Getting started with the STSW-STNRG012GUI GUI for STNRG012 products

Introduction

This document describes the STSW-STNRG012GUI operations to control the STNRG012 product using the STEVAL-PCC020V2 USB to I²C/UART interface board. If you are using the STEVAL-PCC020V1, buy this new version.

The STEVAL-PCC020V2 is a dedicated USB hardware interface used to connect a Windows®-based PC with the STNRG digital power supply controllers, such as the STNRG01x.

This document explains first the software installation and then the detailed use of the interface with the STSW-STNRG012GUI.

This GUI allows monitoring the digital controller status in real-time. It also allows tuning specific parameters according to the customer's needs.

1 GUI features

- Real-time monitoring of the digital controller status
- Access to STNRG012 NVM parameters
- Access to STNRG012 external E²PROM for patch upload, calibration, and event history
- Embedded PFC calibration wizard

2 Software installation

You have to install the USB driver and the PC GUI before using the STEVAL-PCC020V2 interface board.

2.1 Virtual COM port driver installation (SiLabs CP2102)

To use the STEVAL-PCC020V2 interface board, install one of the USB drivers located in the Driver\CP210x_VCP_Windows CD folder:

- “CP210xCVCPInstaller_x86.exe” (for 32-bit OS)
- “CP210xCVCPInstaller_x64.exe” (for 64-bit OS)

Alternatively, you can find the latest version of the drivers on the SiLabs website.

When the interface board is plugged onto the PC, the driver is automatically installed.

2.2 GUI installation

To install the STSW-STNRG012GUI, follow the procedure below.

Step 1. Launch *Setup.msi* located in the download folder.

Step 2. Follow the installation wizard instructions.

By default, the GUI is installed under C:\Program Files (x86)\STMicroelectronics\STNRG012 GUI\. The GUI installer creates an icon in the [**Start**] menu, under STMicroelectronics\STNRG012.

Figure 1. STSW-STNRG012GUI icon

*Note: If a previous version of the software has already been installed, uninstall it through the [**Windows Control Panel Uninstall**] option.*

Note: Install and save the STSW-STNRG012GUI in a folder with administrator rights for reading and writing.

3 GUI overview

The STSW-STNRG012GUI is designed for debugging power supply applications.

It allows:

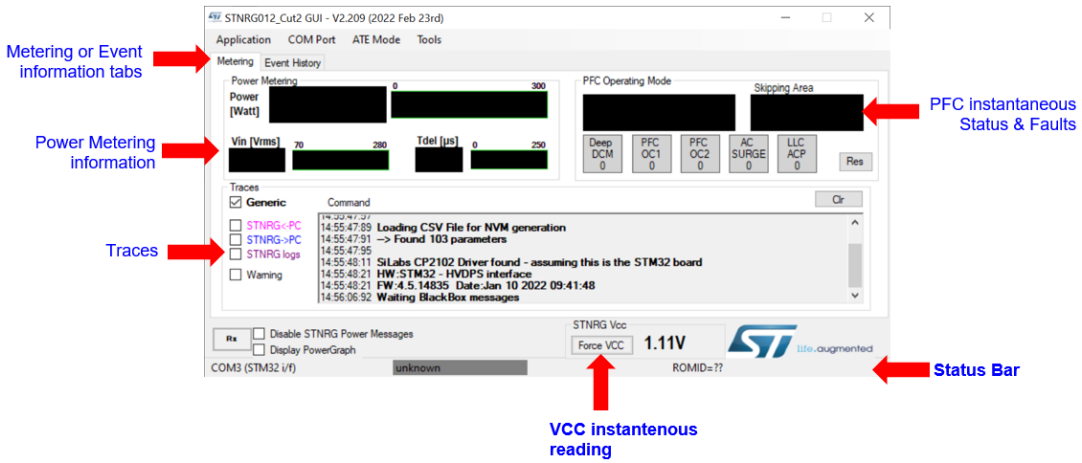
- reading instantaneous power metering information and PFC operating modes;
- reading and modifying STNRG012 NVM parameters defining the supply behavior (gain, fault management, delays, PFC/LLC parameters, etc.);
- reading event history data (fault history, stored in optional E²P);
- programming optional E²P patches;
- accessing internal firmware variables (patch needed).

