PV Communicator Config Tool

User Manual

Language: English

Contact: lancz@emineo.sk

Note: This document is based on screenshots, practical installation notes, and final recommendations for field users.

Introduction

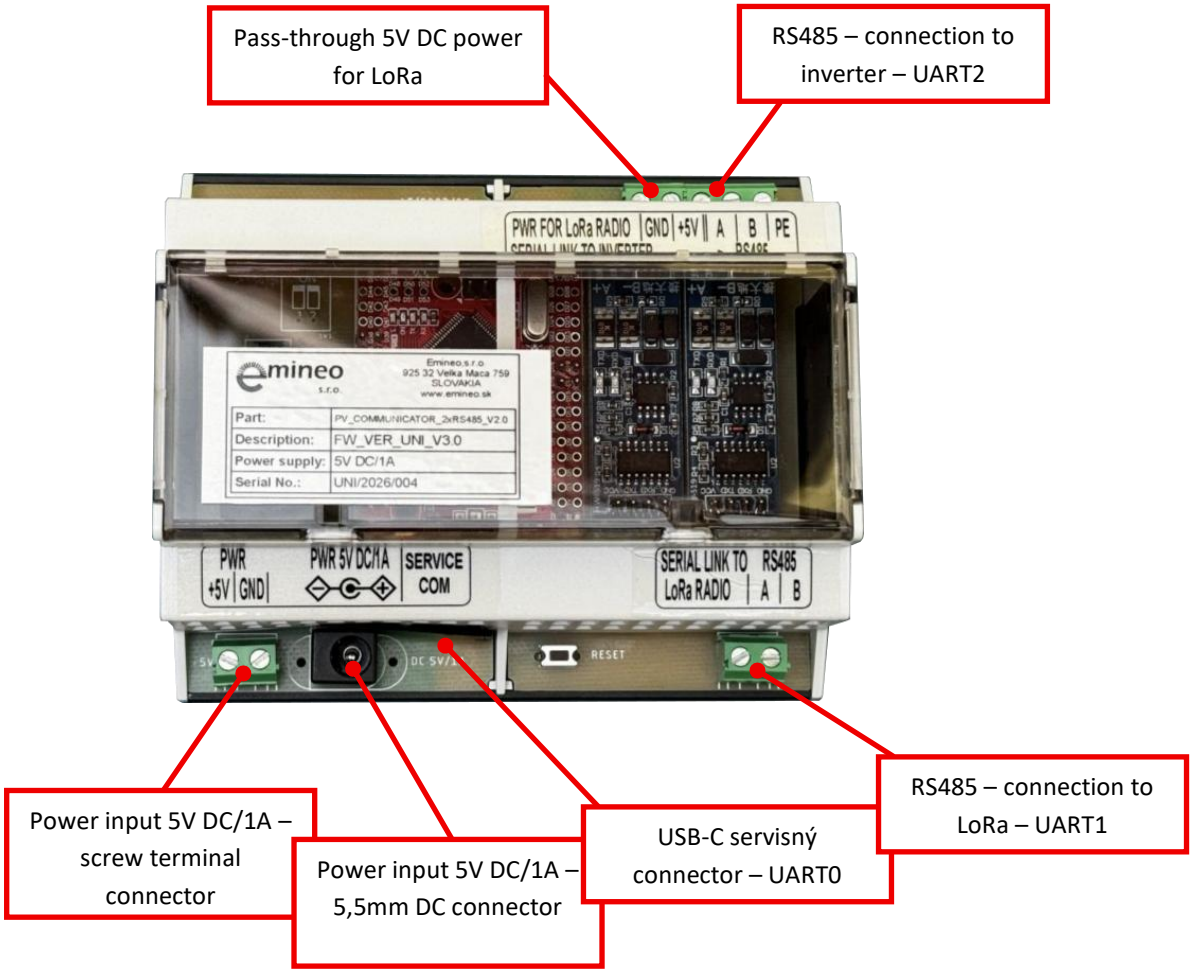
The device you have purchased is generally intended to enable data communication between smart meters and photovoltaic inverters by means of LoRa radio modules operating in the license-free telemetry band.

Warning!

Please note that every such converter introduces a certain delay into the communication path (on the order of hundreds of milliseconds). For this reason, this type of connection cannot be considered a 100% replacement for a standard metallic RS485 communication link. The suitability of such a connection must therefore be carefully assessed for each specific installation site.

The transmitted output power of the LoRa communicator must be set in accordance with local regulatory requirements.

Device Wiring



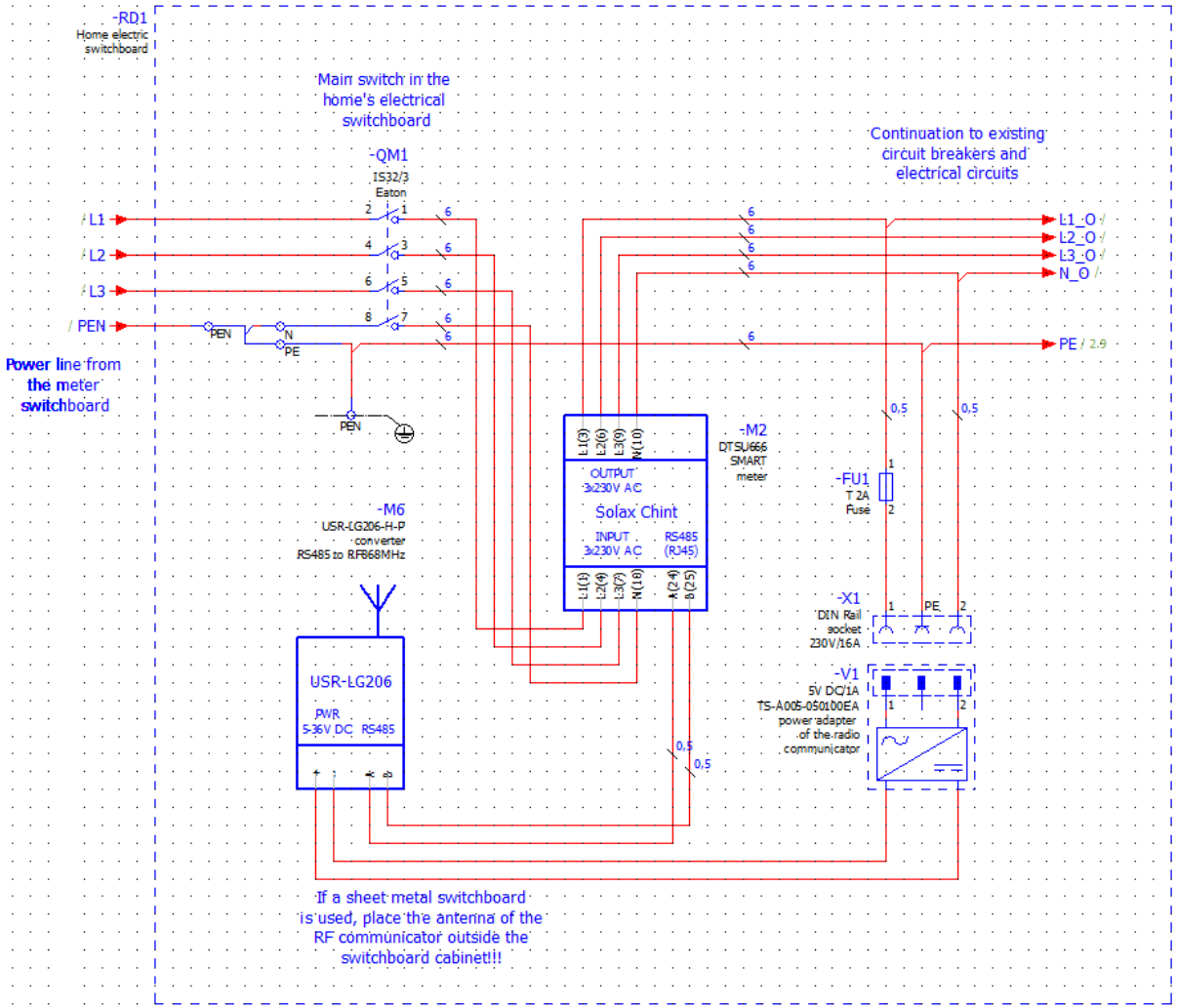
The device is generally equipped with screw terminals. The power inputs (two-pole screw terminal vs. 5.5 mm DC connector) are wired in parallel. This allows the user to use either the 5 V DC / 1 A plug-in adapter supplied with the LoRa module, or for example an external power supply with a 5 V DC / 1 A output.

The LoRa module can be powered from the pass-through power terminals, which means that in practice the same power supply can be used for both devices.

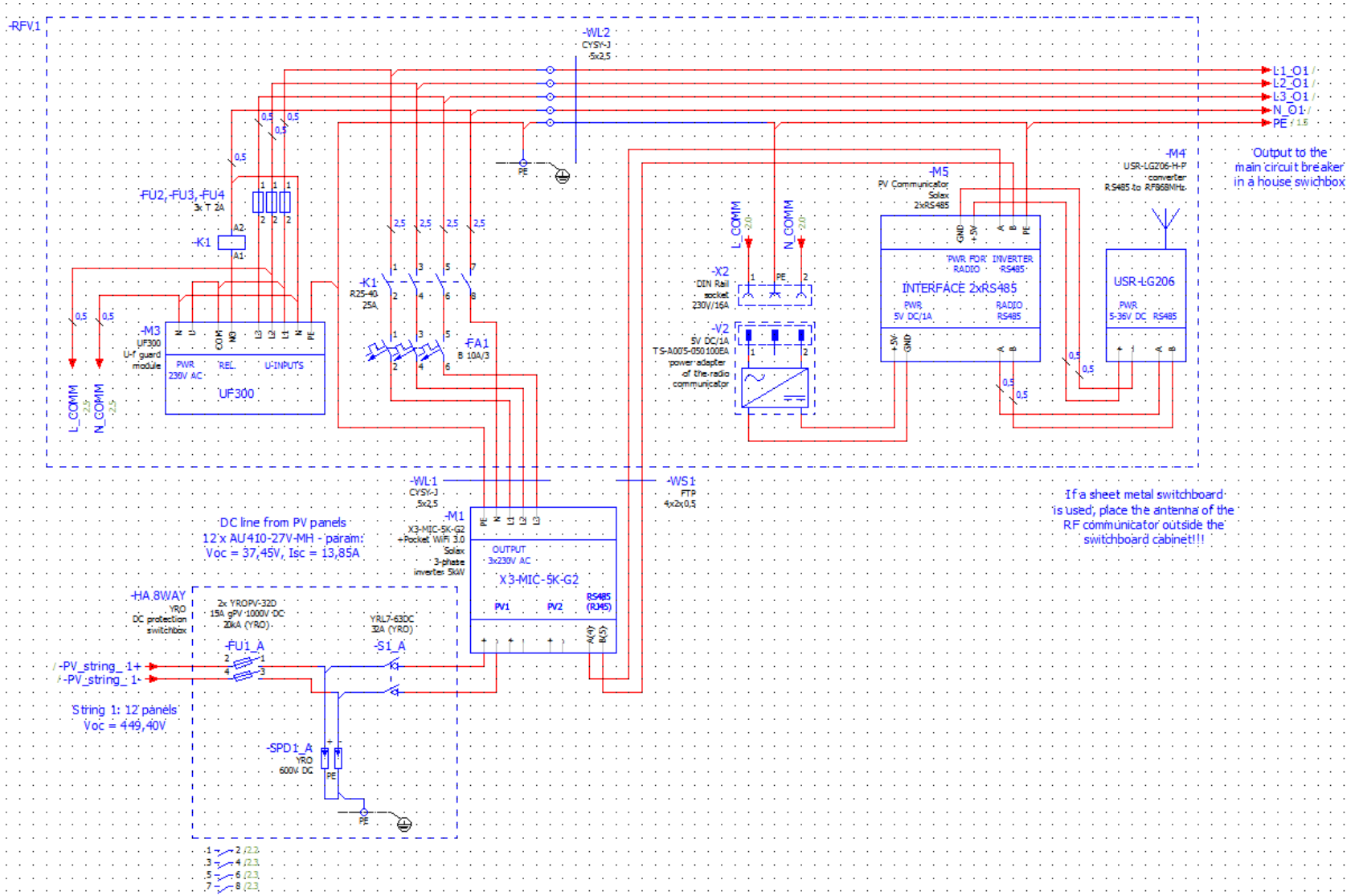
The RS485 connection between the PV communicator and the LoRa module will be short in most cases. However, because the antenna cable of the LoRa module will typically be about 2 to 3 meters long, it is expected that in some cases the PV communicator and the LoRa module may need to be located tens of meters away from the PV inverter in order to achieve optimal antenna placement for the LoRa module. In such cases, especially when the RS485 communication link between the inverter and the PV communicator is routed in parallel with other cables, it is recommended to use suitable surge protection and 120 ohm termination resistors on the RS485 line.

