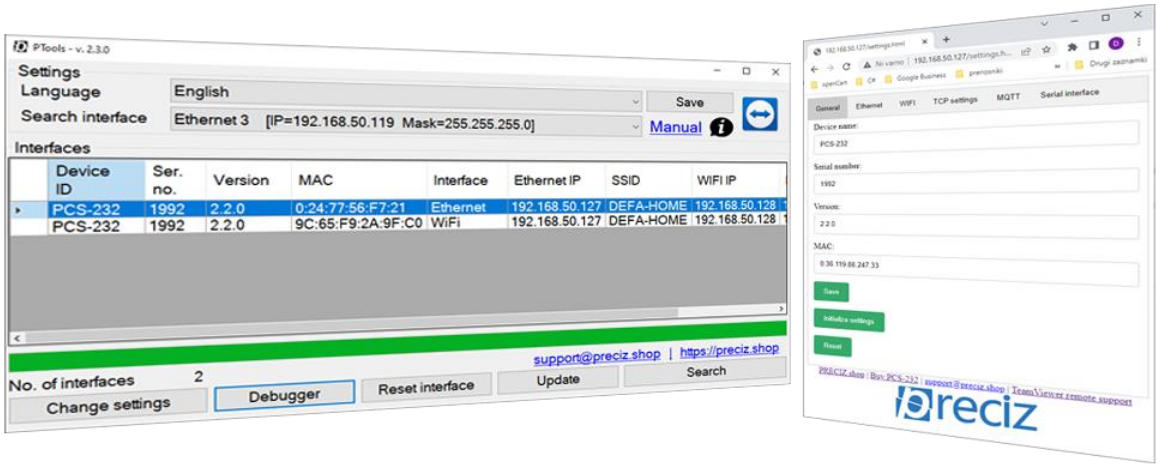
USER MANUAL PTools for desktop & web app



https://preciz.shop https://www.youtube.com/precizsi

Content

1. Description

This manual describes usage of PTools software which is simple software for configuration and testing/debugging communication between serial device and third party software eg. ERP, Cloud, TCP server etc..

With PTools you can configure interface settings:

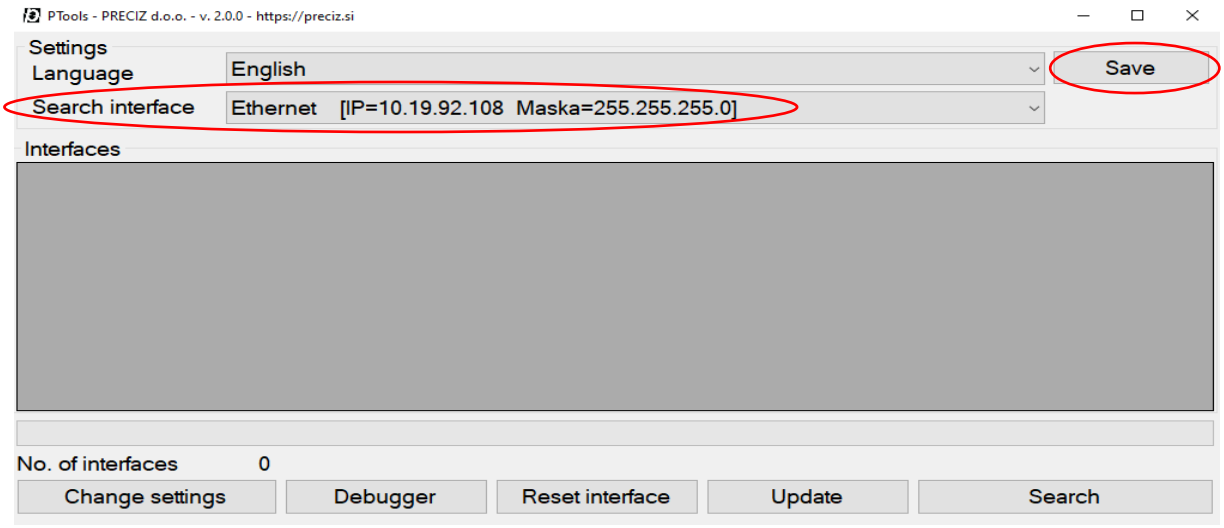
- General settings
- Ethernet settings (only for interfaces with Ethernet)
- WIFI settings (only for interfaces with WIFI)
- TCP server/client settings
- MQTT settings
- Common settings (variables and DNS)
- Serial interface settings

2. Search interface

For PTools to work we have to first select search interface – this is interface over which PTools will try to find all available interfaces.