3.1 Startup screen

The GUI is split in the following areas:

- **Menu Bar**: used to select the operation mode, that is, to communicate with STNRG012, display logs, access E2P directly, NVM programming, etc.
- **Metering or Event tabs**: displays the power metering information or event history
- **Traces and Status**: internal debug traces and status bar that show the STNRG012 current status

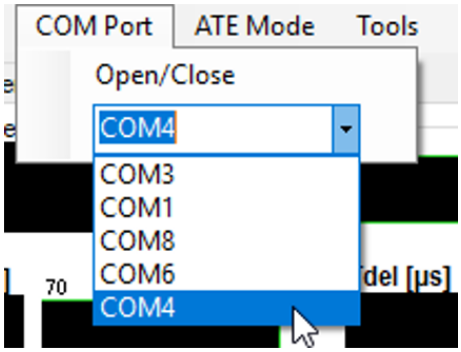
Figure 2. STSW-STNRG012GUI startup screen



3.2 Connection management

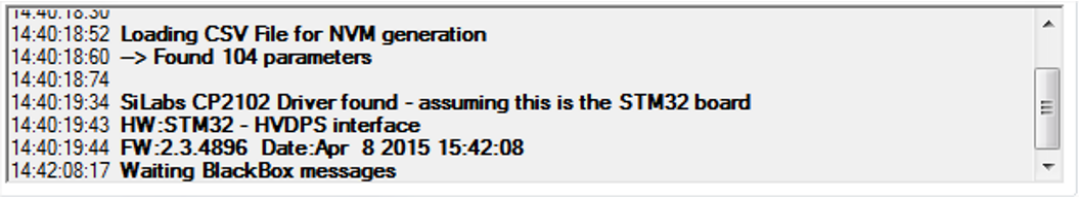
At startup, the GUI automatically detects the COM port to be used (the GUI selects the CP2102-based VCP). In the case of multiple CP2102, you have to select manually the right COM port via the COM port menu. You can also open or close the COM port via the menu shown below.

Figure 3. COM port selection



Once the right COM port is selected, the GUI tries to communicate with the interface board microcontroller, as shown below.

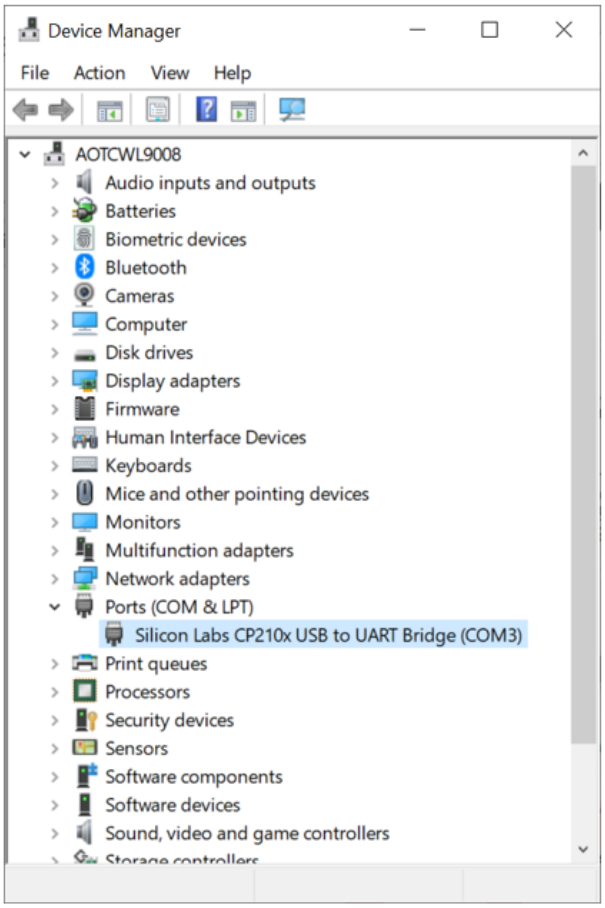
Figure 4. Traces during GUI connection



Once the microcontroller has been detected, the GUI displays the associated hardware and firmware version, and the build date.