The service connector (used to connect the PV communicator module to the configuration software) is USB-C type; in exceptional cases it may be micro USB. It is located under the device cover and accessible through the opening next to the 5.5 mm DC power connector.

Connection example – LoRa module on the smart-meter side.



Connection example – LoRa module and PV communicator on the inverter side.

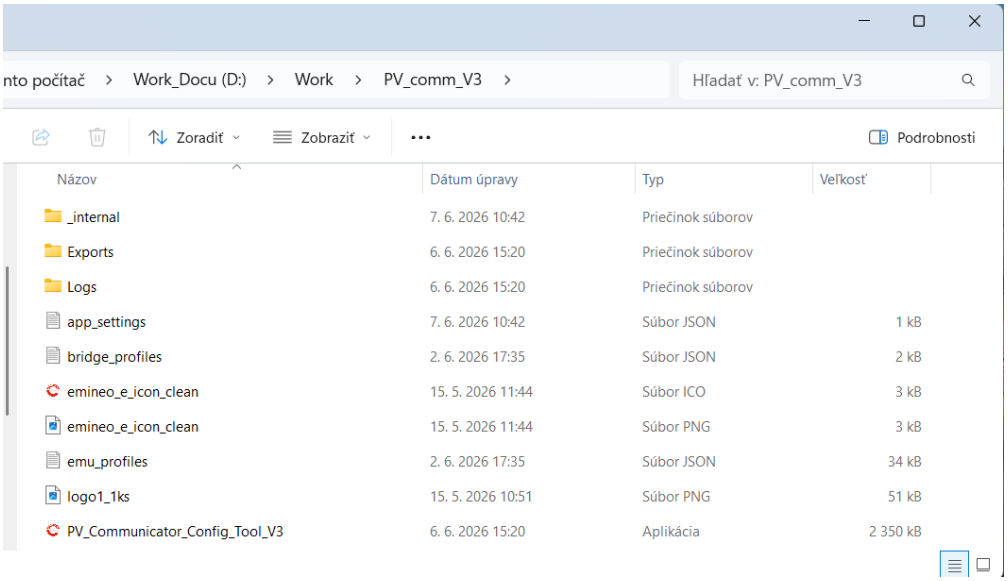


Warning!

- For every installation, considering the location of the distribution boards in which these components are connected, consider the use of suitable protective devices and surge protection.
- Pay attention to correct antenna placement, both for communication range and for lightning protection in outdoor installations.
- Before the installation itself, it is advisable to verify the range, i.e. whether the LoRa modules can see each other. Try to use the lowest transmission power possible while remaining compliant with local regulatory requirements.
- For some inverters, it is necessary, even before the profile learning process, to first detect the smart meter in the inverter configuration while it is still connected through a metallic link. Only after the smart meter is identified does the inverter start communicating with it. Only then is it possible to start the inverter profile learning process. Always make sure that the learning process is first initialized in the GUI of the PV communicator configuration software, and only then power on the inverter itself. This makes it possible to capture also the first communication segments that some inverters send to the smart meter only at startup.

Program Installation

PV Communicator Config Tool is distributed as a portable version. No installation into Windows is required.



- always extract the application first
- _internal is a mandatory part of the package

First start procedure:

1. Download the ZIP archive with the application to your computer.
2. Do not run the application directly from the ZIP archive.
3. Extract the entire archive into a separate folder.
4. Check that the folder still contains the _internal subfolder. This folder is part of the application and is required for correct startup.
5. Run the file PV_Communicator_Config_Tool_V3.exe.
6. At first start, select the correct service COM port of the PV communicator.
7. In the Workspace tab, click the Status button and verify that the application is communicating with the PV communicator correctly.

Important notes:

- The files app_settings.json, bridge_profiles.json and emu_profiles.json contain the application user settings and profiles.

After changing settings, these values are saved to the listed files automatically.

- we do not recommend running the portable version from Program Files
- because of profile and settings writes, it is better to run it from a normal data folder

PC requirements and device connection

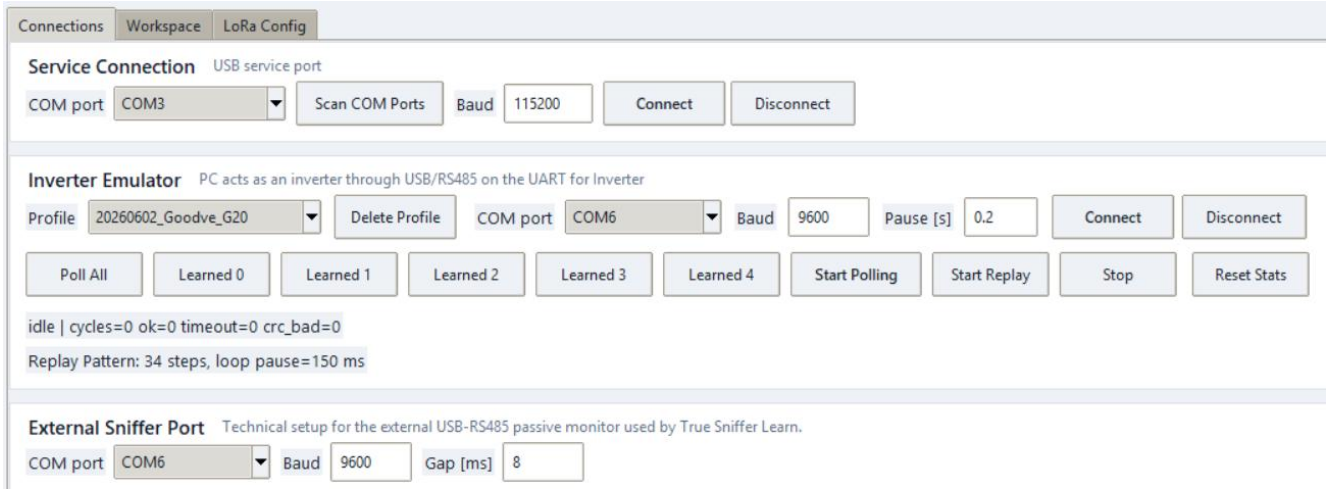
The application is provided as a portable version for the Windows operating system.

Minimum requirements

- a computer running Windows
- a free USB port for the PV communicator service connection
- if needed, an additional USB port for the emulator RS485 converter or the LoRa converter

Communication ports used by the PV communicator

- UART0 = PV communicator service port
- UART1 = LoRa port
- UART2 = inverter port



Typical connection when working with the application

- connect the PV communicator service port to the PC via USB
- if you use the Inverter Emulator, connect its RS485 converter to the PC through another USB port
- if you use LoRa converter configuration, connect the LoRa module to the PC according to the selected service procedure

Recommendation before first use

- after connecting the devices, first check which COM ports Windows assigned
- in the application, use Scan COM Ports
- select the correct PV communicator service port
- click Status and verify that communication is working correctly

Note on learning profiles

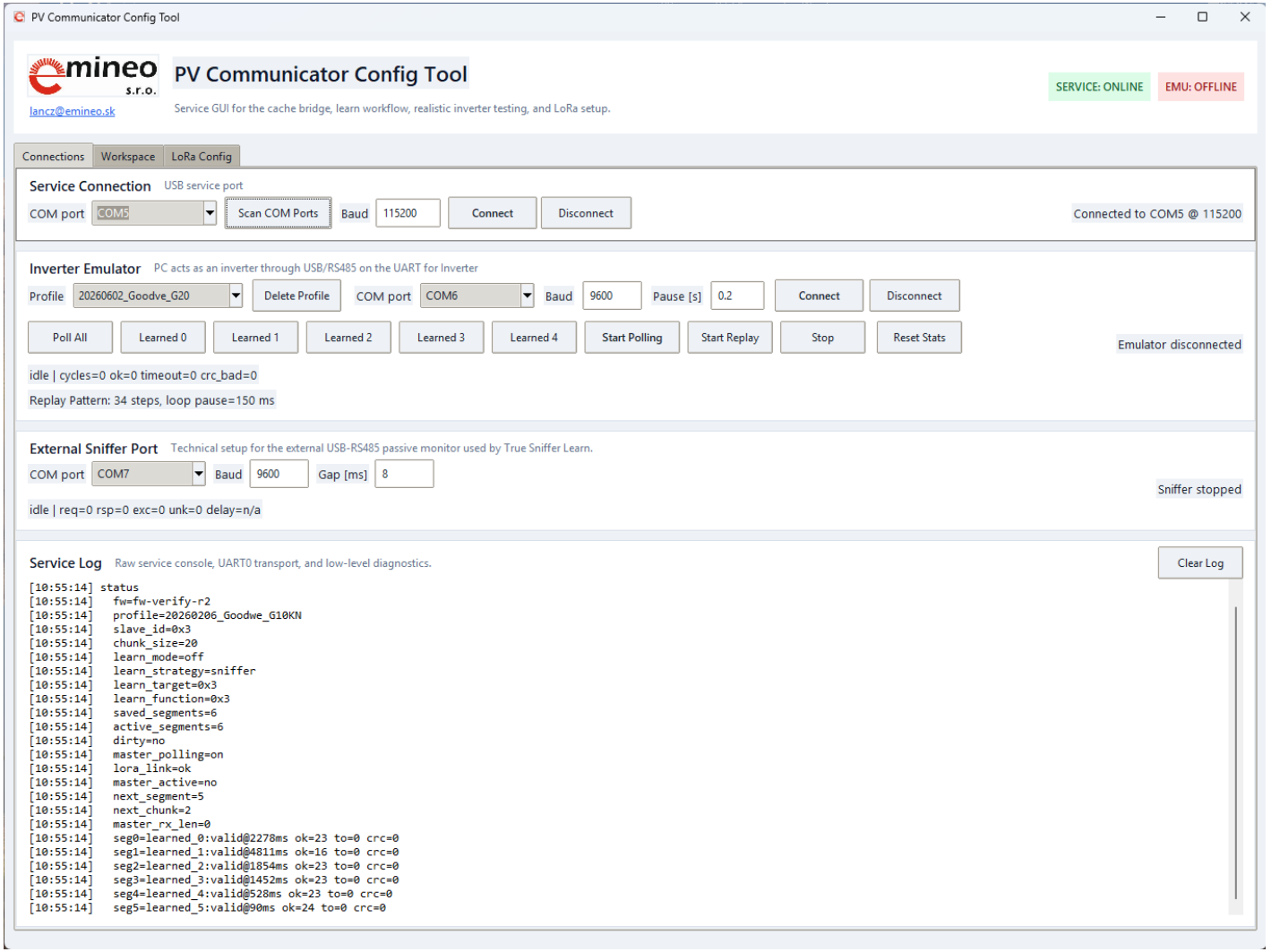
When using Transparent Learn or Sniffer Only, the test or live setup must be prepared correctly for the selected learning mode. Before starting the learning process, it is recommended to verify that service communication with the PV communicator works without errors.

