PICOVEND EZ SELF LITE (controller for self services vending systems) v2020-07-27 (fiscal ready)

Table of Contents

I. Introduction

This device is intended to be used for cash/cashless self service service building. It can be used (but not limited to) for the following applications:

- Kiddie rides or amusement parks (playgrounds);
- Laundry systems;
- Self service car washing systems;
- Paid access;
- Supermarkets carts rentals;
- Other applications that needs to be operated by tokens and requires a token dispenser/change machine;
- Other applications that needs to be operated directly by a paid timer.

For Kiddie Rides, laundry systems and car washing systems, it can be used to build your own change machine/token dispenser that is accepting coins, bills and credit cards to dispense tokens or it can be used directly to keep the toy active for a configurable amount of time.

For supermarkets, it can be used to build a change machine, accepting bills or credit cards to dispense coins/tokens for carts unlocking.

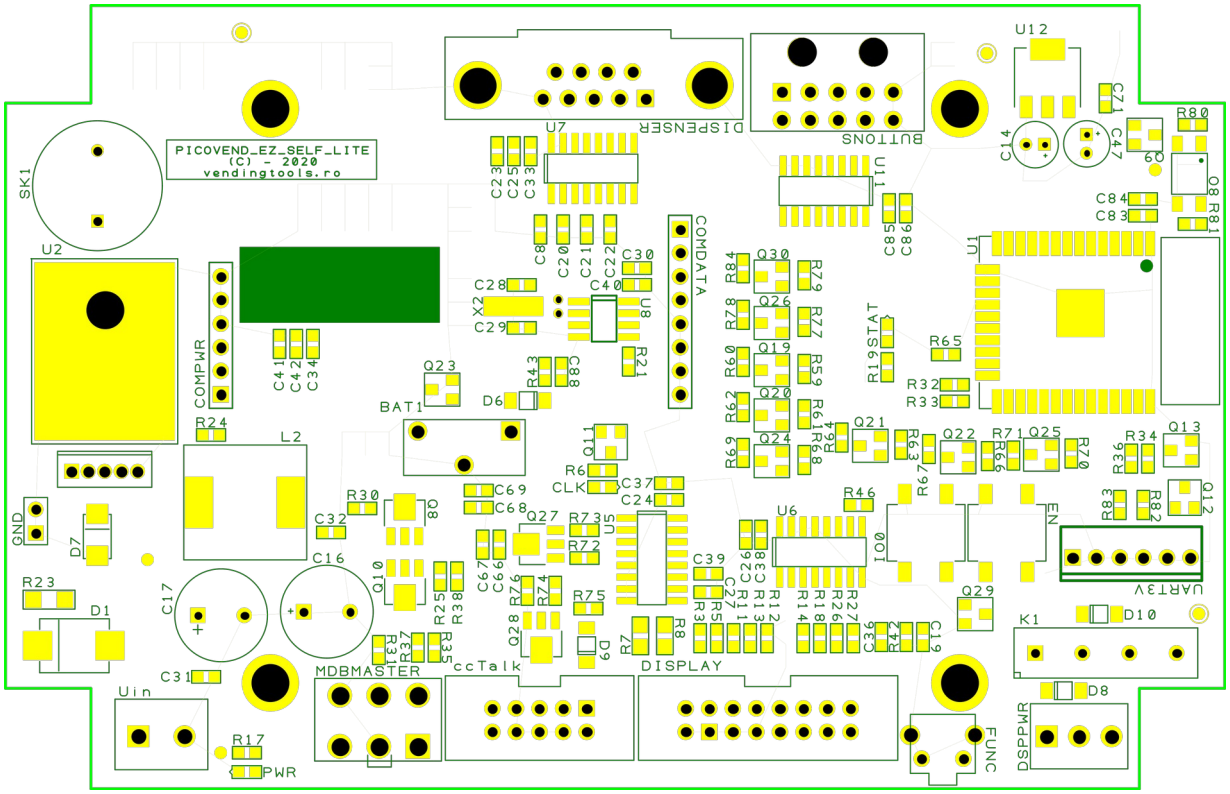
For paid access, it can be used to build a system that is accepting a fixed amount and release a turnstile, an electric door or other control access system.