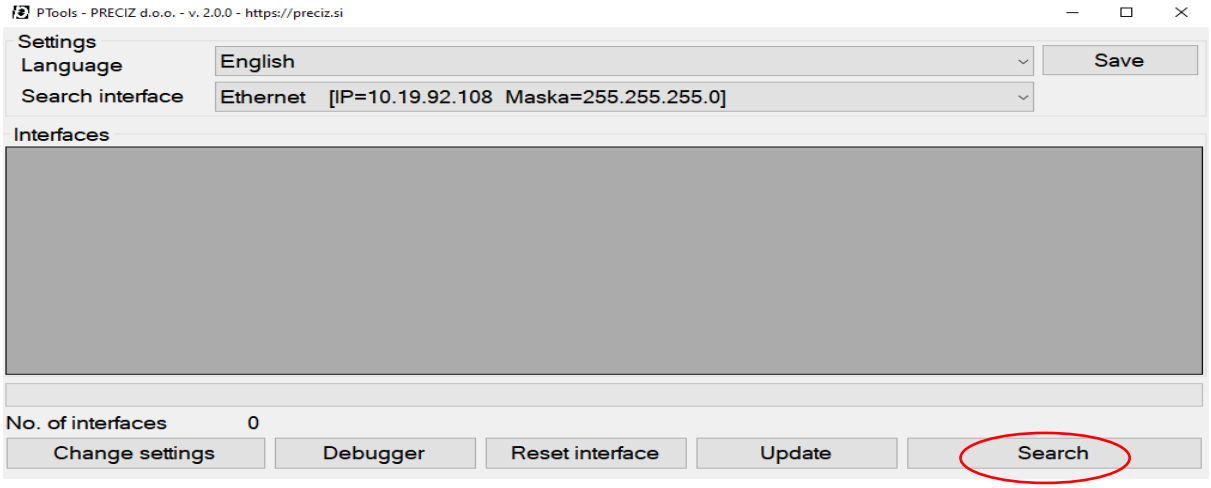
For instance, if you have desktop PC you usually have only one Ethernet card so there will be only one option. If you have a laptop you usually have WIFI and Ethernet interface – but with laptop we usually use WIFI so in that case we have to selected WIFI interface as search interface.

Please note that only working interfaces (this are interfaces with active connection) are shown.

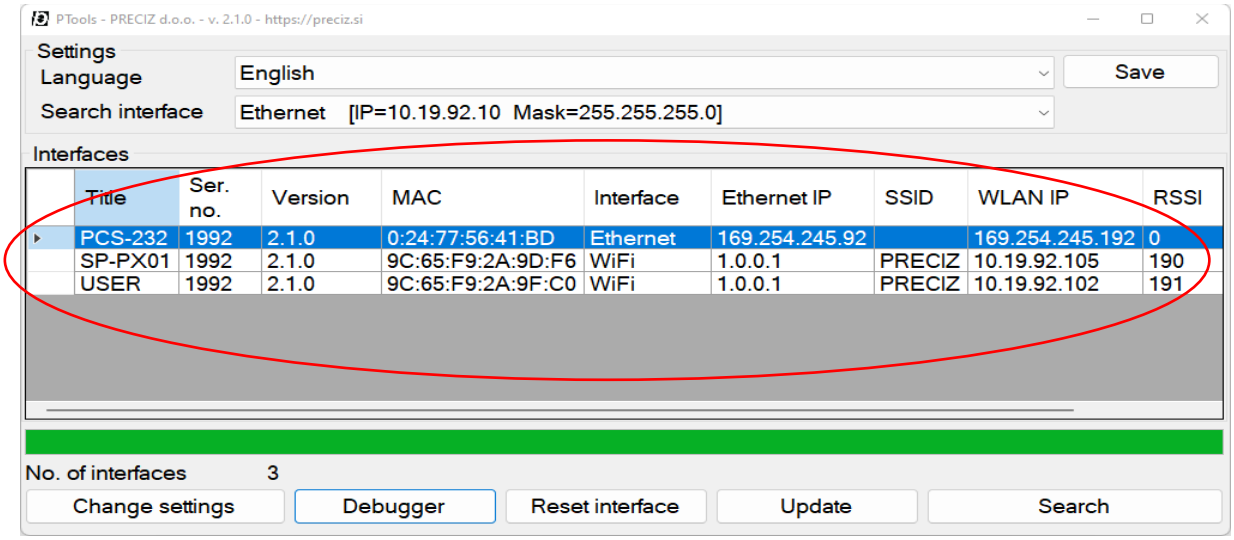


3. Searching for interfaces

After you selected interface on which you wish to perform search you should press Search button.



After that search process will start and progress is shown in a progress bar. All found interfaces are displayed under interfaces:

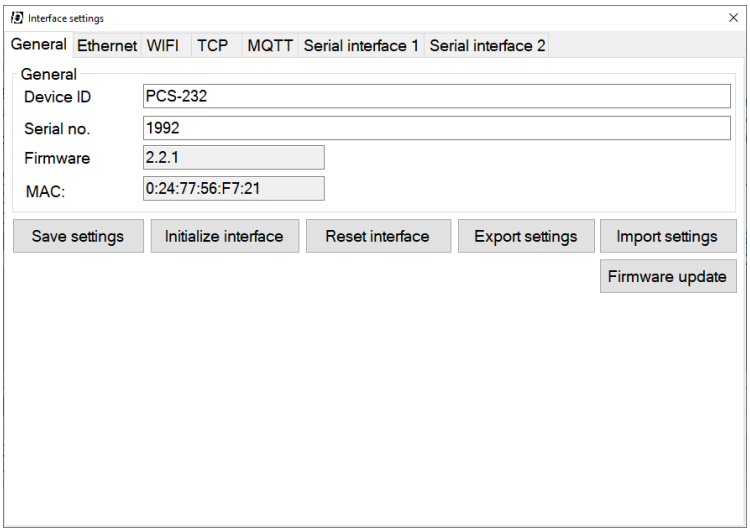


Where also common data of interface are displayed:

1. Device name
2. Ser. Number of interface
3. Firmware version
4. MAC of interface
5. Interface over which device was found (Ethernet or WIFI)
6. Ethernet IP
7. SSID
8. WIFI IP
9. RSSI – signal strength for WIFI
10. And status, there is active connection on the interface (TCP or MQTT)

4. Configuration

After you see interface which you would like to configure click on it and selected device will be blue. After that you can press Change settings button and new window will open where you can configure dedicated settings.