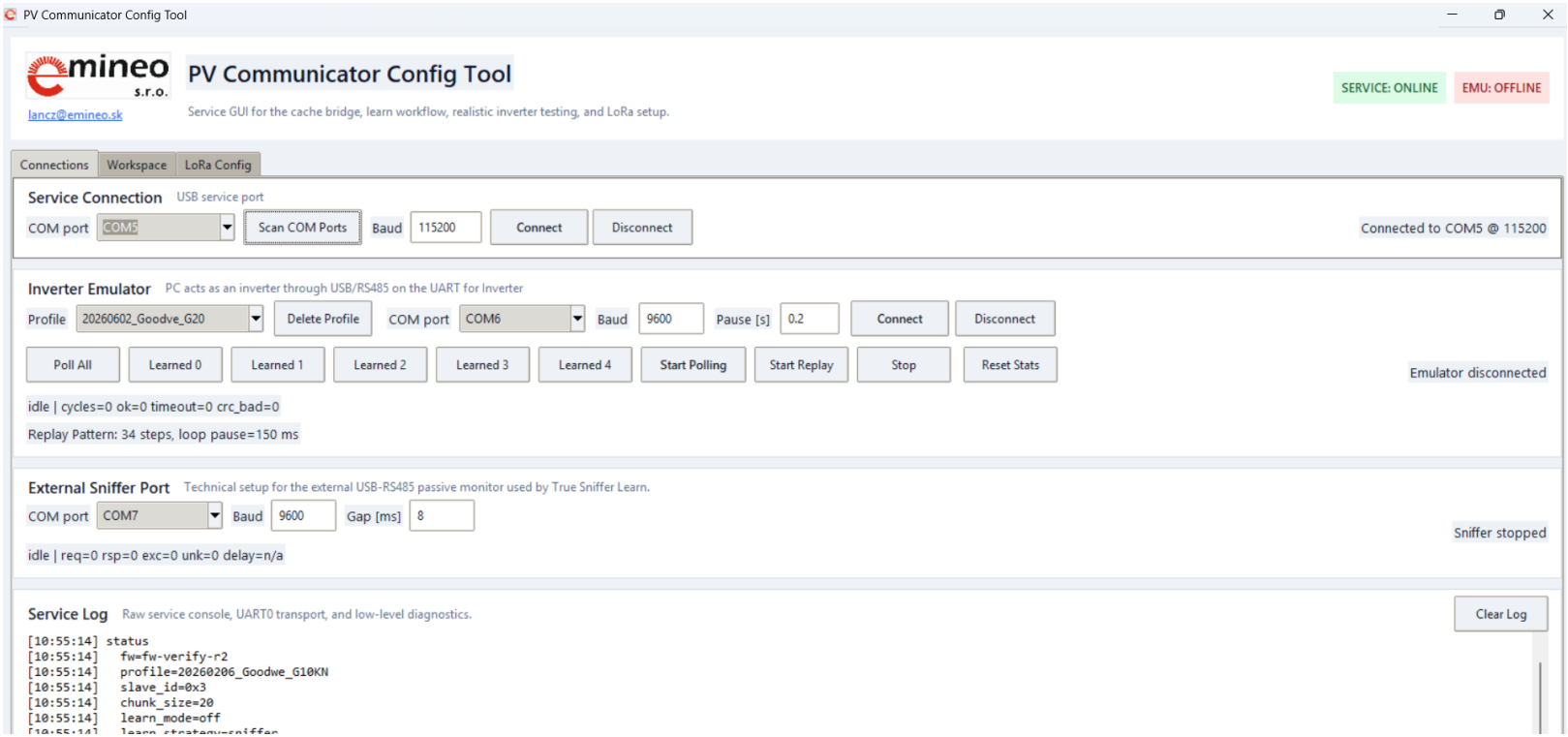
Basic orientation in the application and description of individual sections

The PV Communicator Config Tool application is divided into three main tabs:

- Connections
- Workspace
- LoRa Config

Each tab is intended for a different group of tasks.





Connections
The Connections tab is used for:

- selecting and connecting the PV communicator service COM port
- working with the Inverter Emulator
- configuring the External Sniffer Port - it is used in one of the communication-segment learning modes
- monitoring the Service Log - various status and diagnostic messages

This tab is typically used for:

- connecting the PV communicator to the PC
- selecting an emulator profile, starting or stopping communication emulation
- selecting the Sniffer Port for direct capture of communication on the metallic link between the inverter and the smart meter

Inverter Emulator

The Inverter Emulator section is used for lab testing of the PV communicator without a real inverter. The PC acts here as an emulated inverter over USB/RS485 on the UART for Inverter side and sends requests to the communicator according to the selected profile.

Meaning of individual elements

- Profile - selects the emulator profile according to which requests will be sent
- Delete Profile - deletes the selected profile from the emulator profiles
- COM port - selects the serial port through which the emulator is connected to the PV communicator
- Baud - emulator communication speed, usually 9600
- Pause [s] - time pause between polling cycles during cyclic testing
- Connect - connects the emulator to the selected COM port
- Disconnect - disconnects the emulator from the COM port

Test buttons

- Poll All - sends all requests defined in the selected profile one after another
- individual segment buttons, for example Meter Status, Control Block, Live Block or Learned X - send only the selected request segment
- Start Polling - starts repeated cyclic polling of all requests in the profile
- Start Replay - starts a replay pattern, that is, a more realistic repeated request sequence according to the prepared pattern
- Stop - stops cyclic polling or replay
- Reset Stats - resets emulator statistics

Status information

- status line, for example idle, connected, polling or replay - shows the current operating mode of the emulator
- Replay Pattern - indicates whether the profile has a replay pattern defined and how many steps it contains
- statistics such as cycles / ok / timeout / crc_bad - show the number of cycles, successful responses, timeouts and CRC errors

Difference between Polling and Replay

- Poll All or Start Polling use simple sequential requests according to the profile
- Start Replay uses a prepared, more realistic pattern that better imitates the behavior of a real inverter

In practice:

- Polling is suitable for basic verification of profile function
- Replay is suitable for more demanding communication and stability testing

Why the Inverter Emulator is useful in practice

- it allows testing of the PV communicator without a physically connected inverter
- it helps verify whether a profile works correctly
- it is suitable for replay tests and for simulating more aggressive polling
- it allows comparison of communicator behavior with different profiles and settings

What is considered a normal state

A normal and healthy state is when:

- the emulator connects correctly via Connect
- requests receive responses without timeouts
- the ok counter increases
- timeout and crc_bad remain low or zero
- replay or polling runs stably

What may already be suspicious

Special attention is needed when:

- the emulator does not connect to the COM port
- requests repeatedly end in timeouts
- crc_bad increases
- the profile in the emulator does not match the profile in the PV communicator
- replay or polling runs, but the communicator does not respond as expected

Practical recommendation

During testing it is always useful to check:

- whether the correct profile is selected in the emulator
- whether the same or a corresponding profile is also loaded in the PV communicator
- whether the COM port is correct
- and whether the result in Service Log, Comm. Log and Health Overview matches the expected behavior

External Sniffer Port

The External Sniffer Port section is used for technical setup of an external USB/RS485 monitor used in True Sniffer Learn mode. This section does not start capture by itself; it only prepares parameters for passive monitoring of communication on the metallic RS485 line.

Meaning of individual elements

- COM port - selects the serial port of the external USB/RS485 monitor
- Baud - sniffer communication speed, usually according to the monitored line, for example 9600
- Gap [ms] - line idle time according to which the sniffer splits communication into individual packets

What the Gap [ms] parameter means

Gap [ms] defines how long a pause between bytes must be before the sniffer considers the frame complete.

In practice:

- if bytes are still arriving close together, the sniffer treats them as one packet
- if silence longer than the configured Gap appears, the sniffer closes the packet

This parameter does not change inverter or smart-meter communication. It only affects how the GUI splits captured data into requests and responses.

How to choose a suitable Gap

For a 9600 Bd line, a good starting point is for example:

- Gap = 8 ms

The value can then be fine-tuned according to the log output:

- if the request and response merge into one frame, Gap is usually too large
- if one normal packet is split into several short pieces, Gap is usually too small

Typical signs of an incorrect Gap

If Gap is too large:

- the request and response may merge into one frame
- long or unusual frames appear
- unknown or helper [split] messages may increase

If Gap is too small:

- the frame may split into several incomplete parts
- the response or request may look broken

- short incomplete packets may appear in the log

Why External Sniffer Port is useful in practice

- it prepares technical parameters for True Sniffer Learn
- it allows passive monitoring of real inverter communication on the metallic RS485 line
- it is especially useful when the user needs to create a profile from real operation without interfering with communication

Important note

Capture itself is not started in this section.
This section only sets the sniffer technical parameters.
Capture start and evaluation take place only in:

- Workspace -> Learn -> True Sniffer Learn

Practical recommendation

Before starting True Sniffer Learn, it is recommended to check:

- the correct COM port
- the correct Baud
- a reasonable starting Gap
- and only then start the actual capture in the Learn section

Service Log

The Service Log section is located in the Connections tab and displays the raw service transcript of communication with the PV communicator. It is used mainly for service diagnostics, checking command responses, and tracking detailed runtime messages generated while working with the device and the emulator.
Unlike Comm. Log, this is not a simplified workflow log but a more technical output intended for precise monitoring of what the communicator is actually reporting.

