"PaymentMediaID":100012,"PaymentType":0,"PaymentData":0}</pre>	Starting a new cashless session <b>ATTENTION!!!</b> - repeatedly send this command may scramble VMC session and may turn VMC non-responsive. Always make sure you are closing the current session before beginning another one, following MDB specification (use CANCEL SESSION REQUEST and correctly answer to VMC messages that may be triggered by this command).
Device answer	
<pre>- {"NumberValue":[5,252,252,252,252,240], "StringHexValue":"05FCFCFCFCF0", "MessageType":"MDBMessage"} - ACK or {"NumberValue":[5,253,253,253,253,240], "StringHexValue":"05FDFDFDFDF0", "MessageType":"MDBMessage"} - NAK</pre>	ACK comes after the command was successfully sent

### 2. Cashless cancel session request

Command from user app	
Example: <pre>{"Message":"CancelSessionRequest"}</pre>	Requesting a cashless cancel session to VMC. After the SESSION COMPLETE message from, you must answer with END SESSION
Device answer	
<pre>- {"NumberValue":[5,252,252,252,252,240], "StringHexValue":"05FCFCFCFCF0", "MessageType":"MDBMessage"} - ACK or {"NumberValue":[5,253,253,253,253,240], "StringHexValue":"05FDFDFDFDF0", "MessageType":"MDBMessage"} - NAK or {"NumberValue":[3,19,4,23], "StringHexValue":"03130417", "MessageType":"MDBMessage"} - SESSION COMPLETE</pre>	ACK comes after the command was successfully sent

### 3. Cashless vend approved

Command from user app	
Example: { "Message": "VendApproved", "VendAmount": 100 }	Approving the vend request from the VMC
Device answer	
- { "NumberValue": [5, 252, 252, 252, 252, 240], "StringHexValue": "05FCFCFCFCF0", "MessageType": "MDBMessage" } - ACK or { "NumberValue": [5, 253, 253, 253, 253, 240], "StringHexValue": "05FDFDFDFDF0", "MessageType": "MDBMessage" } - NAK	ACK comes after the command was successfully sent

### 4. Cashless vend denied

Command from user app	
Example: { "Message": "VendDenied" }	Reject the vend request from the VMC
Device answer	
- { "NumberValue": [5, 252, 252, 252, 252, 240], "StringHexValue": "05FCFCFCFCF0", "MessageType": "MDBMessage" } - ACK or { "NumberValue": [5, 253, 253, 253, 253, 240], "StringHexValue": "05FDFDFDFDF0", "MessageType": "MDBMessage" } - NAK	ACK comes after the command was successfully sent

### 5. Cashless end session

Command from user app	
Example: { "Message": "EndSession" }	Send END SESSION message to VMC. Required after receiving SESSION COMPLETE from VMC.
Device answer	
- { "NumberValue": [5, 252, 252, 252, 252, 240], "StringHexValue": "05FCFCFCFCF0", "MessageType": "MDBMessage" } - ACK or { "NumberValue": [5, 253, 253, 253, 253, 240], "StringHexValue": "05FDFDFDFDF0", "MessageType": "MDBMessage" } - NAK	ACK comes after the command was successfully sent

### 6. Cashless canceled

Command from user app	
Example: { "Message": "Canceled" }	Send CANCELED message to VMC. Required after receiving READER CANCEL from VMC.
Device answer	
- { "NumberValue": [5, 252, 252, 252, 252, 240], "StringHexValue": "05FCFCFCFCF0", "MessageType": "MDBMessage" } - ACK or { "NumberValue": [5, 253, 253, 253, 253, 240], "StringHexValue": "05FDFDFDFDF0", "MessageType": "MDBMessage" } - NAK	ACK comes after the command was successfully sent

## 7. Cashless revalue approved

Command from user app	
Example: {"Message":"RevalueApproved"}	Send REVALUE APPROVED message to VMC. Required after receiving REVALUE REQUEST from VMC, if your application managed to add the amount to user's account
Device answer	
- {"NumberValue":[5,252,252,252,252,240], "StringHexValue":"05FCFCFCFCF0", "MessageType":"MDBMessage"} - ACK or {"NumberValue":[5,253,253,253,253,240], "StringHexValue":"05FDFDFDFDF0", "MessageType":"MDBMessage"} - NAK	ACK comes after the command was successfully sent