4.1. General settings

In General setting tab we can set the following settings:

- Device ID: name/id of device to easier distinct different devices.
  NOTE: that this ID is also used for DHPC where device id will have suffix -E for Ethernet and -W for WIFI.
  NOTE: device id is also used in JSON response for HTTP/REST
  {"deviceid":"PCS-232","data":""}
- SERIAL NUMBER: We can set the serial number manually.
- MAC: the MAC address of interface
  NOTE: MAC address is different for every interface so for Ethernet is different than for WIFI interface.

4.1.1. Factory reset – initialization of default settings

In general we can also initialize settings to factory default by pressing button Initialize interface.

4.1.2. Import/export settings

Via PTools program you can export and import configuration settings.

If you want to save your settings you just click on Export save settings button in General tab and save dialog will open up where you can specify file name and path where to store it.

NOTE: All settings except WIFI and MQTT password are saved.

If you want to import settings click on Import button and open file dialog will open up. Select XML file from which you would like to import settings.

NOTE: Settings are not automatically sent to interface. To save settings on interface you must click on button Save settings in every tab you want to save settings to interface.

4.1.3. Reboot – reset of interface

Interface can be rebooted via app by pressing button reset interface.

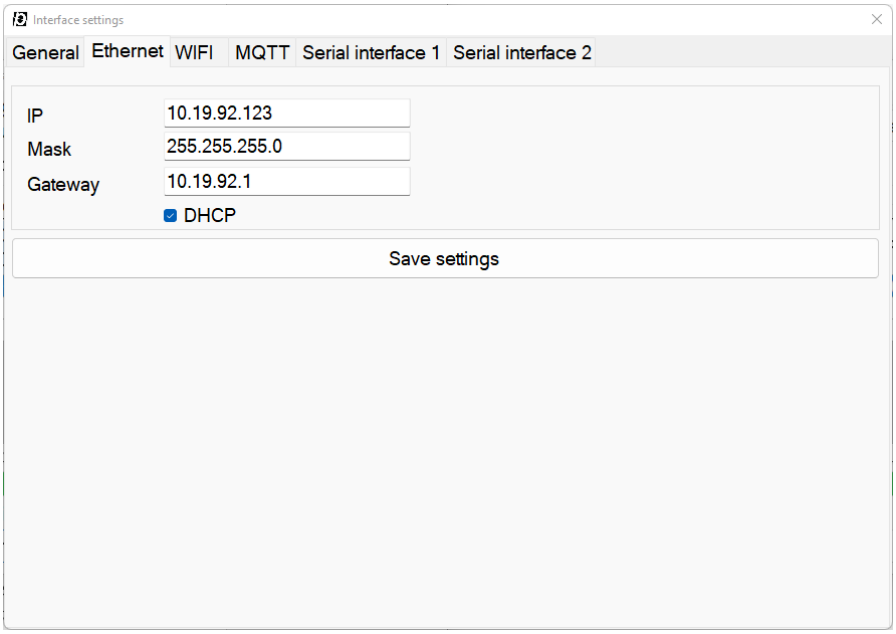
4.2. Ethernet configuration

Under the Ethernet tab we can set:

- IP address,
- Mask
- Gateway

or we can enable DHCP.

To apply changes you must click on Save settings button. Interface will be automatically rebooted.



4.3. WIFI configuration

Under the WIFI tab we can set static or dynamic IP:

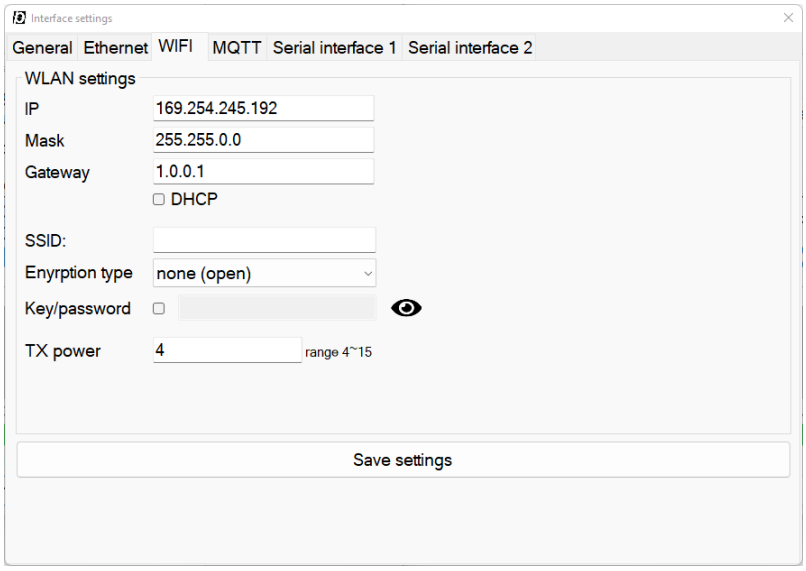
- IP address
- Mask
- Gateway

or we can enable DHCP for dynamic IP.