Two different firmware versions are available for this device:
- First is using one or many MDB payment systems (coin acceptor, bill validator and card reader) to dispense tokens or coins from a ccTalk hopper;
- Second is using one or many MDB payment systems (coin acceptor, bill validator and card reader) to control a relay output for a configured time, depending on the cash/cashless amount.

The controller is supporting multiple tokens/time values for different amounts (for example, on 1.00EUR it can dispense 1 token or enable the timer for 60 seconds and for 2.00EUR it can dispense 3 tokens or enable the timer for 180 seconds). For cashless (including bank card payments) it supports an automatic credit withdrawal on each transaction, for example, when a card is presented, the controller will always authorize a fixed amount or 2.00EUR and if the amount is authorized, it will dispense a fixed number of tokens or it will enable the timer for a fixed amount of time.

II. Hardware

A. Board overview



Picture 1: PICOVEND EZ SELF LITE

B. Connectors description

1. Uin – connect your external power supply to this connector in order to power the device and, also, the connected peripherals (MDB payment systems, ccTalk payment systems, etc.). You need to make sure your power supply is matching the connected MDB and ccTalk power requirements (voltage and current). The maximum momentary drained current simultaneously drained from MDB and ccTalk should not exceed 4A.
- PIN#1 (the squared shape pin) is for +VDC;
- PIN#2 is for power GND.

2. MDBMASTER – this connector allows the device interfacing with MDB peripherals (bill validator/recycler, coin acceptor/changer and cashless device);

3. ccTalk – this is the connector for standard 10pin ccTalk interfaces. It's pinout is the following:
- PIN #1 – ccTalk data;
- PIN #2 – N/C;
- PIN #3 – N/C;
- PIN #4 – GND;
- PIN #5 – N/C;
- PIN #6 – N/C;
- PIN #7 – VCC POWER;
- PIN #8 – GND;
- PIN #9 – N/C;
- PIN #10 – VCC POWER;

4. DISPLAY – is the connector for external alphanumerical display, provided with the controller.

5. DISPENSER – this is the connector for RS232 interface (not used in the current version, we can develop, on request, a custom firmware that can use some RS232 peripherals, such as an RFID card

dispenser, a card reader, a printer or something similar). This is a regular DB9 connector, with RX, TX and GND pins.
6. UART3V – this is the connector for 3V3 UART interface is the connector for firmware upload.
- PIN #1 – N/C;
- PIN #2 – GND;
- PIN #3 – TXD;
- PIN #4 – RXD;
- PIN #5 – RTS# not used in this configuration;
- PIN #6 – N/C.

7. BUTTONS – it is used only in timed output mode, to start the internal timer and energize the relay coil. It always need to connect a normal open push button (not supplied by our company) to pins 1 and 9. In timer working mode, closing a contact between this pins will start the countdown.
- PIN #1 – Button #1;
- PIN #2 – Button #2;
- PIN #3 – Button #3;
- PIN #4 – Button #4;
- PIN #5 – Button #5;
- PIN #6 – Button #6;
- PIN #7 – Button #7;
- PIN #8 – Button #8;
- PIN #9 and #10 GND. You need to use N/O buttons, with one wire connected to it's corresponding pin on BUTTONS connector and the other wire connected to GND. GND is common for all buttons.

8. DSPPWR – it is a NO relay output that is closed in timed mode, during enable time. The relay only supports 100mA at 100VAC or 500mA at 20VDC. If you need to control a peripheral with higher current, you need to use an external relay or contactor and make sure it's coil is rated to the mentioned values.
- PIN #1 (the squared pin) – contact A;
- PIN #2 – contact B;
- PIN #3 – system GND.