*Note: If the GUI does not find a SiLabs-based VCP, an error message appears. Check if the SiLabs VCP is correctly recognized by pressing plus [**Pause**]. Then, select [**Device Manager**]>[**Silicon Labs CP210x USB to UART Bridge (COM3)**], as shown in the following figure.*

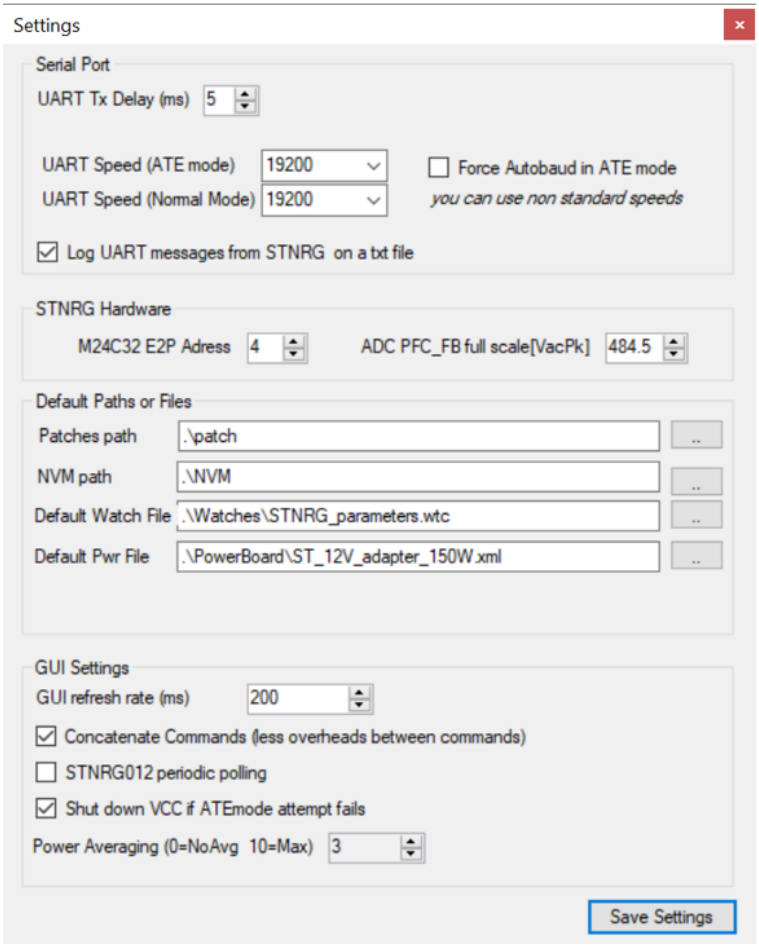
Figure 5. SiLabs VCP in the [Device Manager]



3.3 Settings

You can access the GUI settings through the [**Application**]>[**Settings**] menu.

Figure 6. [Settings] menu



Some settings (for example, the GUI refresh rate or power averaging) can be changed in real-time. Press the [**Save Settings**] button to save the settings into the *config.xml* file.