What typical messages can appear here

- responses to service commands, for example:
  - status
  - profile info
  - segments
  - master on
  - master off
- confirmations such as:
  - ok: ...
- warnings and retry messages, for example:
  - master retry ...
- diagnostic runtime messages from the communicator
- emulator runtime messages started from the Connections tab, for example:
  - connect
  - disconnect
  - polling
  - replay
  - TX/RX frames
  - timeouts
  - cycle start / done

Why Service Log is useful in practice

- it verifies whether the communicator accepted and executed a service command
- it shows exact responses to diagnostic commands
- it helps when resolving retries, timeouts and other communicator-level problems
- it is important for service troubleshooting when the user needs to know exactly what the device replied

How it relates to Health Overview

Health Overview shows a simplified resulting state, while Service Log shows the detailed background of that state.
In practice:

- if Health Overview shows, for example, a warning, retry or a problem in Last Master Issue, a more precise context can often be found in Service Log
- if Health Overview shows a profile change or a change in polling mode, Service Log confirms by which command and with what response it happened

Simply put:

- Health Overview answers the question:
  - "What does the system look like right now?"
- Service Log answers the question:
  - "What exactly did the device report?"

What is considered normal logging

Typical and expected examples include:

- ok: ... responses
- status output
- profile info output
- segments output
- normal emulator TX/RX messages during testing
- starts and completions of polling or replay cycles

These messages are a natural part of normal operation and service work.

What may already be suspicious

Special attention is needed when:

- repeated retries or timeouts appear
- the communicator does not respond to service commands as expected
- the emulator reports errors or does not receive a response
- the segments output shows invalid segments at a time when the data should already be ready
- Service Log and Health Overview contradict each other

Example:

- Health Overview shows a problem in Last Master Issue
- and Service Log at the same time repeatedly shows a retry on a specific segment
  In such a case, this is a real diagnostic signal, not just a cosmetic deviation.

Relationship to Comm. Log

- Service Log belongs to actions and diagnostics in the Connections tab
- Comm. Log belongs mainly to workflow actions in Workspace

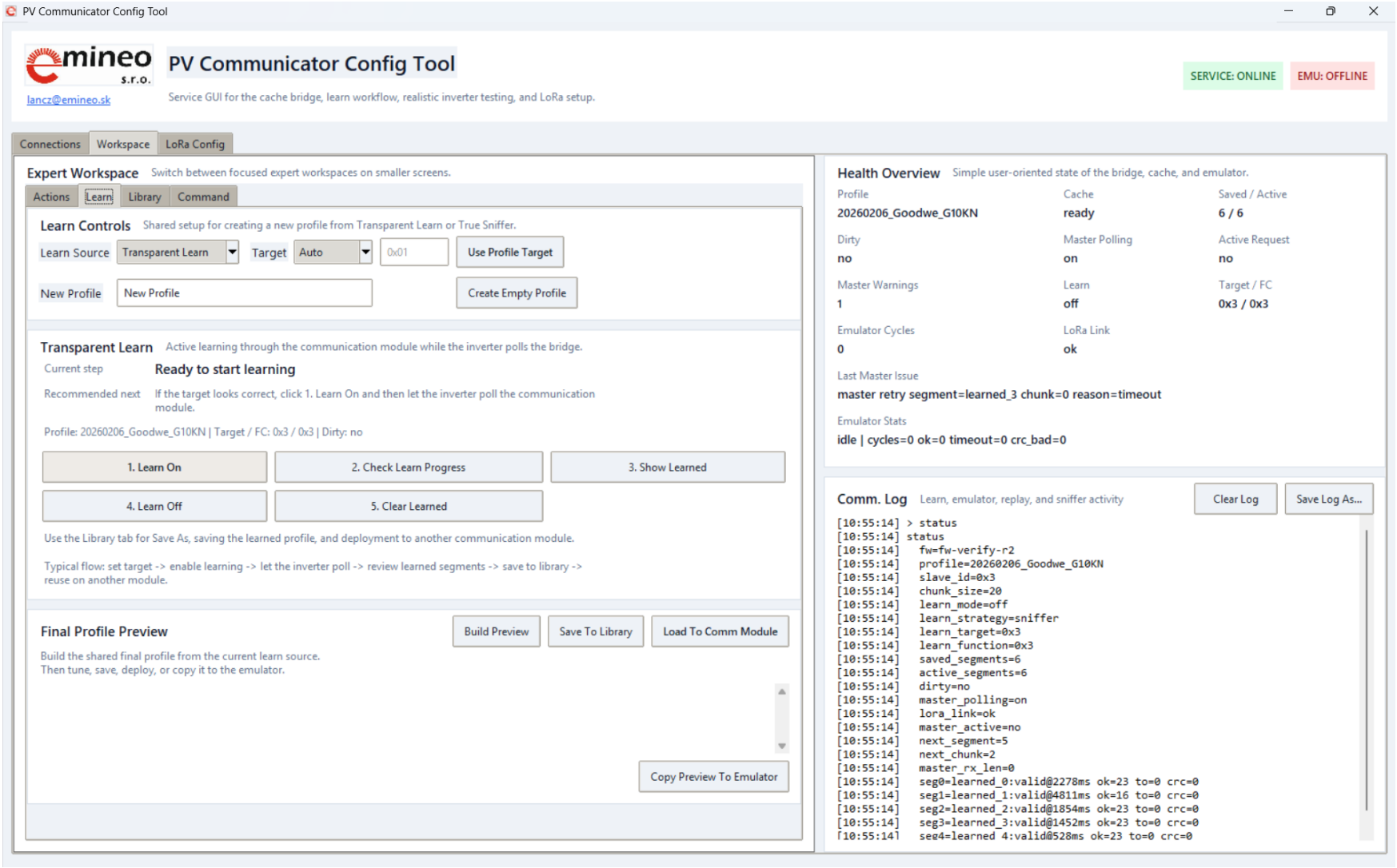
Simple rule:

- if the user works with the service port, emulator or detailed communicator diagnostics, they should mainly watch Service Log
- if they work with the learn workflow, profile library and final preview in Workspace, they should mainly watch Comm. Log

Practical recommendation

During normal work it is useful to:

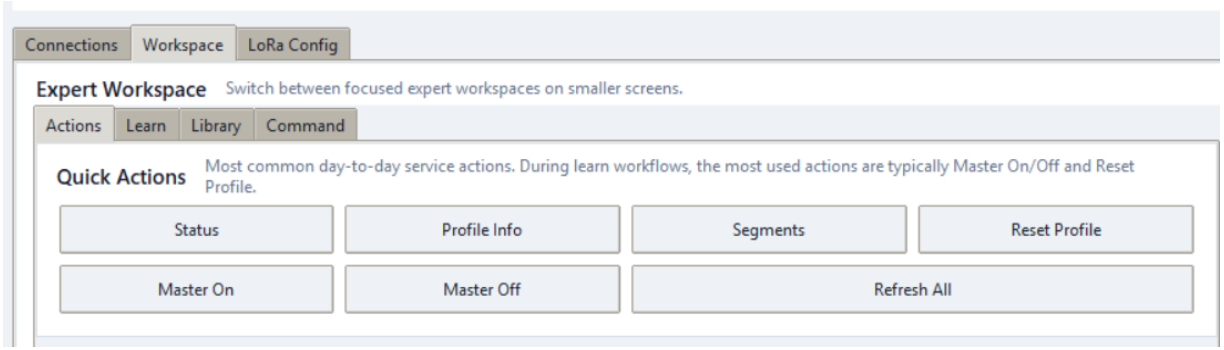
- first orient yourself through Health Overview
- then, as needed, use:
    - Comm. Log for workflow events
    - Service Log for detailed service diagnostics



Workspace Tab

The Workspace tab is the main working window of the application. It is used for:

- quick service actions
- learning new profiles
- working with the profile library
- sending service commands through Command

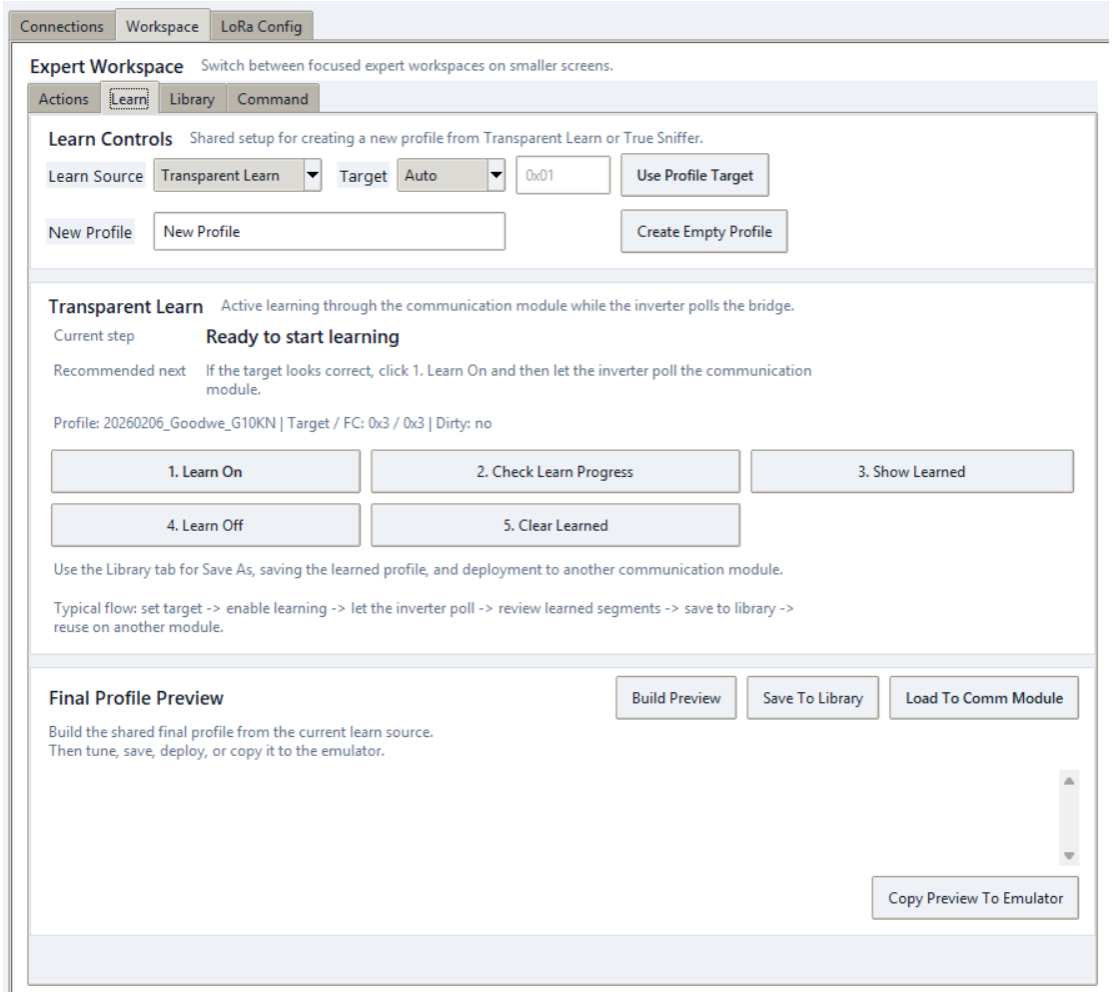


Actions

The Quick Actions section contains the most frequently used service buttons, for example:

- Status - displays the basic current state of the PV communicator, profile, learn mode and LoRa link
- Profile Info - prints information about the profile currently loaded in the communicator
- Segments - displays a detailed list of segments in the current profile including validity, request count and timeouts
- Reset Profile - restores the built-in base profile in the communicator and cancels current temporary changes in RAM
- Master On - turns on master polling, i.e. regular reading of data from the smart meter via LoRa
- Master Off - turns off master polling, i.e. stops regular smart-meter data refresh
- Refresh All - reads the current state from the communicator and refreshes the information displayed in the GUI

These buttons are used for quick checking of the PV communicator state and for routine service actions.