9. FUNC button – the button function will be detailed below in the configuring section.

III. Token dispenser/change machine working mode

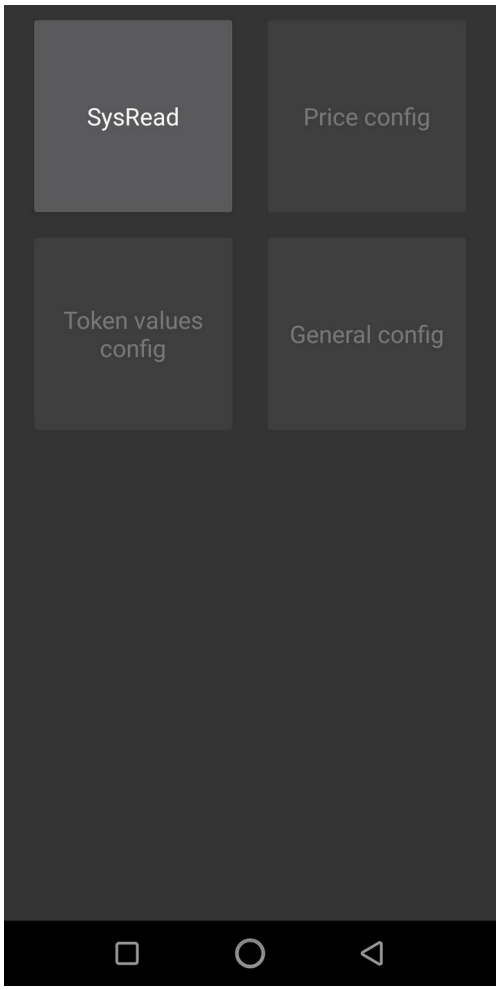
Configuration is available by using an Android application only. There are no hidden menus and buttons on this device and the Android application is the easiest way to configure the device
The Android application is connecting over Bluetooth. The controller Bluetooth needs to be activated for configuring and deactivated after configuration finished.

A. Bluetooth activation

To activate the Bluetooth communication, apply power to the controller while keeping the “FUNC” button pressed. The buzzer will emit one short beep, followed by another short beeps and the display will show “Bluetooth active” message.
Go to your Android phone Bluetooth menu and pair your phone with PVEZSELF in range. Please note that you need to have only one controller with Bluetooth active at a time.

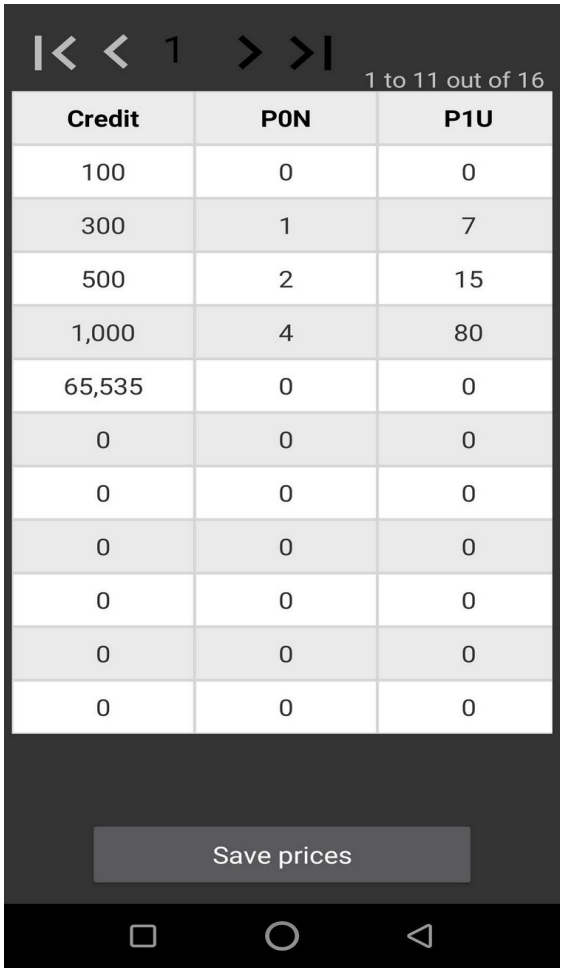
B. Android application flow

1. Start the application that will automatically try to connect to PVEZSELF device. If successful, you will get a message on your phone and the following screen will be displayed.