Table 1. GUI setting parameters

Serial port	
UART Tx delay (ms)	Optional Tx delay. Keep it at 0 ms.
UART speed (ATE mode)	STNRG012 UART speed during the logging phase
UART speed (normal mode)	STNRG012 UART speed during the ATE mode
Forces autobaud in ATE mode	When entering the ATE mode, it performs the autobaud algorithm. This is supposed not to be necessary, as the STNRG012 internal RC oscillator is trimmed in production to guarantee the correct UART speed
Log UART messages from STNRG012	Option to log the UART exchange on a file (uart_trace.txt on the GUI executable directory)
STNRG012	
M24C32 E2P address	Hardware address of the external E2P. STNRG012 is always assuming 4 (100)
ADC PFC_FB full scale	Full-scale equivalent value of the PFC_FB pin, which is the voltage expected at the bulk capacitor when the voltage at the PFC_FB pin is at the ADC full scale (2.5 V). Keep this value, if you are using the standard resistive bridge divider (9 MΩ/46.7 kΩ)
Default paths	
Patches path	Default path for the E2P patches
NVM path	Default path for the NVM settings
Default watch files	Default path for the watch settings, used to monitor the firmware internal variables
Default power file	Calibration file for the power metering
GUI settings	
GUI refresh rate	Delay in ms among each GUI refresh
Concatenate commands	Messages sent to the STNRG are concatenated to avoid USB overhead (`Write1-Write2...Read1-Read2` instead of `Write1-Read1-Write2-Read2…`)
STNRG012 periodic polling	Periodically polls the STNRG012 status
Shut down the VCC if the ATE mode fails	If enabled, when the device cannot enter the ATE mode, it switches off the VCC supply
Power averaging	Averaging filter for real-time power display: • 0 = no averaging • 10 = maximum averaging

1. *Used only when the [**Power Monitor**] window is active.*

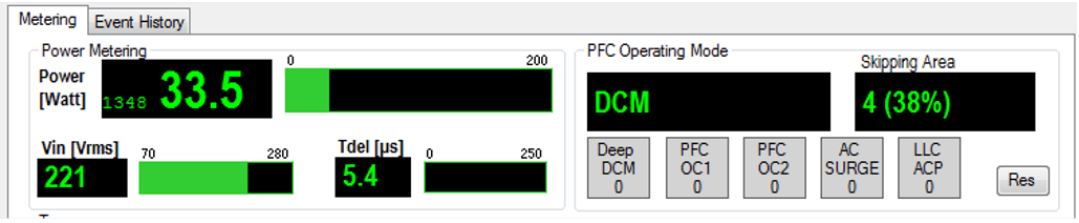
4 GUI normal mode

4.1 Power metering

During the normal mode, the STNRG012 sends the information used to compute the actual power delivered by the PFC, that is:

- the estimated power computed by the power integration algorithm
- some factors used for power estimation correction:
  - V_{IN} (mains) voltage
  - PFC mode of operations (DCM, valley skipping, TM)
  - Time between PFC pulses (DCM mode only)
  - Phase angle modulation ratio (at low power only)
- Some flags about temporary PFC faults (deep DCM, PFC_OCP1, etc.).

Figure 7. GUI metering information panel



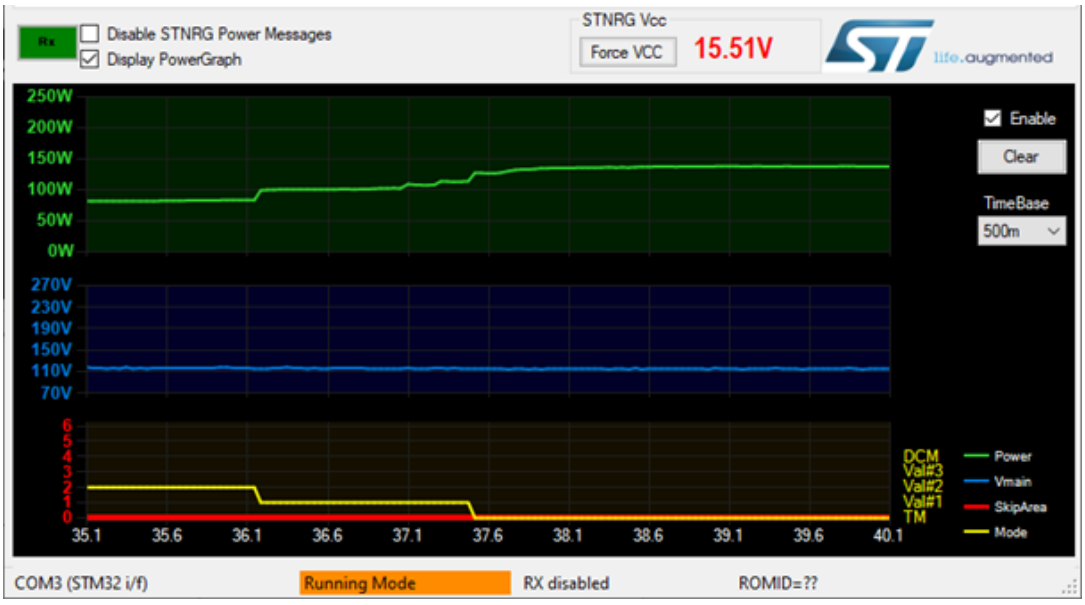
Note: If the STNRG012 power messages are disabled, the boxes shown above are empty.