For WIFI we must enter:

- SSID
- Select encryption type, if WIFI is not OPEN
- Enter key/password

To apply changes you must click on Save settings button. Interface will be automatically rebooted.



NOTE: if SSID is left blank the WIFI will not be enabled.

NOTE: if you want to change password you have to check checkbox before input field.

4.3.1. WIFI encryptions

You can choose between four types of encryption:

1. Open – encryption disabled
2. WEP64
3. WEP128
4. WPA – tkip algorithm
5. WPA2 – aes algorithm

We strongly advise to use WPA2 encryption.

In case you use WEP64 you must enter passphare in hexadecimal format which must be 10 characters long.

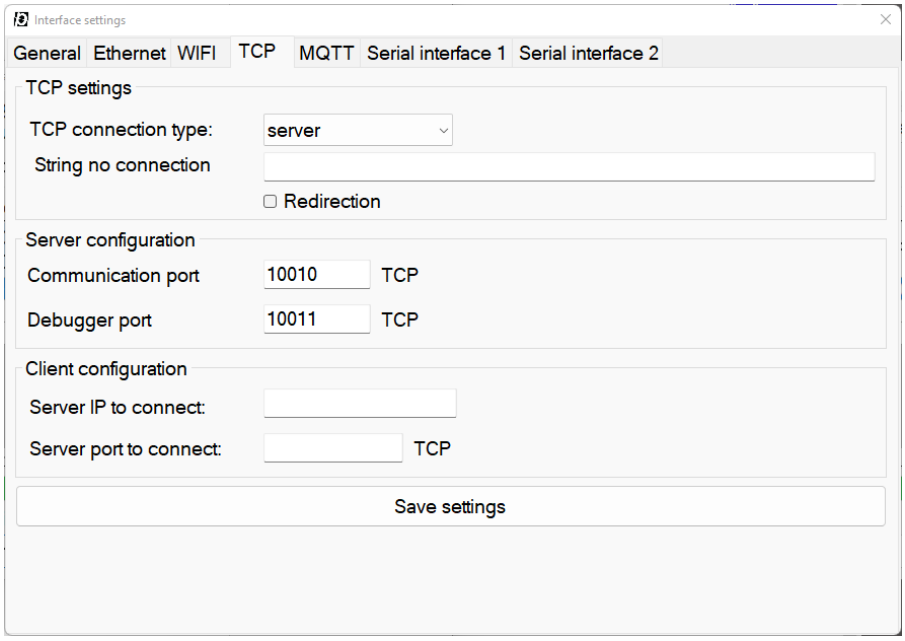
In case you use WEP128 you must enter passphare in hexadecimal format which must be 26 characters long.

Please note that WEP64/128 encryption will only work with one static key (not dynamic where keys are changed).

4.4. TCP configuration

Under the TCP tab we can configure TCP settings. First we have to define TCP type of connection.

Interface can present itself as server (programs establish connection with interface) or as client – interface establish TCP connection with program.



4.4.1. TCP mode (server or client)

For server mode we have to specify TCP port on which interface will be listening – by default this is 10010.

If we want that interface establish TCP connection we have to first select type of connection which can be client (Ethernet) or client (WIFI) – if we select client (Ehernet) connection will be established over Ethernet interface or if we selected client (WIFI) connection will be established over WIFI interface.

In client mode we have to enter IP and TCP port to which interface will connect.

Interface also monitors connection so, if connection is lost for instance WIFI signal is lost or server is down – interface will automatically try to reconnect until connection is established again.

4.4.2. String no connection

Interface is monitoring connection status over TCP and MQTT. In case connection is lost it will send over serial interface string to device so device will be notified that connection was lost.