Press “SysRead” button, the application will read current controller settings. Some data will be displayed and some short beeps can be hear some short beeps

Press “Back” button and go to “Price config”

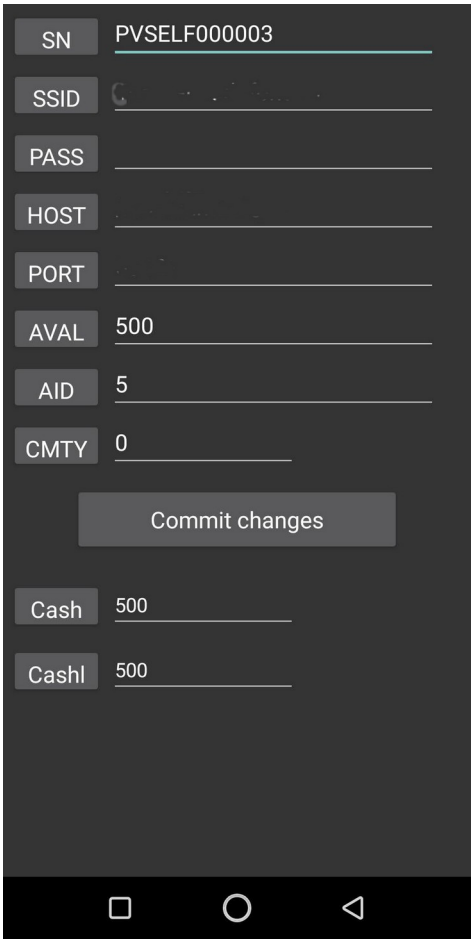


In the “Credit” column you need to set the target amounts and in the “P0N” column you need to set the number of tokens/coins to be dispensed for each target amount. The latest credit should be always set to 65535 and the P0N for this target credit should be always 0. The target credit amounts should be in ascending order, for example, first row should have credit 150, the second should have 300 and so on. For example, if the credit is set to 1.50 and the P0N is set to 1, then for 1.50EUR the device will dispense 1 token. Also, if the second row has a credit of 300 and a P0N of 3, for 3.00EUR, there will be 3 tokens dispensed. This way, you can also set some bonuses on higher target credit amounts. P1N column does not matter on this configuration. To modify a value of one table cell, just apply a long press on the desired cell.

After setting the desired values, press “Save prices” button and wait.

The “Token values” option will set the tokens values, if the controller is connected with an MDB coin acceptor that receives tokens. The configuration depends on your coin acceptor settings and they mus match (for example, set on channel #1 150 if your coin acceptor is sending a “token received” message on channel #1 for 1.50EUR tokens. Push “save token values” when you are done setting the token values, then hit “Back button to go back to the main screen.

The “General config” option allows you to set some general controller settings. Please take care on those settings, since they can brick your controller if don’t pay attention.



- **SN** is the controller serial number, that will be also periodically show on controller’s display. This string should be always 12 characters length, any other length will make the controller acting unpredictable.

- **SSID** – it is not used in this version, can be anything, including blank
- **PASS** - it is not used in this version, can be anything, including blank
- **HOST** - it is not used in this version, can be anything, including blank
- **PORT** - it is not used in this version, can be anything, including blank
- **AVAL** – it is the automatic value that the controller will try to authorize when an MDB cashless payment transaction is started. For example, if this value is 500, the controller will try to authorize a 5.00EUR payment with cashless device or bank card and will dispense the number of tokens corresponding with this amount, set in “Price config” section.
- **AID** – is the product ID reported to the cashless device when a cashless transaction needs authorization.
- **CMTY** – it should be always 0 for this configuration.

After modifying any of the above settings, push the label in front of the field to save it to the device. When finishing all modification, push “Commit changes” to save all settings in the NV flash memory.

Restart the device when all settings are done to allow the controller load and set the new configuration.