Learn

The Learn Controls section contains shared settings for both learning modes, Transparent Learn and True Sniffer Learn, for example:

- Learn Source - switches the learning source between Transparent Learn and True Sniffer Learn
- Target - determines whether the target Device ID is used automatically (Auto) or manually (Manual)
- Target value - in manual mode, used to enter the target Device ID, for example 0x01 or 0x03
- Apply Target / Use Profile Target - confirms the manual target or uses the automatic target according to the current status/profile
- New Profile - the name of the new working profile being created during the learn process
- Create Empty Profile - creates a new empty profile in communicator RAM or prepares a new profile context for the following workflow

Transparent Learn

The Transparent Learn section is used for active learning of segments directly through the PV communicator (that is, along the inverter -> PV communicator -> LoRa -> smart meter path) while the inverter polls the bridge, for example:

- Current step - displays the current state of the transparent learn process
- Recommended next - recommends the next suitable step according to the current situation
- Progress text - shows supplemental state information, for example the profile, target and learning progress
- 1. Learn On - turns on learn mode in the communicator
- 2. Check Learn Progress - reads the current communicator state and updates learn information in the GUI
- 3. Show Learned - prints the currently learned segments from the communicator
- 4. Learn Off - turns off learn mode in the communicator
- 5. Clear Learned - deletes the currently learned segments from communicator RAM

True Sniffer Learn

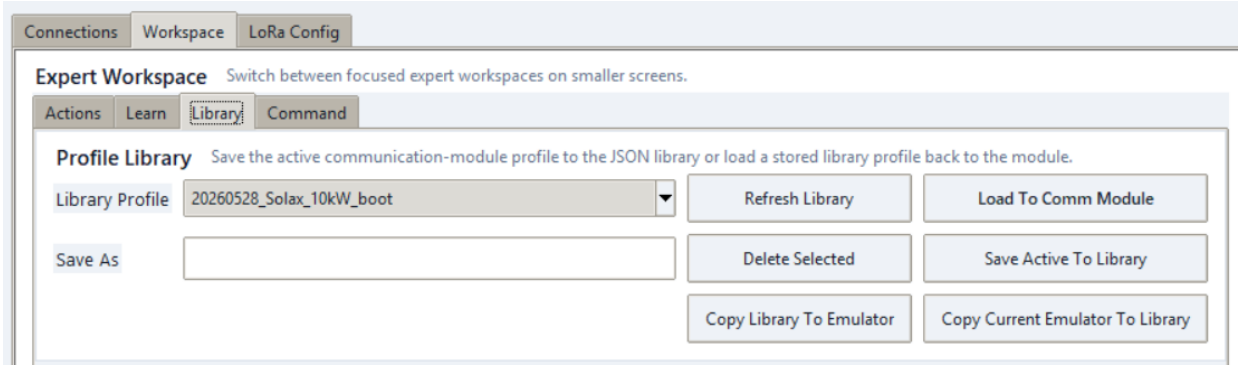
The True Sniffer Learn section is used for passive learning of request segments from an external metallic RS485 line through a USB/RS485 monitor, that is, outside the LoRa path, for example:

- Current step - displays the current state of the capture process
- Recommended next - recommends the next suitable step during sniffing
- Progress text - shows sniffer state, frame statistics and the target Device ID
- 1. Capture On - starts external sniffer capture
- 2. Check Capture Progress - refreshes statistics and ongoing capture state
- 3. Show Captured - prints captured request segments into Comm. Log
- 4. Capture Off - stops external sniffer capture
- 5. Clear Captured - deletes current captured data and resets capture statistics

Final Profile Preview

The Final Profile Preview section is used to assemble and finalize the resulting profile from both learn modes, for example:

- Build Preview - creates the final profile payload from the current learning source
- Preview window - displays the resulting profile as a profile_export line
- Save To Library - saves the final profile into the JSON profile library
- Load To Comm Module - loads the final profile into the PV communicator
- Copy Preview To Emulator - copies the final profile into emulator profiles



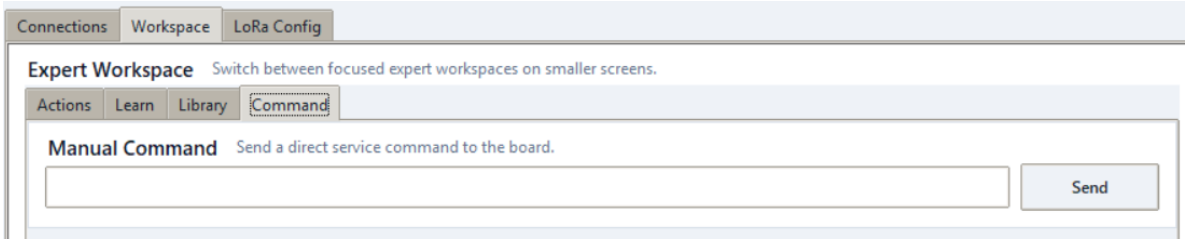
Library

The Library section is used for saving, loading and copying ready-made profiles between the library, the PV communicator and the emulator, for example:

- Library Profile - selection of a profile stored in the JSON library
- Refresh Library - reloads the profile list from the library
- Load To Comm Module - loads the selected library profile into the PV communicator
- Save As - the name under which the current profile will be saved to the library
- Delete Selected - deletes the selected profile from the library
- Save Active To Library - saves the currently active profile from the PV communicator into the library
- Copy Library To Emulator - copies the selected profile from the library into emulator profiles
- Copy Current Emulator To Library - saves the currently selected emulator profile back into the library

Short practical explanation:

- Library is the place where finished profiles are stored for later use
- a profile can be saved there after the learn process or loaded back from there into the communicator or emulator



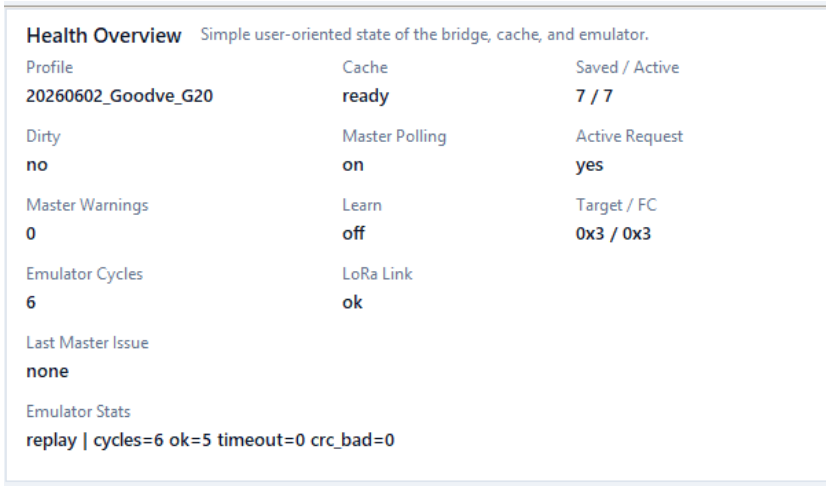
Command

The Command section is used for manually sending service commands directly to the PV communicator, for example:

- Command field - field for entering a service command in text form
- Send - sends the entered command to the communicator through the service port
- Service Log - the response to the entered command is displayed in the service log in the Connections tab

Short practical explanation:

- Command is intended mainly for service actions, diagnostics and advanced testing
- the detailed list of service commands and their meaning will be provided in a separate manual appendix



Health Overview

The Health Overview section shows a simplified current state of the PV communicator, active profile, data cache, learn mode, LoRa link and emulator. It is the main diagnostic window for quickly verifying whether the system is running correctly without having to read detailed service outputs.

Meaning of individual fields

- Profile - the name of the currently loaded profile in the PV communicator
  This field shows which profile is currently active. If the name does not match what the user expects, it is advisable to first verify that the correct profile was actually loaded into the communicator.
- Cache - a brief status of the data cache
  The cache is internal memory in which the PV communicator keeps the most recently read smart-meter data. Typical states may include ready, warming up, or another working state. If the cache is not ready, the inverter may not receive fresh data or the communicator may respond less optimally for a while.
- Saved / Active - the number of stored segments and the number of currently active segments
  The first number expresses the number of segments stored in the profile. The second number expresses the number of segments currently used by the communicator at runtime. In a normal healthy state, these values should make sense and usually match.
- Dirty - information about unsaved changes in RAM
  If this shows yes, it means the profile in RAM has been changed but has not yet been saved. If it shows no, the current runtime state corresponds to the saved version. This field is especially important after a learn process or after manual profile changes.
- Master Polling - the state of master polling
  It shows whether regular reading of data from the smart meter is enabled.
  - on means the communicator is actively requesting data from the smart meter
  - off means polling is disabled and the cache is not being refreshed

- Active Request - information about whether a request is currently running
  This field shows whether the communicator is in the middle of an active communication operation. It is especially useful for diagnostics when the user wants to see whether the system is moving or currently idle.
- Master Warnings - the number of recorded warnings or retry events
  This is the count of problematic situations during master communication, for example a retry after timeout. A low or zero value is normal. A higher number may indicate degraded communication, weaker LoRa transmission or overly aggressive polling.
- Learn - the state of learn mode
  It shows whether learn is off or active. The user can quickly see whether the communicator is currently in training mode or already running in normal operation.
- Target / FC - the current target and Function Code for the learn workflow
  This field shows which Device ID and MODBUS Function Code Learn is working with. It is especially important when several devices are present on one bus and the user needs to be sure that the correct device is being learned.
- Emulator Cycles - the number of completed emulator cycles
  This field shows how many polling or replay cycles the emulator has already completed. It is useful in lab testing when the user wants to verify that the emulator is actually running and generating repeated traffic.
- LoRa Link - a brief status of the LoRa connection
  It shows whether the link to the smart meter over LoRa appears healthy. If the state is not healthy, the problem is often outside the profile itself, for example in the radio path, LoRa module configuration or smart-meter availability.
- Last Master Issue - the last recorded problem during master communication
  This field is very useful for quick diagnostics. Instead of reading the whole log, the user immediately sees the last specific problem, for example a retry, timeout or another warning. It is one of the most important indicators when solving communication instability.
- Emulator Stats - brief emulator statistics
  It displays summary counts, for example:
  - number of cycles
  - number of successful responses
  - number of timeouts
  - number of CRC errors
    This field is important especially when testing profiles through the emulator.