4.4.3. Redirection TCP/serial

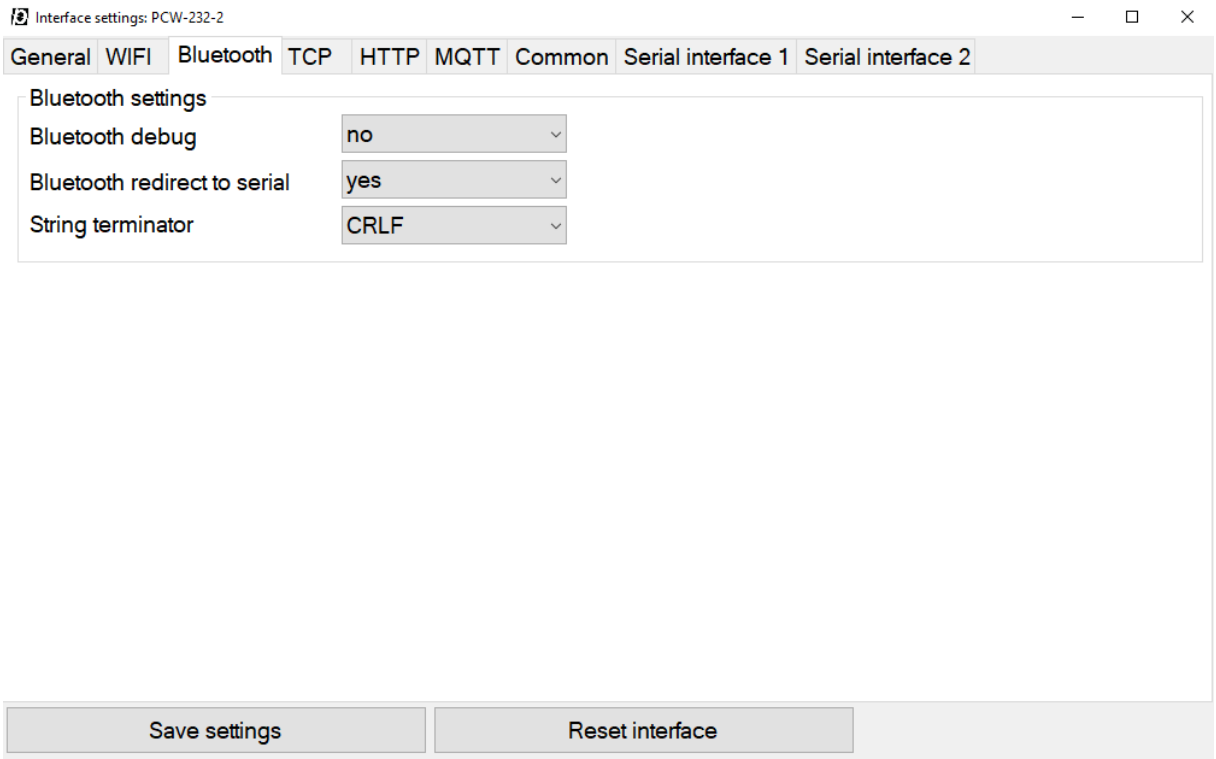
We can enable redirection which is meant to directly pass data received on socket to serial interface and data received on serial interface to socket. This is useful for heavy duty applications to reduce time for processing data.

By default this option is disabled. If you enable this option debugger mode where you can monitor traffic between serial interface and socket will not work as CPU does not process data.

When you enable MQTT this option is automatically disabled in interface because MQTT does not support redirection.

4.5. Bluetooth

For interfaces which support Bluetooth one can set Bluetooth debugging to debug over Bluetooth or to use Bluetooth for communication instead of Ethernet or WIFI.

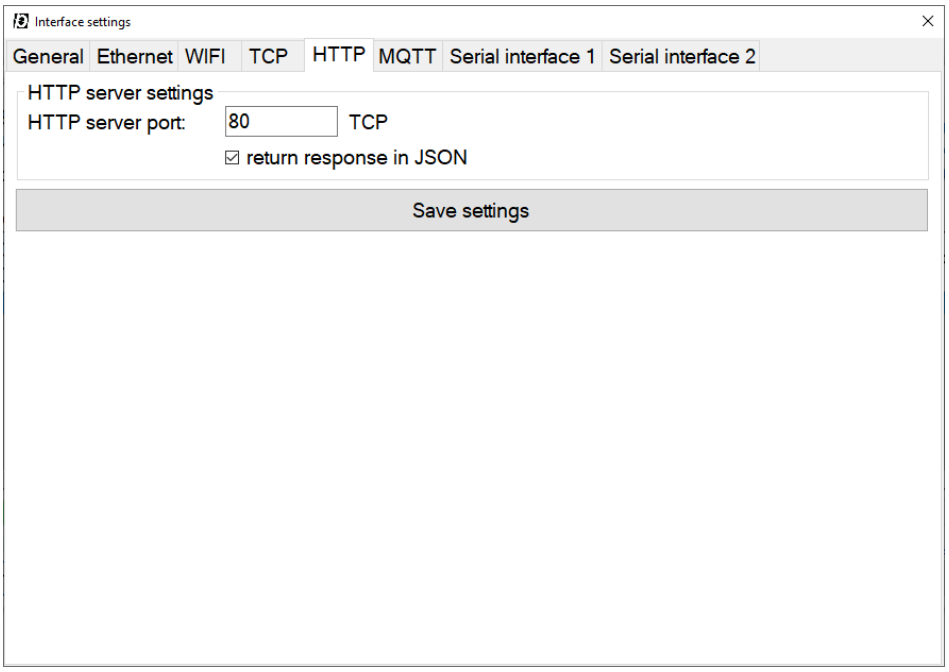


Note: Bluetooth can be used also for configuration of interface.

4.6. HTTP configuration

Under HTTP tab we can configure HTTP server port and response format where you can set that response will be changed to JSON format.

In case you set response to be in JSON format the format will be: {“deviceid”:”your_device_id”, “data”:”response”}