IV. Timer working mode

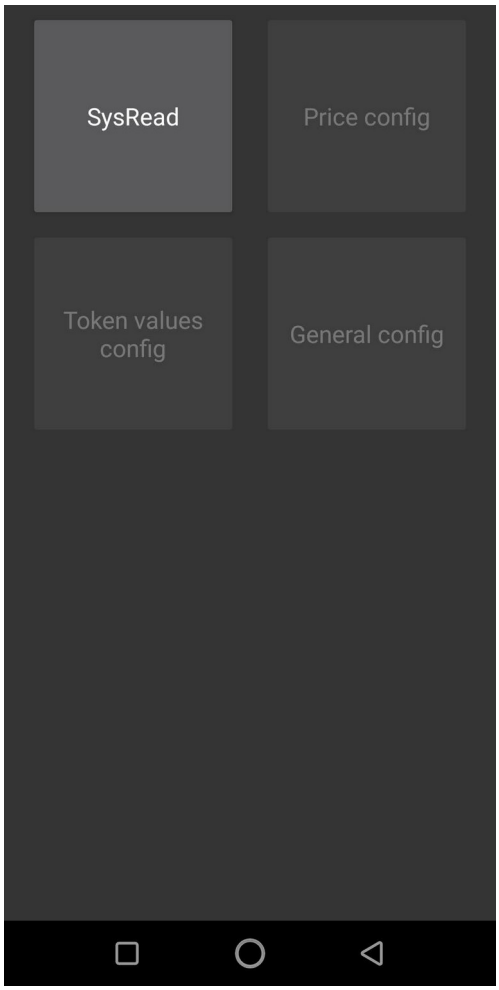
Configuration is available by using an Android application only. There are no hidden menus and buttons on this device and the Android application is the easiest way to configure the device.
The Android application is connecting over Bluetooth. The controller Bluetooth needs to be activated for configuring and deactivated after configuration finished.

A. Bluetooth activation

To activate the Bluetooth communication, apply power to the controller while keeping the “FUNC” button pressed. The buzzer will emit one short beep, followed by another short beeps and the display will show “Bluetooth active” message.
Go to your Android phone Bluetooth menu and pair your phone with PVEZSELF in range. Please note that you need to have only one controller with Bluetooth active at a time.

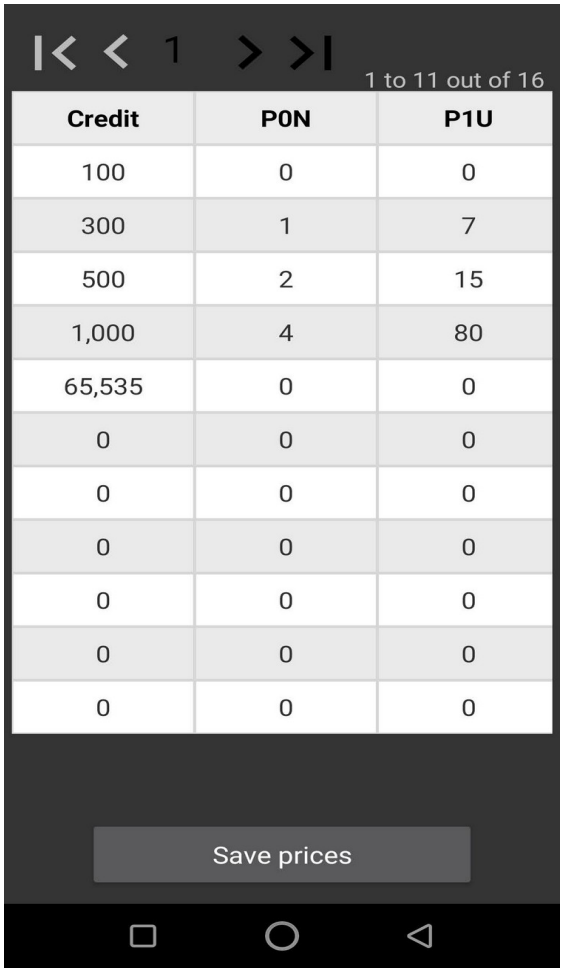
B. Android application flow

1. Start the application that will automatically try to connect to PVEZSELF device. If successful, you will get a message on your phone and the following screen will be displayed.