Important: The power metering is not available in the burst mode, to save the MCU power energy, hence, the efficiency at low power).

4.2 Power graph report

By checking the [**Display PowerGraph**] box, the history of the [**PowerGraph**] event is displayed. It shows the long-term stability of the power supply or the mode changes compared to the load.

Figure 8. GUI [PowerGraph] report



4.3 Event history and factory data prerequisite: E²P

Event history and factory data are available only if an external E²PROM (M24C32) is connected to the STNRG012. This allows retrieving some information in case of system failures.

Since the fault history and factory data are sent only at the system power-up, you have to turn the power supply off and on to get the status.

However, if the UART uplink communication is enabled (a patch is needed), it is possible to send a request to the STNRG012 to show immediately the event and fault history by pressing the [**Get Factory Data**] button.

Another option is to read directly the E²P content.

4.4 Event history

The fault history is stored in the optional external E²P.

The faults are stored in an eight-position circular buffer; hence, only the last eight faults are stored.

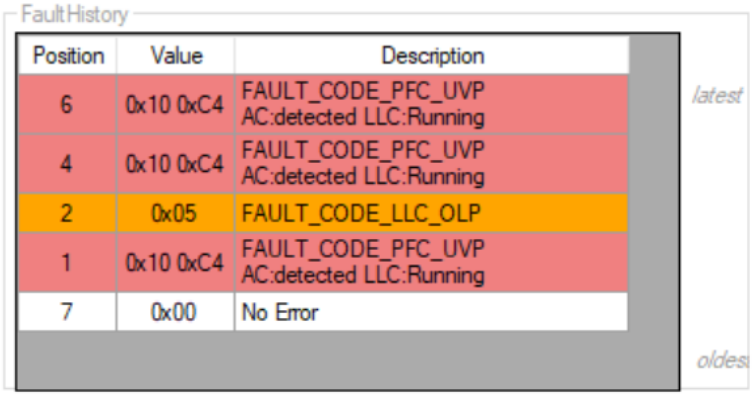
At each power-up, the STNRG012 sends the content of the fault history, which is a sort of black box, to the host.

The GUI displays the faults in a chronological order (the latest at the bottom).

Note: There are two types of faults:

- *the standard fault → one position per fault in the circular buffer;*
- *the fault with debug information → two positions per fault in the circular buffer; this fault provides more firmware information.*

Figure 9. GUI fault history



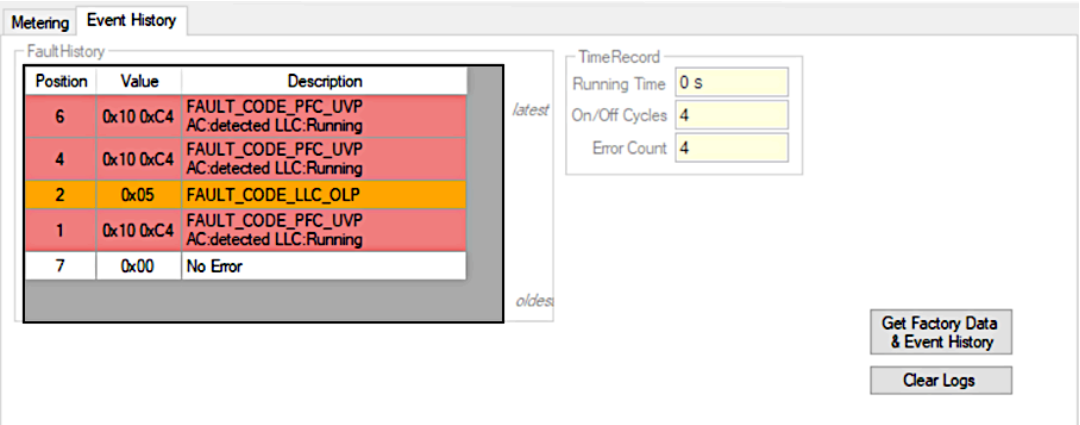
The number of faults stored depends on the fault type (between 4 and 8).