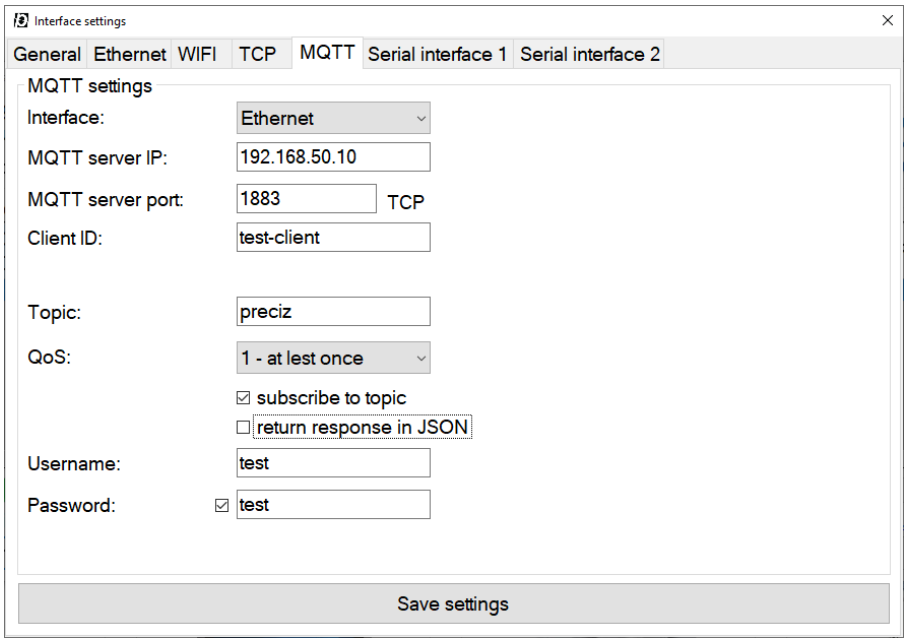
4.7. MQTT configuration

Under the MQTT tab we can configure interface to connect to MQTT server/broker.

First we must select interface over which connection should be performed this can be Ethernet or WIFI.

Then we have to enter:

- MQTT broker/server IP/host/url
- TCP port
- Client ID
- Topic
- QoS – quality of service
- If you want to receive messages on topic you have to check checkbox subscribe to topic.
- If you want that interface encapsulate returned string into JSON format you must check checkbox return response in JSON – returned response will be: {“deviceid”:”your_device_id”, “data”:”response”}
- If broker/server you are connecting to require authentication you have to enter also username and password.



4.8. Common (variables and DNS)

In tab common you can set specific DNS – if you enter under MQTT server url interface detects automatically that it needs to resolve it to IP therefore DNS specified will be used. If not DNS is specified and URL is entered interface uses Google DNS 8.8.8.8 to resolve IP.

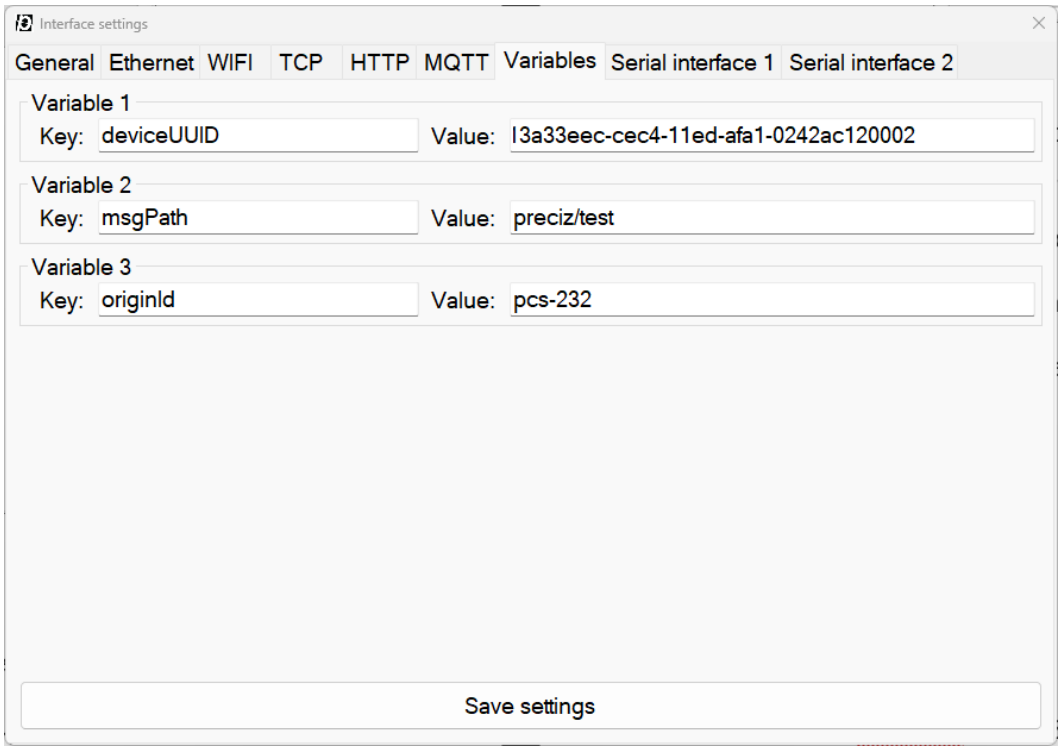
If you need to send additional data with string you can use variables. There are three variables available where you can specify key and value.

NOTE: maximum length of key is 24 characters and maximum length of value is 48 characters.

If key is specified than variable is automatically included in JSON response.

NOTE: you must check return response in JSON in MQTT or HTTP tab so data from RS-232 device are automatically converted into JSON.