Press “SysRead” button, the application will read current controller settings. Some data will be displayed and some short beeps can be hear some short beeps

Press “Back” button and go to “Price config”

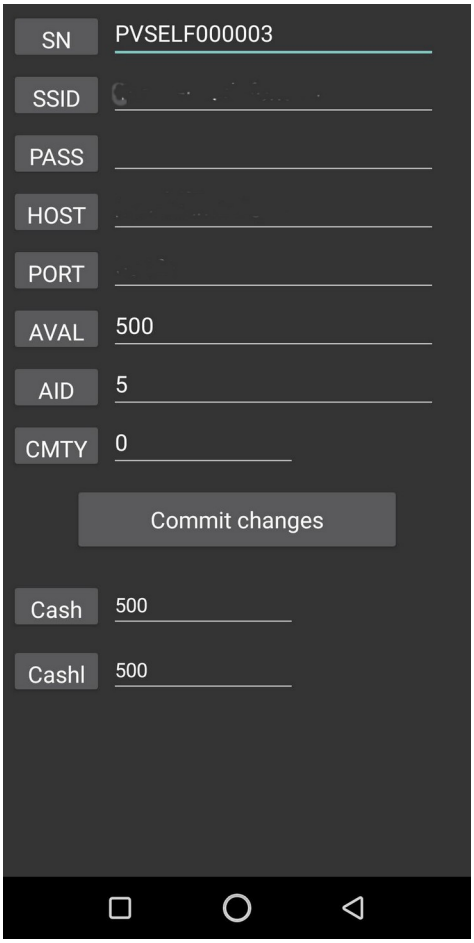


In the “Credit” column you need to set the target amounts and in the “P0N” column you need to set the time (in seconds) for each target amount. The latest credit should be always set to 65535 and the P0N for this target credit should be always 0. The target credit amounts should be in ascending order, for example, first row should have credit 150, the second should have 300 and so on. For example, if the credit is set to 1.50 and the P0N is set to 30, the relay output contacts will be closed for 30 second. Also, if the second row has a credit of 300 and a P0N of 300, the relay output contacts will be closed for 5 minutes (300 seconds). This way, you can also set some bonuses on higher target credit amounts. P1N column does not matter on this configuration. To modify a value of one table cell, just apply a long press on the desired cell.

After setting the desired values, press “Save prices” button and wait.

The “Token values” option will set the tokens values, if the controller is connected with an MDB coin acceptor that receives tokens. The configuration depends on your coin acceptor settings and they mus match (for example, set on channel #1 150 if your coin acceptor is sending a “token received” message on channel #1 for 1.50EUR tokens. Push “save token values” when you are done setting the token values, then hit “Back button to go back to the main screen.

The “General config” option allows you to set some general controller settings. Please take care on those settings, since they can brick your controller if don’t pay attention.



- **SN** is the controller serial number, that will be also periodically show on controller’s display. This string should be always 12 characters length, any other length will make the controller acting unpredictable.

- **SSID** – it is not used in this version, can be anything, including blank
- **PASS** - it is not used in this version, can be anything, including blank
- **HOST** - it is not used in this version, can be anything, including blank
- **PORT** - it is not used in this version, can be anything, including blank
- **AVAL** – it is the automatic value that the controller will try to authorize when an MDB cashless payment transaction is started. For example, if this value is 500, the controller will try to authorize a 5.00EUR payment with cashless device or bank card and will dispense the number of tokens corresponding with this amount, set in “Price config” section.
- **AID** – is the product ID reported to the cashless device when a cashless transaction needs authorization.
- **CMTY** – it should be always 0 for this configuration.

After modifying any of the above settings, push the label in front of the field to save it to the device. When finishing all modification, push “Commit changes” to save all settings in the NV flash memory.

Restart the device when all settings are done to allow the controller load and set the new configuration.

V. 8 relays timer working mode

This set of commands is working only when an optional PICOVEND EZ ESP 8X slave relay board is connected on the MDB interface of PICOVEND EZ SELF LITE module. Please check the optional PICOVEND EZ ESP 8X board manual for it's usage/specifications. Mainly, this board is used to control up to 8 external circuits, being equipped with 8 relays.