For instance, the figure above shows:

- a fault **PFC_UVP with AC presence** (position 6), which uses two positions in the buffer
- a fault **PFC_UVP with AC presence** (position 4), which uses two positions in the buffer
- a fault **LLC_OLP** (position 2), which uses only a position in the buffer
- a fault **PFC_UVP with AC presence** (position 1), which uses two positions in the buffer
- a **No Error** in position 7 of the buffer

4.5 Time record display

Figure 10. **GUI event history panel**



Like the fault history, the event history data are sent at each STNRG012 power-up.
The table below shows the time record parameters.

Table 2. **GUI E²P time record parameters**

Time record	
Running time	Power supply cumulated active time
On/off cycles	Number of power supply restart events
Error count	Number of errors

5 Power metering calibration

5.1 Background

The STNRG012 provides continuous power-metering information to the host about:

- the PFC integrated power in raw format, which is the result of the PID integrator;
- the input voltage;
- the PFC operating mode and skipping area;
- the PFC fault status.

The instantaneous raw power estimation can be computed as follows:

$$P_{Raw}[W] = PFC_{lsb} \times \left[128 \times \left(\frac{FSR_{Vin}}{256}\right) \times \frac{t_{smed}}{L}\right] \quad (1)$$

where:

- P_{Raw} is the PFC power (not corrected);
- PFC_{lsb} is the PFC integrated power in raw format;
- FSR_{Vin} is the full-scale ADC voltage reading = 480 V;
- t_{smed} is the *smed* event (PFC timer) minimal duration = 1/60 MHz = 16.67 ns;
- L is the PFC inductor value (typically 250 µH on the evaluation board).

The raw power is almost proportional to the actual PFC power.

On the basis of the PFC mode and input voltage, it is possible to correct the raw power to deduce the actual power consumption.

As several parameters have to consider, a very simple approach is to use a calibration method for each PFC operating mode.

The output power is also compensated with the skipping area and input voltage, using the formula below:

$$Pout[W] = (P_{Raw} \times C_{Mode} + C_{Vin}) \times C_{PAM} \times (1 + C_{Tdel}) \quad (2)$$

where:

- P_{Raw} is the raw power;
- C_{Mode} is the PFC mode correction factor (typically between 0.7 and 1);
- C_{Vin} is the input voltage correction using a second order polynomial (a+b*Vin+c*Vin²). However, since we typically consider only two voltages (EU/US), the second order term is set to 0;
- C_{PAM} is the skipping area correction factor (1 for no PAM, 0.26 for the minimum PAM);
- C_{Tdel} is the DCM mode correction using a first order equation (a+b*Tdel).

5.2 Metering calibration

To display the power metering correction factors described in the previous section, go to the [**Application → Power Metering Calibration**] menu. The GUI highlights the current PFC operating mode in yellow.

To calibrate the PFC mode parameters, follow the procedure below.

Step 1. Set a load to make the system enter the transition mode (near nominal power, that depends on the board).

Step 2. Enter the associated reference power read on a precision power meter in the [**Ref Pwr**] box.