Example:



Returned response in JSON:

{

"deviceid": "Mettler Toledo SICS",

"data": "SIX1 S 2 N N R 0 0 0 1 M 1.0996 1.0200 0.0794 kg",

"deviceUUID": "13a33eec-cec4-11ed-afa1-0242ac120002",

"msgPath": "preciz/test",

"originID": "pcs-232"

}

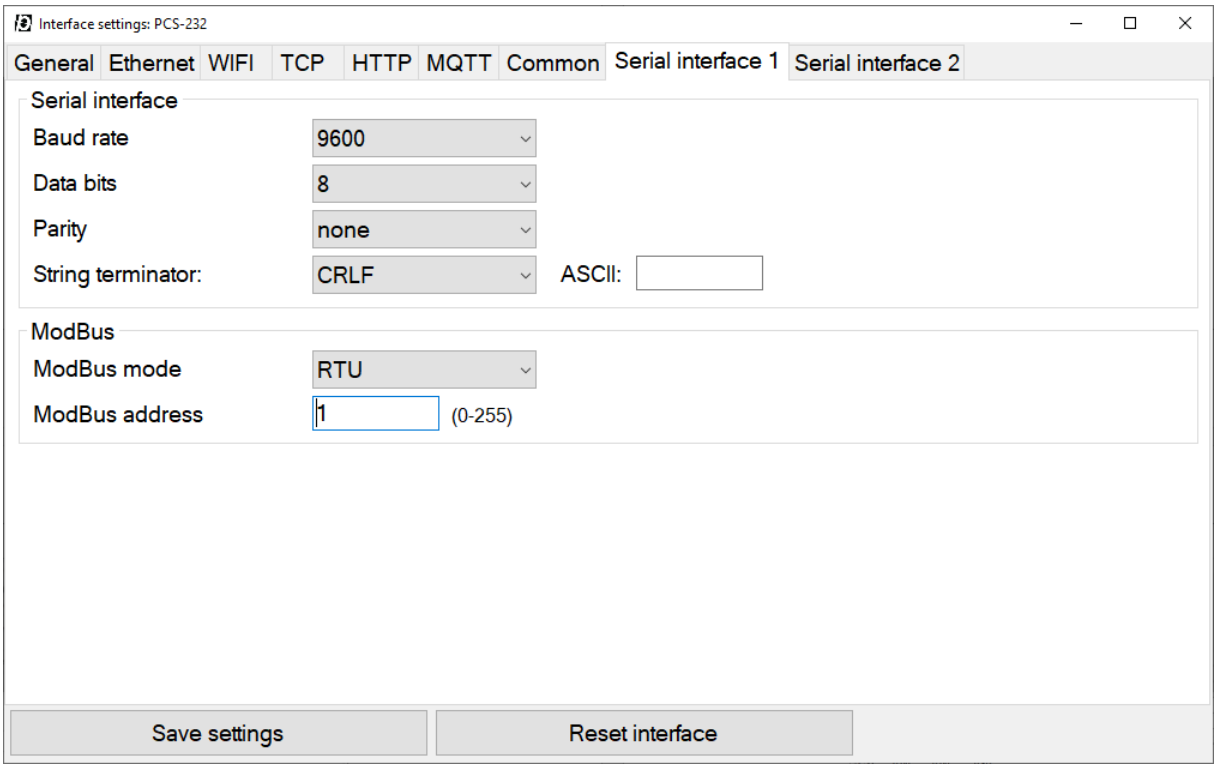
NOTE: maximum length of JSON response is 256 characters.

4.9. Serial interface configuration

Under the serial interface tab we can configure the following serial interface settings:

- Baud rate
- Data bits
- Parity
- String terminator

String terminator is used to determine end of data so it can correctly show data in debugger mode. Usually serial device have string terminator set to CR (carriage return) and LF (line feed) which is also default value. In case your device use different terminator you can insert ASCII character in decimal form for instance for ETX – which is ASCII 3 you should enter 3.



4.10. ModBus

Some interfaces support ModBus communication. If you set ModBus mode to RTU than you can communicate via TCP/IP or MQTT with ModBus device.

Please note that string you send to device must be in HEX format and must include string terminator. When you send command to ModBus device you do not have to calculate CRC checksum also interface does automatically check, if response checksum is valid, if not you will not get any response.

Example of command to be sent to device:

01030000000A

Byte	Value	Description
1	01	Modbuse device address
2	03	Modbus function
3	00	Starting register MSB
4	00	Starting register LSB
5	00	No. of registers MSB
6	0A	No. of registers LSB

In above example we send command to ModBus device with address 1, we execute function 03 to read holding register AO from register 40001 and we want 10 registers (#0A).

Response will be like this:

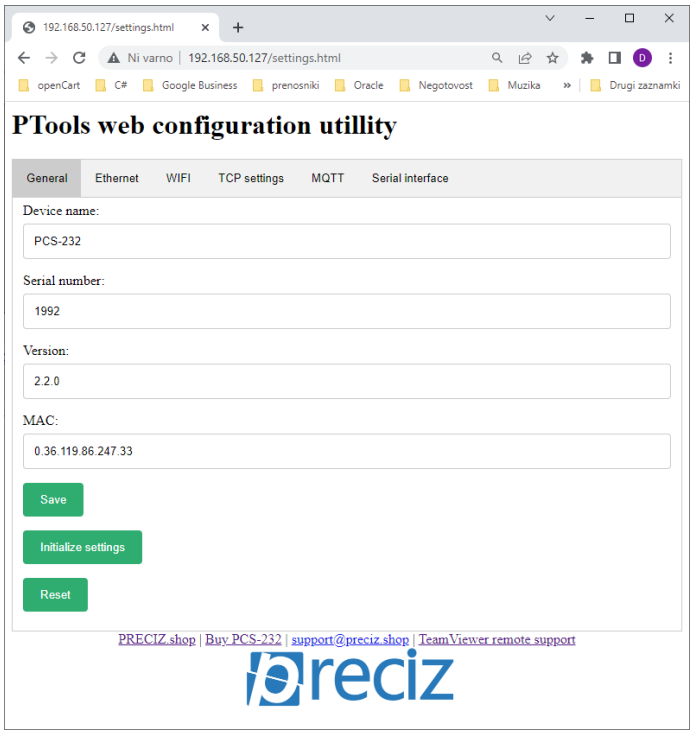
0103140000202B0000202B0004030F0000000000000000