What to look at first

During a quick routine check, it makes sense to watch mainly these fields:

- Profile - whether the correct profile is loaded
- Cache - whether the data is ready
- Master Polling - whether polling is enabled
- LoRa Link - whether the link is healthy
- Last Master Issue - whether retries or timeouts are appearing

If these fields are in order, the system is usually in a functional state.

What is considered a normal state

A healthy state is typically when:

- the correct Profile is loaded
- Cache is ready or is gradually filling after startup
- Master Polling is on
- LoRa Link is ok
- Master Warnings is 0 or only a low value
- Last Master Issue is none or does not change for a long time
- Dirty is no if the user is not currently working with a learn process

What may already be suspicious

Special attention is needed when:

- Profile does not correspond to the expected device
- Cache remains not ready for too long
- Master Polling is off even though normal operation is expected
- Master Warnings increases
- Last Master Issue repeatedly shows a retry or timeout
- LoRa Link does not show a healthy state
- Dirty remains yes even though the profile should already have been saved

Practical recommendation

The Health Overview section should be used as the first quick diagnostic filter.
If the user sees a problem here, the next step is usually:

- Status
- Segments
- Service Log
- or, when testing through the emulator, also checking Emulator Stats

Typical problem scenarios

1. **The wrong profile is loaded in the communicator**

Typical symptom:
- the Profile field does not match the device currently connected
- communication may work unstably or not at all

What it means:
- a profile for a different inverter or a different smart meter is loaded in the communicator
- the segments and polling logic then may not correspond to real operation

Recommended procedure:
- check the name shown in Profile
- if it does not match, load the correct profile from Library
- then verify Status, Segments and Health Overview again

2. **The cache is not filling or remains not ready**

Typical symptom:
- Cache does not reach a normal working state for a long time
- communication may be slow or incomplete

What it means:
- the communicator does not yet have fresh data prepared from the smart meter
- this may be a LoRa-path problem, polling being turned off, or a weak connection

Recommended procedure:
- check Master Polling
- check LoRa Link
- look at Last Master Issue
- or use Segments to verify whether the segments are valid and updating

3. **Master polling is turned off**

Typical symptom:
- Master Polling = off
- the cache is not being refreshed

What it means:
- the communicator is currently not reading data from the smart meter
- the profile may be loaded correctly, but without active polling the data will not be fresh

Recommended procedure:
- turn on Master On
- then verify that Cache and LoRa Link behave normally

4. **The LoRa link is not healthy**

Typical symptom:
- LoRa Link does not show a healthy state
- retries, timeouts or stale cache data may appear

What it means:
- the problem is probably not in the profile, but in communication between the PV communicator and the smart meter

Recommended procedure:
- check LoRa modules, power supply, addressing and radio connection
- look at Last Master Issue
- open a more detailed Service Log if needed

5. **Master warnings are increasing**

Typical symptom:
- Master Warnings gradually increases
- Last Master Issue shows retries or timeouts

What it means:
- communication works, but not perfectly cleanly
- this may indicate intermittent dropouts, weaker LoRa transmission or an overly demanding polling profile

Recommended procedure:
- check LoRa Link stability
- verify that the correct profile is in use
- during aggressive communication also watch Segments and, if needed, sched_debug

6. **Last Master Issue repeatedly shows timeout or retry**

Typical symptom:
- a similar problem keeps repeating in the Last Master Issue field
- for example a retry on a specific segment

What it means:
- the communicator is hitting a problem while obtaining or refreshing a specific data block
- this may indicate a radio-path problem, timing issue or profile suitability issue

Recommended procedure:
- open Segments and identify the problematic segment
- check whether the segment remains valid
- if the problem repeats, save logs and continue with deeper diagnostics

7. **Dirty = yes even after finishing work**

Typical symptom:

- Dirty remains yes even though the user expects a finished and saved state

What it means:

- the profile in RAM has been changed, but the changes have not yet been saved
- this is common after a learn process or after a profile import

Recommended procedure:

- if the changes are correct, save the profile
- if this is only a test state, the user may decide not to save it or to replace it with another profile

8. **Emulator Cycles increases, but the test does not produce the expected result**

Typical symptom:

- the emulator is running and the cycle counter is increasing
- but the communicator or the profile does not behave as expected

What it means:

- the emulator is generating communication, but the profile or test setup may not correspond to the real device
- or a different profile may be loaded in the communicator than in the emulator

Recommended procedure:

- check the profile name in Profile
- compare the profile in the communicator and the profile in the emulator
- check Emulator Stats, Service Log and Segments

9. **Learn is enabled, but no new segments are appearing**

Typical symptom:

- Learn is active
- but Show Learned or capture does not bring the expected new segments

What it means:

- either the inverter is not sending new unknown requests
- or the current profile already covers that communication
- in True Sniffer, this may also be caused by an incorrect target setting

Recommended procedure:

- check Target / FC
- verify that the correct device is being learned
- consider a cleaner profile baseline or a new start of the learn process

10. **Everything looks correct, but the user still sees a problem in the higher-level system**

Typical symptom:

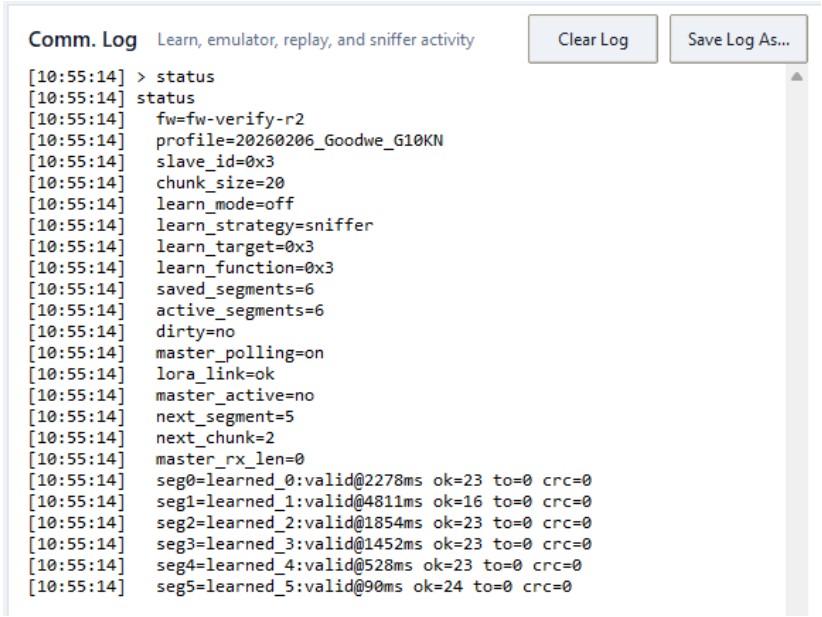
- Health Overview looks healthy
- but, for example, the inverter mobile app occasionally reports an abnormal state

What it means:

- the problem may be subtler and may not show up immediately in the basic overview
- it may involve borderline timing, aggressive polling or occasional response delay

Recommended procedure:

- look at Last Master Issue
- check Segments
- save a detailed log if needed
- and in more complex cases continue with deeper service diagnostics



Comm. Log

The Comm. Log section shows ongoing working messages generated mainly in the Learn and Library subtabs. It serves as a user-friendly operational log where it is possible to follow the learn workflow, sniffer capture, preview actions, profile-library work, and selected events related to the emulator or replay mode.
Unlike Service Log, this is not a raw service transcript but a practical log for normal GUI work.

What typical messages can appear here

- [sniffer] ... - messages from True Sniffer Learn, for example capture start, progress state, or the printout of captured segments
- [preview] ... - messages related to Final Profile Preview, for example saving a profile into the library or loading it into the communicator
- [library] ... - messages when saving and loading profiles from the library
- [sync] ... - messages when copying profiles between the library and the emulator
- live output of captured summary after using Show Captured
- information about which profile was saved, loaded or exported

Why Comm. Log is useful in practice

- it quickly shows what just happened after pressing a button in Workspace
- it helps verify that a profile was actually:
    - created
    - saved
    - loaded into the communicator
    - or copied to the emulator
- it allows the learn or preview workflow history to be copied without searching through raw service responses

How it relates to Health Overview

Comm. Log and Health Overview complement each other:

- Health Overview shows the current state of the system at one moment
- Comm. Log shows which steps and events led to that state

In practice:

- if Health Overview shows the correct profile, ready cache and healthy link state, Comm. Log helps trace when and how the profile was created or loaded
- if a suspicious state appears in Health Overview, Comm. Log helps verify whether it was preceded, for example, by:
    - starting the learn process
    - changing the profile
    - sniffer capture
    - copying to the emulator

What is considered normal logging

Typical and expected examples include:

- information about capture start or stop
- confirmation that a profile was saved to the library
- confirmation that a profile was loaded into the communicator
- confirmation that a profile was copied to the emulator
- printout of captured or learned segments

These messages are part of the normal workflow and do not indicate a problem.

What may already be suspicious

Special attention is needed when:

- the expected confirmation is missing from Comm. Log after pressing a button
- unsuccessful profile save or load messages keep appearing
- Show Captured or the learn workflow produces no segments
- the result in Comm. Log does not correspond to what the user expects in Health Overview

Example:

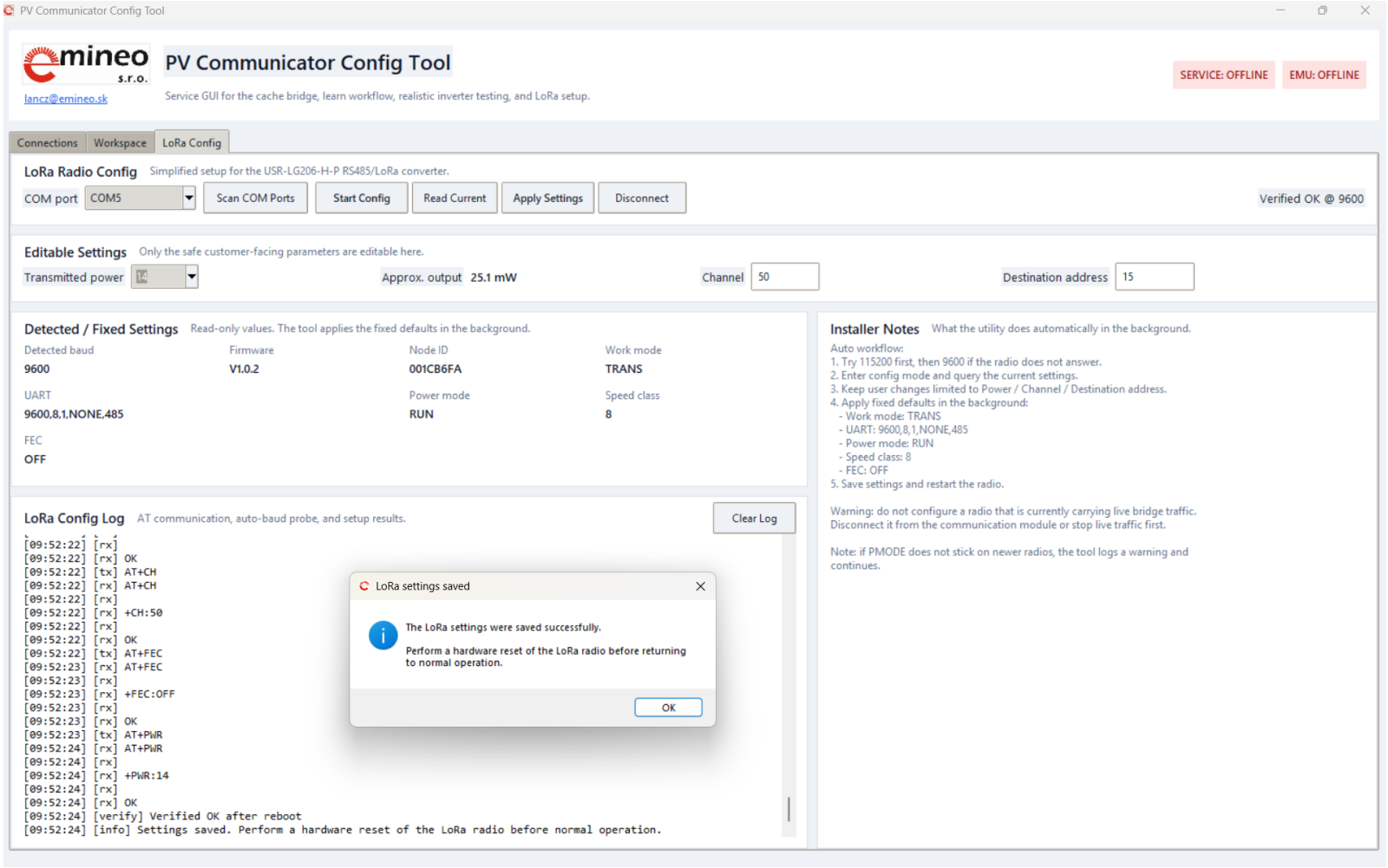
- Comm. Log reports that a profile was saved or loaded
- but Health Overview -> Profile still shows a different profile
  In that case it is advisable to verify whether the operation really completed correctly.

Relationship to Service Log

- Comm. Log is the working log for actions started from Workspace
- Service Log is the technical service log for low-level responses and diagnostics from Connections

Simple rule:

- if the user works in Learn, Library or with Final Profile Preview, they should mainly watch Comm. Log
- if they are troubleshooting communicator details, service commands or the emulator from Connections, they should mainly watch Service Log



LoRa Config Tab

The LoRa Config tab is used for service configuration of the LoRa converter.

It is used mainly when it is necessary to:

- select the correct COM port of the LoRa module
- read the current configuration
- adjust the basic parameters
- and safely write them back to the device

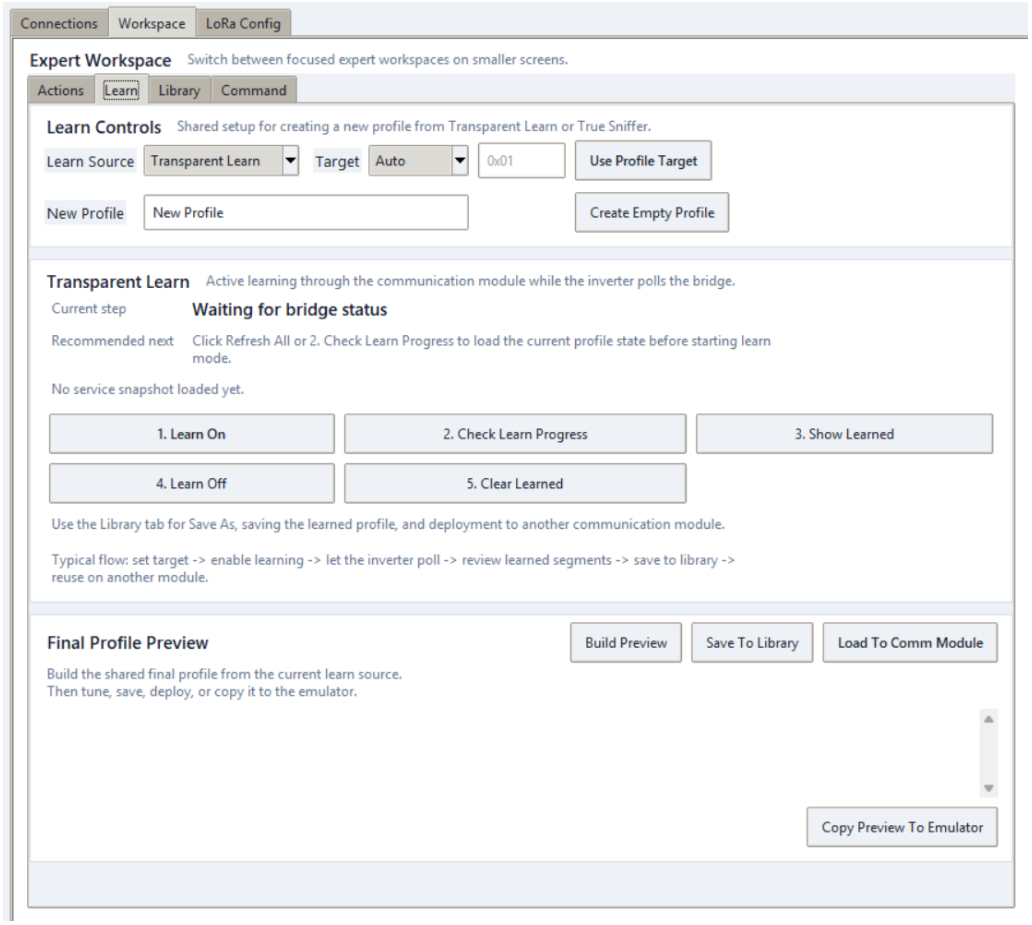
LoRa Radio Setup

1. In the LoRa Config subtab, select the correct COM port of the connected LoRa module and click Start Config.
2. Before changing the settings, disconnect the RS485 wires of the LoRa module from the PV communicator. During LoRa configuration, the radio shares its communication line with the service port, so a connected PV communicator could interfere with the setup process or distort the responses.
3. Click Read Current to load the current radio parameters.
4. In the Editable Settings section, set the user-facing parameters:
   • Transmitted power - transmitted power in dBm
   • Approx. output - approximate power conversion to mW
   • Channel - communication channel
   • Destination address - destination address of the peer device
5. Click Apply Settings. The GUI also automatically fills in and saves the fixed internal parameters required for correct operation.
6. After a successful save, check the Verified OK @ 9600 status and the log message confirming that the settings were verified after restart.
7. Finally, perform a hardware reset of the LoRa module so that the new settings are correctly taken over into normal operation.

Important note

• Do not configure LoRa during live communicator operation.
• Before configuration, disconnect the radio from the communication module or stop live communication.
• During configuration, the RS485 line of the LoRa module must be disconnected from the PV communicator.

Creating a New Profile



Transparent Learn
Transparent Learn mode is used to create a new profile directly through the PV communicator during real communication between the inverter and the smart meter. In this mode, the communicator watches the requests coming from the inverter and gradually creates new profile segments from them.

This mode is especially suitable when:

- the user does not have an external RS485 sniffer
- or wants to create the profile directly during normal operation along the inverter -> PV communicator -> LoRa -> smart meter path

What to prepare before starting

Before starting Transparent Learn, it is advisable to:

- connect the PV communicator service port to the computer
- verify correct communication with the smart meter
- have a name ready for the new profile
- know which Device ID should be learned
- check Target in Learn Controls

If the user wants full confidence, they can read before the learn process:

- Status
- Profile Info
- Segments

What to do step by step

1. In Learn Controls, select:
    - Learn Source = Transparent Learn
2. Set Target
    - Manual if the user knows the specific Device ID
    - Auto if they want to use the current target according to the profile or communicator
3. Enter New Profile
    - that is, the name of the new profile that will be created
4. Click Create Empty Profile
    - this prepares a new empty profile base
5. Click 1. Learn On
    - this enables learn mode in the communicator
6. Let the inverter communicate normally
    - during operation, the communicator will capture requests and create segments from them
7. Use 2. Check Learn Progress during the process
    - this refreshes the learn-process state and loads current information into the GUI
8. Use 3. Show Learned as needed
    - this prints the currently learned segments
9. When the collection is finished, click 4. Learn Off
    - this disables learn mode
10. If the user wants to start completely from scratch, they can use 5. Clear Learned
    - this deletes the currently learned segments from RAM

11. After the learn process is finished, use Final Profile Preview
    - Build Preview
    - check the result
    - then, as needed:
        - Save To Library
        - Load To Comm Module
        - Copy Preview To Emulator

What is expected during correct operation

A normal and healthy course is when:

- Current step and Recommended next make logical sense
- Check Learn Progress shows that the communicator is working in learn mode
- Show Learned gradually displays new segments
- Target / FC in Health Overview is correct
- Comm. Log and Service Log do not show repeated errors
- after some time, a usable result appears in Final Profile Preview

What is okay in this mode

It is considered normal when:

- learned segments appear gradually, not all at once
- the communicator needs several seconds or more to capture a stable image of operation
- sometimes the learned profile does not change immediately after every single request
- after startup it may take a while before operation settles

If the current profile already covers a large part of the requests, learn may seem slower or may capture only a few new segments.