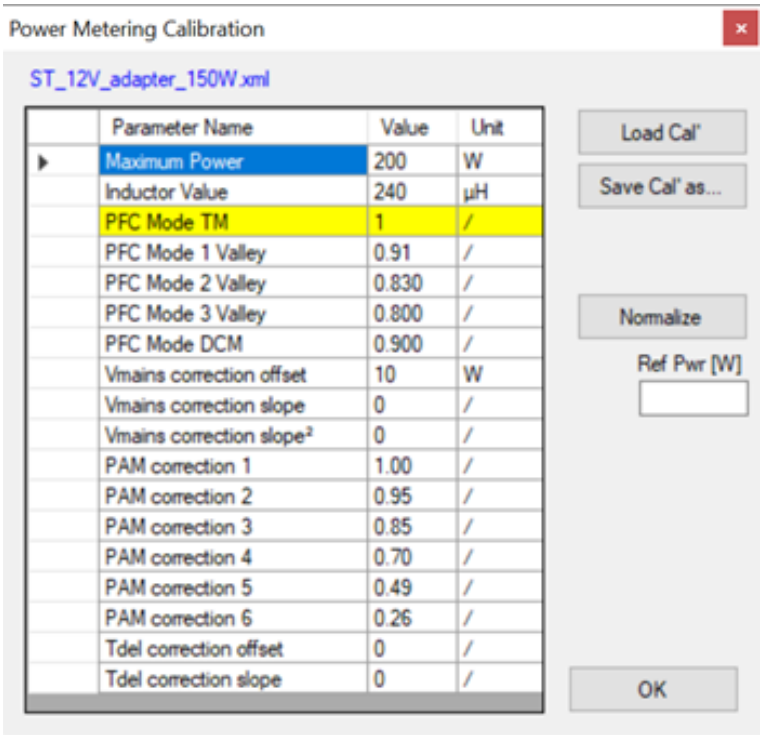
Step 3. Press the [**Normalize**] button.

The GUI automatically computes and updates the correction factor associated to the current PFC operating mode.

The power displayed in the GUI must be equal to the reference power. This operation has to be repeated for every PFC mode (DCM, *Valley#1/2/3*).

The other parameters (PAM correction, Tdel correction, Vmains) have to be tuned manually.

Figure 11. GUI power metering calibration window



6 ATE mode

6.1 Normal and ATE mode: differences

The STNRG012 supports two main modes of operation:

- **Normal (or running)**: is the GUI normal operation mode
- **ATE**: is mainly used to program the STNRG012 NVM section and is based on the ATE protocol

Table 3. Differences between normal and ATE modes

Mode	Running (normal) mode	ATE mode
Metering information	Available through the metering protocol (STNRG012 to host)	Not available
ATE protocol	No (default)	Yes
Communication	STNRG012 →host only (default)	Bidirectional
PFC/LLC operations	Yes	No
STNRG012 supply	Self-supplied	External supply
NVM write	No	Yes

Note: It is also possible to enable the ATE protocol in running mode for the bidirectional communication. This requires a specific patch to enable the UART uplink communication.

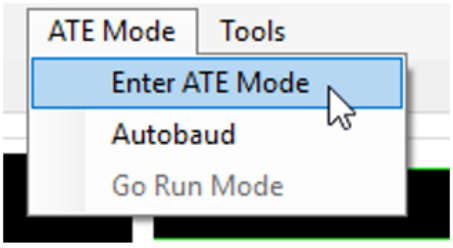
*Note: If the ATE protocol is enabled during the normal mode, the GUI is able to handle simultaneously **[Metering information]** and **[ATE protocol]**. However, the bandwidth on the UART link might be impacted. So, it is recommended to disable the metering protocol by checking the **[Disable STNRG Power messages]** box.*

Figure 12. GUI: disable STNRG012 power messages

6.2 Entering ATE mode

The ATE mode menu is used to set the STNRG012 in ATE mode.

Figure 13. ATE mode menu



The available options are:

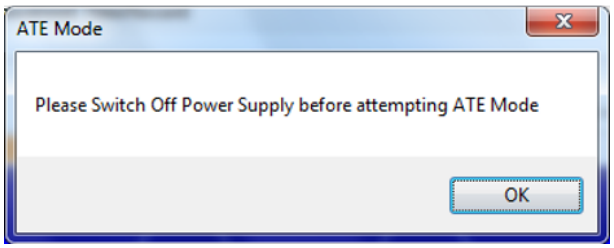
- [**Enter ATE Mode (Force VCC)**]: attempts the ATE mode procedure.
  The GUI:
  1. asserts the SCL pin low (requests to enter the ATE mode);
  2. switches on the internal VCC generation;
  3. checks that the ATE mode is effectively working;
- [**Auto baud**]: is used for internal debug purposes but also to check the UART speed. Basically, it performs the autobaud and computes the optimal UART link speed;
- [**Go Run Mode**]: switches off the internal VCC generation.