5. Default settings

Ethernet IP:	169.254.245.92
WIFI IP:	169.254.245.192
Subnet:	255.255.0.0
DHCP:	Disabled
TCP Server port:	10010
TCP Debugger port:	10011
DNS	8.8.8.8
Redirection:	disabled
Baudrate:	9600
Databits:	8
Parity:	None
String terminator:	CRLF (ASCII 13 and 10)

By default Ethernet interface has IP: **169.254.245.92** which is APIPA – automatic private IP addressing.

You can configure interface also via web browser by visiting: http://169.254.245.92/settings.html if you connected over Ethernet or http://169.254.245.192/settings.html if you connected over WIFI.



NOTE:

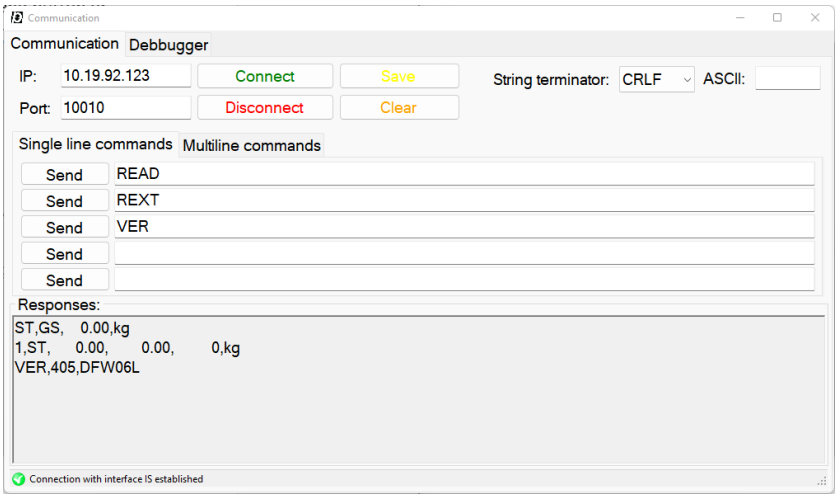
You can also connect interface directly to computer via Ethernet cable. In PTools you select ethernet interface to which you connected and press Save and after that Search button which is the easiest and fastes method.

6. Communication and debugging

Communication/debugging is used to quickly test, if interface is working – you just select interface and click Debugger button. After that new window is opened where you have two tabs:

- Communication – it is used to test and simulate commands.
- Debugger – it is used to monitor traffic between interface/device and program.

If you wish to send command you first press Connect (status of connection is shown in bottom left corner). After connection is established you can put into textbox command and press Send button or press Enter. Responses are shown in response field. If device is continuously sending data they will be immediately shown in responses field after connection is established.

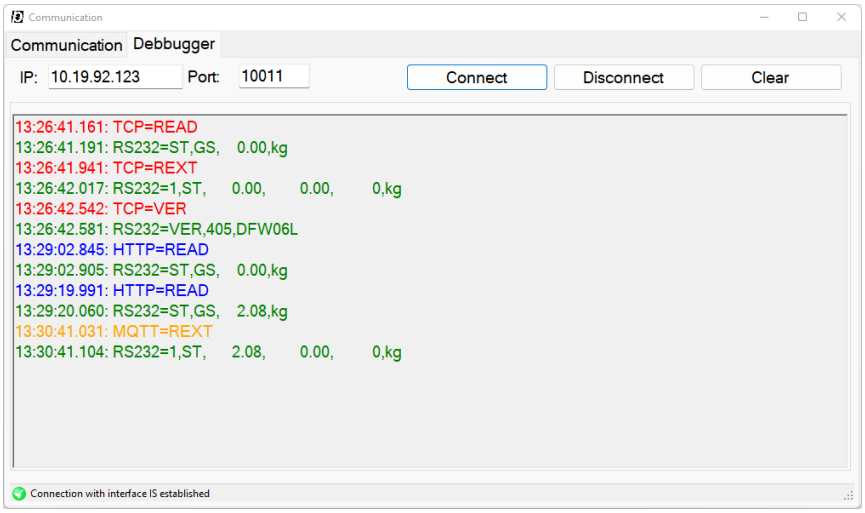


If you wish to monitor traffic you can click on Debugger tab and click Connect (status of connection is shown in bottom left corner).

NOTE: if you wish to use debugger – redirection must be disabled otherwise connection is refused.

You will see which data was send and over which socket was it send.

RS232 is the response from the device while all other data TCP, HTTP and MQTT are received string to interface.



7. Firmware update (OTA-Over The Air)

You can update firmware by selecting interface and after that pressing button update. You have to select file with new firmware and after that press upload button to start process.

After update process is finished you will get message with information if process was finished successfully or not. If process failed (eg. WIFI signal lost) you can restart process.