## 8. Cashless revalue denied

Command from user app	
Example: {"Message":"RevalueDenied"}	Send REVALUE DENIED message to VMC. Required after receiving REVALUE REQUEST from VMC, if your application did not managed to add the amount to user's account
Device answer	
- {"NumberValue":[5,252,252,252,252,240], "StringHexValue":"05FCFCFCFCF0", "MessageType":"MDBMessage"} - ACK or {"NumberValue":[5,253,253,253,253,240], "StringHexValue":"05FDFDFDFDF0", "MessageType":"MDBMessage"} - NAK	ACK comes after the command was successfully sent

## 9. Cashless display request

Command from user app	
Example: {"Message":"DisplayRequest", "DisplayMessage":" This is a test message", "DisplayTime":30}	Send DISPLAY REQUEST message to VMC. Required after receiving REVALUE REQUEST from VMC, if your application did not managed to add the amount to user's account. Some machines requires an opened session for this, some are accepting this message at any time. Please check with your VMC before. This example will display for 3 seconds, the following message: " This is a test " " message " on a VMC with 2 x 16 characters display. Your application is required to correctly format the text (include leading spaces for a correct alignment). Use information received in GET STATUS command, regarding VMC display number of rows, number of columns and type. - DisplayTime resolution is 0.1sec (30 x 0.1 = 3sec)
Device answer	
- {"NumberValue":[5,252,252,252,252,240], "StringHexValue":"05FCFCFCFCF0", "MessageType":"MDBMessage"} - ACK or {"NumberValue":[5,253,253,253,253,240], "StringHexValue":"05FDFDFDFDF0", "MessageType":"MDBMessage"} - NAK	ACK comes after the command was successfully sent

## C. Unsolicited messages

Unsolicited messages may be received from the device at any time.

Unsolicited messages may be triggered by other peripherals status change (accepting a bill or a coin, returning change, enabling or disabling, some peripheral defect reporting, etc.)

You need to treat those message. Identifying another peripheral messages are easy, considering that the first byte is the message length and the next byte(s) are the VMC message.

You need to implement those parsing procedures on your application or you may simply ignore them if you don't need those information.

Other unsolicited messages category is the for the VMC messages to the cashless device (for example, VEND REQUEST, READER CANCEL, REVALUE REQUEST, VEND/CASH SALE, READER ENABLE, READER DISABLE, etc.)

You need to correctly implement those message parsing procedures in order to make sure your cashless solution is fully functional.

For this commands, you may need to study the MBD cashless section in the MDB official documentation, to understand the MDB cashless transactions and to find some information about the messages and data format.

Examples:

```
- {"NumberValue":[7,19,0,0,50,0,10,79],  
"StringHexValue":"0713000032000A4F",  
"MessageType":"MDBMessage"} - it is a vend request message from VMC, with a product value  
of 50 and a product ID 10.
```

```
- {"NumberValue":[4,51,51,128,128],  
"StringHexValue":"0433338080",  
"MessageType":"MDBMessage"} - it is a bill validator related message (VMC polls the bill  
validator and the peripheral is answering with a BILL STAKED message, bill type 0
```

```
- {"NumberValue":[5,21,0,0,100,121],  
"StringHexValue":"051500006479",  
"MessageType":"MDBMessage"} - it is a REVALUE REQUEST message from VMC, with the  
revalue amount of 100.
```

## Appendix 1 – Cashless stage codes

Reported cashless stage	Description
0	Cashless power-up or reset by the vending machine or by the user application
1	Setup config data message received from VMC
2	MAX/MIN prices received from VMC
3	Cashless reported JUST RESET to VMC
4	VEND REQUEST received from VMC
5	READER CANCEL received from VMC
6	VEND SUCCESS received from VMC
7	VEND FAILURE received from VMC
8	SESSION COMPLETE received from VMC
9	READER DISABLE received from VMC
10	READER ENABLE received from VMC
11	READER CANCEL received from VMC
12	DATA ENTRY received from VMC
13	Cashless just answered to VMC EXPANSION REQUEST
14	REVALUE REQUEST received from VMC
15	REVALUE LIMIT REQUEST received from VMC
16	Cashless reported READER SETUP DATA to VMC
17	EXPANSTION REQUEST ID received from VMC