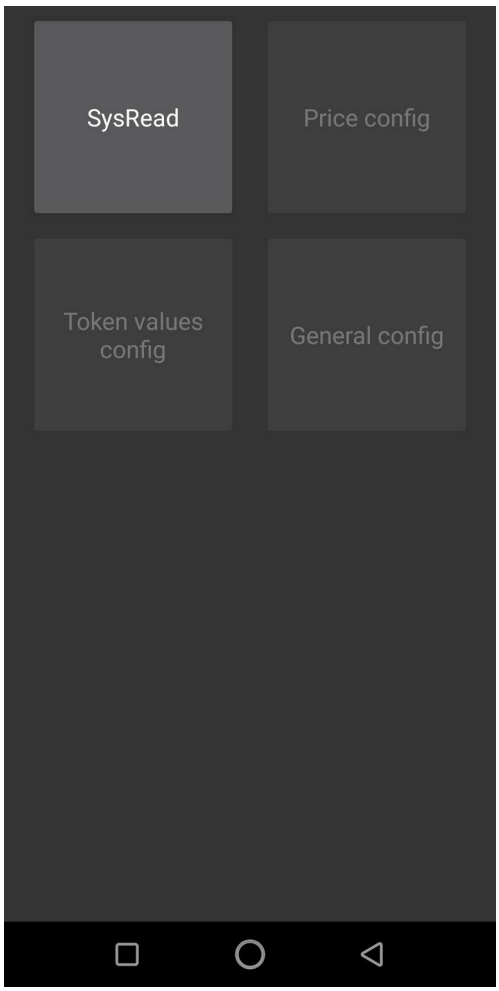
Configuration is available by using an Android application only. There are no hidden menus and buttons on this device and the Android application is the easiest way to configure the device.
The Android application is connecting over Bluetooth. The controller Bluetooth needs to be activated for configuring and deactivated after configuration finished.

A. Bluetooth activation

To activate the Bluetooth communication, apply power to the controller while keeping the "FUNC" button pressed. The buzzer will emit one short beep, followed by another short beeps and the display will show "Bluetooth active" message.
Go to your Android phone Bluetooth menu and pair your phone with PVEZSELF in range. Please note that you need to have only one controller with Bluetooth active at a time.

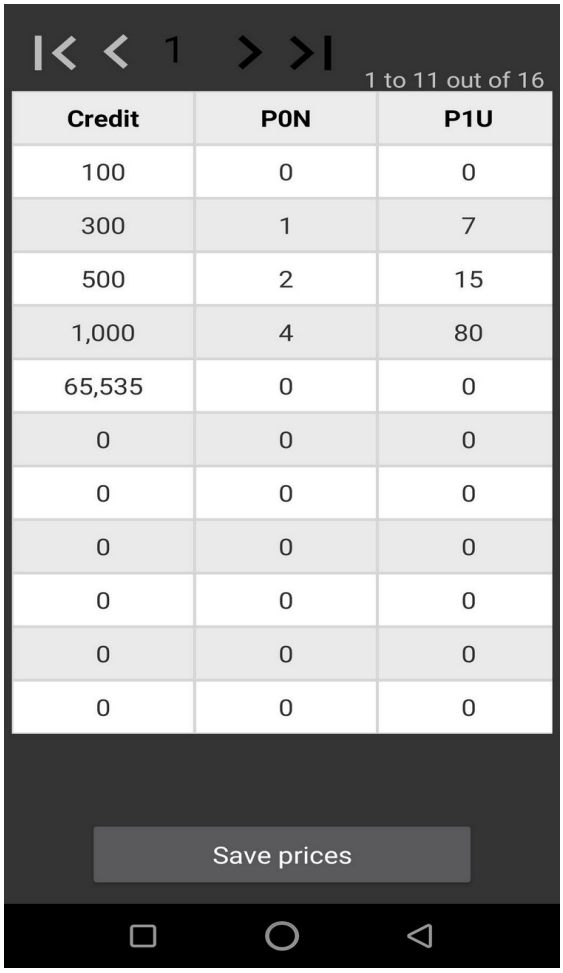
B. Android application flow

1. Start the application that will automatically try to connect to PVEZSELF device. If successful, you will get a message on your phone and the following screen will be displayed.