What may already be suspicious

Special attention is needed when:

- Learn On completes, but Show Learned does not show anything new for a long time
- Target / FC does not match the expected device
- Comm. Log or Service Log shows problematic messages
- the communicator is running, but segments do not appear
- Dirty changes, but the resulting profile makes no sense

This may mean, for example:

- an incorrectly selected target
- the profile already covers the given communication
- the inverter is not sending new requests
- or the communication at that moment is not representative

How to evaluate the result

After the learn process is finished, it is good to check:

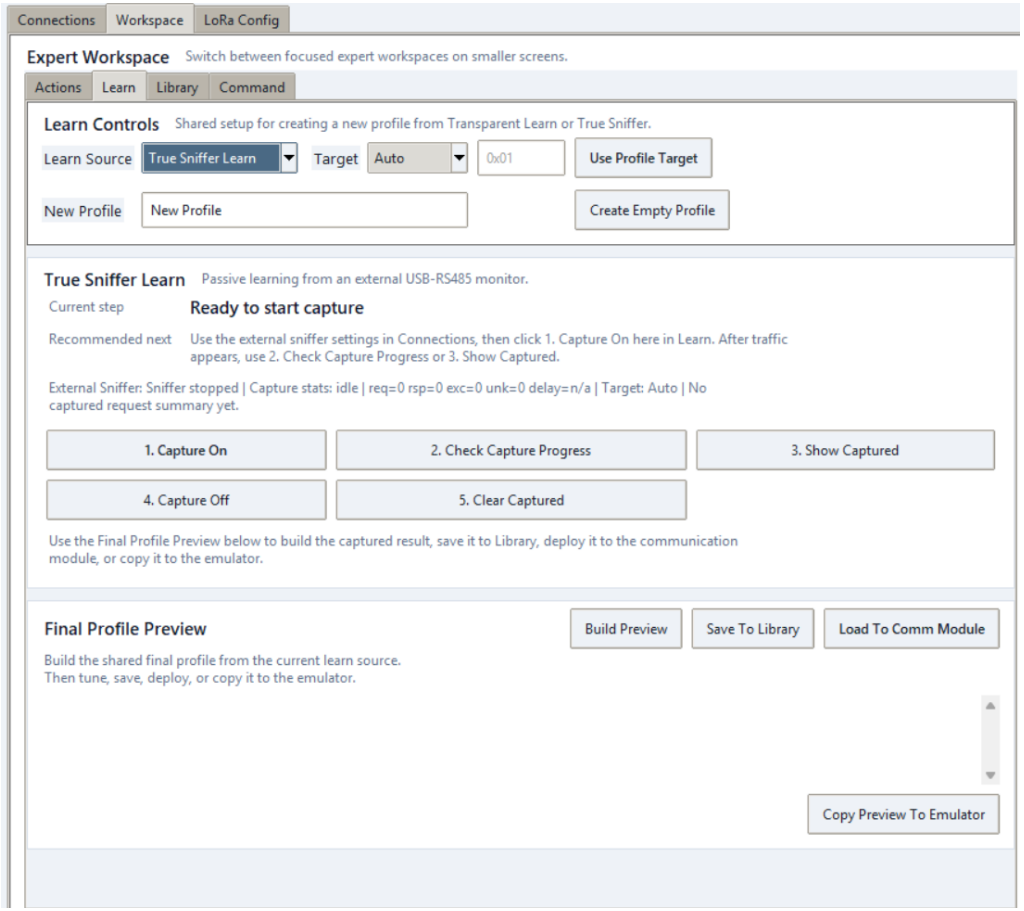
- whether the learned segments correspond to real communication
- whether the number of segments makes sense
- whether the resulting profile_export in Final Profile Preview looks logical
- whether the profile name matches the device used

If the user trusts the result, they can:

- save the profile to the library
- load it into the communicator
- or copy it to the emulator

Practical recommendation

Transparent Learn is very suitable for creating a functional profile directly in operation.
However, if the user needs the most faithful possible picture of metallic RS485 communication, especially for detailed tuning or when there is suspicion of unusual line behavior, it is advisable to supplement or compare the result through True Sniffer Learn as well.



True Sniffer Learn

True Sniffer Learn mode is used to create a new profile from real communication on the metallic RS485 line by means of an external USB/RS485 monitor. Unlike Transparent Learn, communication does not go through the PV communicator here; instead, traffic between devices on the RS485 bus is monitored passively and directly.

This mode is especially suitable when:

- the user wants to capture real requests directly from the metallic line
- they want to compare or supplement the learn process independently of the LoRa path
- they need a more faithful picture of real inverter operation

What to prepare before starting

Before starting True Sniffer Learn, it is advisable to:

- connect the external USB/RS485 monitor to the metallic RS485 line
- correctly set the External Sniffer Port section in the Connections tab
- check:
    - COM port
    - Baud
    - Gap [ms]
- prepare the name of the new profile
- check Target in Learn Controls

Important:

- this section only prepares the capture workflow
- the sniffer technical setup itself is done in Connections
- the actual capture start is done only in Workspace -> Learn

What to do step by step

1. In Learn Controls, select:
    - Learn Source = True Sniffer Learn
2. Set Target
    - Manual if the user knows the specific Device ID
    - Auto if they want automatic target resolution according to status or profile
3. Enter New Profile
    - that is, the name of the new profile that will be created
4. Click Create Empty Profile
    - this prepares a new working profile base
5. Check in Connections -> External Sniffer Port that the following are correct:
    - COM port
    - Baud
    - Gap [ms]
6. Click 1. Capture On
    - this starts external sniffer capture
7. Let real communication run on the RS485 line
    - as needed during inverter startup or during normal operation
8. Use 2. Check Capture Progress during the process
    - this refreshes the ongoing capture state and capture statistics
9. Use 3. Show Captured as needed

    - this prints the captured request segments into Comm. Log

10. When enough data has been captured, click 4. Capture Off
    - this stops capture
11. If the user wants to start over, they can use 5. Clear Captured
    - this deletes the current capture data and statistics
12. After capture is finished, use Final Profile Preview
    - Build Preview
    - check the resulting profile
    - then, as needed:
        - Save To Library
        - Load To Comm Module
        - Copy Preview To Emulator

What is expected during correct operation

A normal and healthy course is when:

- Current step and Recommended next correspond to the current capture phase
- Capture stats show that frames are increasing
- Show Captured prints meaningful request segments into Comm. Log
- the captured segments correspond to real communication on the bus
- the resulting profile in Final Profile Preview makes logical sense

What is okay in this mode

It is considered normal when:

- request segments appear gradually
- some less frequent segments appear only after a longer time or only during device startup
- capturing the full profile sometimes requires longer observation or an inverter restart
- ongoing statistics change according to current activity on the line

If the user observes only a short segment of operation, it is normal that the resulting profile may be incomplete.

What may already be suspicious

Special attention is needed when:

- capture is running, but no request segments are appearing
- Show Captured does not print anything usable

- Target does not match the correct device on the bus
- the log shows merged or broken frames
- the resulting profile is suspiciously empty or incomplete

This may mean, for example:

- incorrectly selected COM port
- incorrect Baud
- an unsuitable Gap [ms] setting
- incorrect Target
- or a non-representative communication segment

How to evaluate the result

After capture is finished, it is good to check:

- whether the captured requests make sense
- whether the number of segments matches the expected operation
- whether Build Preview creates a logical profile_export
- whether the profile name corresponds to the device from which it was created

If the user trusts the result, they can:

- save the profile to the library
- load it into the communicator
- or copy it to the emulator

Practical recommendation

True Sniffer Learn is especially suitable when the user needs to create the profile from the most faithful possible view of metallic communication.
It is a very suitable option when verifying a new device, checking a real line, or when the user wants to compare the result with Transparent Learn.

Recommended practical workflow for creating a new profile

(this section is a summary of the previous explanations)

Option A – Transparent Learn

Use this path when you want to create a profile directly through the PV communicator during normal operation (with direct inverter-to-smart-meter communication over LoRa).

1. Connect the PV communicator service port to the computer.
2. Verify that the communicator is communicating with the smart meter and that Master Polling is working.
3. In Workspace -> Learn, set:
    - Learn Source = Transparent Learn
    - the correct Target
    - a name in the New Profile field
4. Click Create Empty Profile.
5. Click 1. Learn On.
6. Let the inverter communicate normally.
7. After several seconds or minutes, click:
    - 2. Check Learn Progress
    - 3. Show Learned
8. If the result looks good, click:
    - 4. Learn Off
9. In Final Profile Preview, click:
    - Build Preview
10. If the profile looks correct, continue with:
    - Save To Library
    - Load To Comm Module
    - or Copy Preview To Emulator

Option B – True Sniffer Learn

Use this path when you want to create a profile from real metallic RS485 communication using an external sniffer.

1. Connect the external USB/RS485 monitor to the metallic RS485 line.
2. In Connections -> External Sniffer Port, set:
    - the correct COM port
    - the correct Baud
    - a suitable Gap [ms]
3. In Workspace -> Learn, set:
    - Learn Source = True Sniffer Learn
    - the correct Target

    - a name in the New Profile field
4. Click Create Empty Profile.
5. Click 1. Capture On.
6. Let real communication run, or restart the device if needed.
7. During the process, click:
    - 2. Check Capture Progress
    - 3. Show Captured
8. When the result is sufficient, click:
    - 4. Capture Off
9. In Final Profile Preview, click:
    - Build Preview
10. If the profile looks correct, continue with:
    - Save To Library
    - Load To Comm Module
    - or Copy Preview To Emulator

How to choose the correct path

- Transparent Learn is suitable for quick creation of a functional profile directly in operation through the PV communicator and the LoRa path.
- True Sniffer Learn is suitable when you want to create the profile from real metallic RS485 communication and have the most faithful possible picture of the line.

What to check after creating the profile

After creating the profile, it is advisable to verify:

- whether the profile name is correct
- whether Build Preview created a logical profile_export
- whether the profile was saved correctly into Library
- whether Profile in Health Overview matches after loading the profile into the communicator
- whether the system reports warnings, retries or timeouts

If something does not work

If the result does not look correct:

- check Target
- check Comm. Log
- check Service Log
- check Health Overview
- and then return to the more detailed description of the relevant section of the manual

How to quickly assess whether the resulting profile makes sense (applies to both segment-learning modes for creating a new profile)

After clicking Build Preview, the final profile is shown as a single profile_export line. This line should be checked at least briefly before saving it to Library or loading it into the PV communicator.

Example of a resulting profile:

0x2012,8,sniff_2012_8,3,3 | 0x101E,2,sniff_101E_2,1,1 | 0x1028,2,sniff_1028_2,1,1 | 0x0B,1,sniff_000B_1,1,1

How to read one profile line

Example:
0x2012,8,sniff_2012_8,3,3

Meaning of the individual fields:

- 0x2012 = start register
- 8 = number of registers
- sniff_2012_8 = internal segment name
- the first number 3 = segment priority
- the second number 3 = refresh class

Simply put:

- 3,3 means: this is the main block, keep it fresh and refresh it often
- 1,1 means: this is a helper or occasional block, refreshing it less often is sufficient

Practical meaning:

- if you know that a given block is the main live block of the inverter, usually keep it at 3,3
- if it is a supplementary or status block, 1,1 is usually sufficient

How to recognize that the traffic was not representative

The result may be incomplete or less accurate, for example when:

- only 1 or 2 blocks were learned even though you know the meter normally uses more segments
- the learn process covered only a short period of operation
- the communication did not include boot requests or steady normal operation
- in Transparent Learn, the profile was learned only from slow manual polling from an external program, not from a real inverter

• only 1 or 2 blocks were learned even though you know the meter normally uses more segments
• the learn process covered only a short period of operation
• the communication did not include boot requests or steady normal operation
• in Transparent Learn, the profile was learned only from slow manual polling from an external program, not from a real inverter
In such a case it is advisable to:
• do not save the profile blindly yet
• review the result in Final Profile Preview
• and manually tune it if necessary

Example:

• if True Sniffer Learn for Solax produced the main block as 3,3 and the helper blocks as 1,1
• you can manually rewrite the same values into the preview from Transparent Learn if you know it is the same smart meter

Final recommendation

If you have doubts about whether the profile is good:
• first save it to Library
• then copy it to the emulator
• and verify it safely in the emulator before making a live change on the real setup