Press “SysRead” button, the application will read current controller settings. Some data will be displayed and some short beeps can be hear some short beeps

Press “Back” button and go to “Price config”

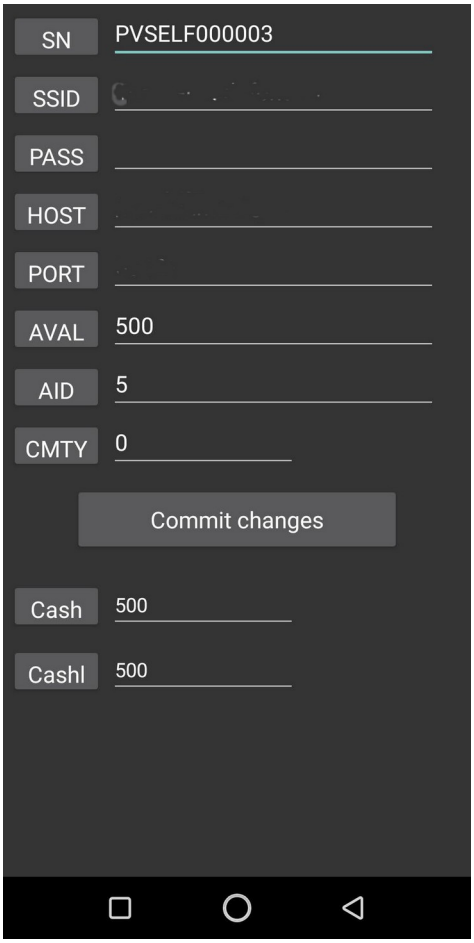


In the “Credit” column you need to set the target amounts for each relay (row #1 is for relay #1, row #2 is for relay #2, etc.) and in the “P0N” column you need to set the time (in seconds) for each target amount.
We are recommending to set the same target amount for all relays and to set different time for each relay if you need different prices.
For example, set first 8 “Credit” rows to 100 (1.00EUR) and then set P0N as follows:
- row 1 – 120sec
- row 2 – 90sec
- etc.
Doing this, you will have a price of 1.00EUR for 2 minutes on relay #1 and 1.00EUR for one and a half minute on relay #2. When switching from one “program/button” to another, the controller will convert the remaining time back in credit and use the calculated credit to compute the new remaining time for the selected button.
P1N column does not matter on this configuration. To modify a value of one table cell, just apply a long press on the desired cell.

After setting the desired values, press “Save prices” button and wait.

The “Token values” option will set the tokens values, if the controller is connected with an MDB coin acceptor that receives tokens. The configuration depends on your coin acceptor settings and they mus match (for example, set on channel #1 150 if your coin acceptor is sending a “token received” message on channel #1 for 1.50EUR tokens. Push “save token values” when you are done setting the token values, then hit “Back button to go back to the main screen.

The “General config” option allows you to set some general controller settings. Please take care on those settings, since they can brick your controller if don’t pay attention.



- **SN** is the controller serial number, that will be also periodically show on controller’s display. This string should be always 12 characters length, any other length will make the controller acting unpredictable.

- **SSID** – it is not used in this version, can be anything, including blank
- **PASS** - it is not used in this version, can be anything, including blank
- **HOST** - it is not used in this version, can be anything, including blank
- **PORT** - it is not used in this version, can be anything, including blank
- **AVAL** – it is the automatic value that the controller will try to authorize when an MDB cashless payment transaction is started. For example, if this value is 500, the controller will try to authorize a 5.00EUR payment with cashless device or bank card and will dispense the number of tokens corresponding with this amount, set in “Price config” section.
- **AID** – is the product ID reported to the cashless device when a cashless transaction needs authorization.
- **CMTY** – it should be always 0 for this configuration.

After modifying any of the above settings, push the label in front of the field to save it to the device. When finishing all modification, push “Commit changes” to save all settings in the NV flash memory.

Restart the device when all settings are done to allow the controller load and set the new configuration.