Once the device has successfully entered the ATE mode, the status bar is updated and the ROM ID is displayed (here is 0x00C2).

Figure 14. ATE mode status bar

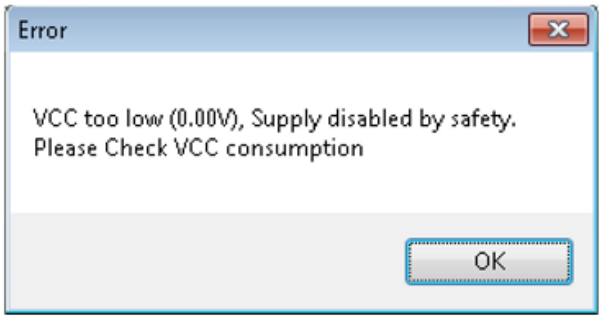
Note: The GUI prevents accessing the ATE mode or forcing VCC when it detects that the STNRG012 is running in the normal mode. The main power supply must be switched off before entering the ATE mode.

Figure 15. GUI error when trying to enter ATE mode while in normal mode



Note: Once the VCC is applied (just after the soft-start), the GUI measures the VCC voltage value: if it is below a given threshold (17 V), the GUI assumes an overconsumption and the VCC is automatically disabled.

Figure 16. GUI error when the VCC is applied



6.3 Autobaud feature

If the device is not able to enter the ATE mode, it might be due to UART communication issues.

The STNRG012 timing relies on an internal calibrated oscillator, but it does not have the accuracy of an external crystal.

After the ST factory calibration (trimming), the resulting accuracy is about 2.3%. It is enough to ensure the normal UART communication.

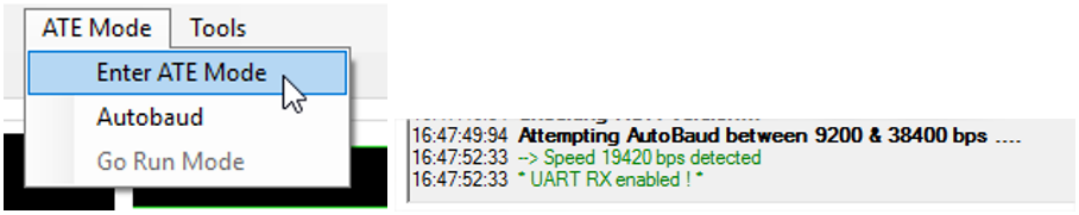
However, if the accuracy is outside the UART tolerances, the GUI might not be able to communicate with the STNRG012 and the ATE mode might fail.

To avoid this failure, the GUI has an autobaud function, which tries to communicate with the STNRG012 chip at various speeds.

So, if the ATE mode sequence fails, select the autobaud feature.

If the GUI is able to communicate with the chip, it displays the resulting UART speed and applies it for the ATE mode.

Figure 17. **Autobaud menu and results**

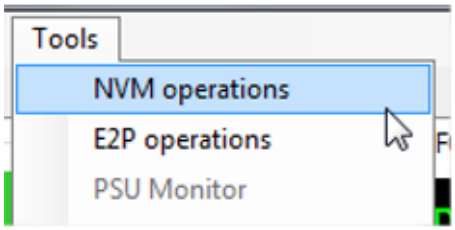


*Note: The computed speed is not updated in the [**Application Settings**] menu.*

6.4 NVM operations

In the ATE mode, NVM operations are accessible via the **Tools** menu.

Figure 18. **NVM operation menu**



All the power supply customization parameters are stored in the STNRG012 on-chip NVM memory.
You can read/write the NVM by:

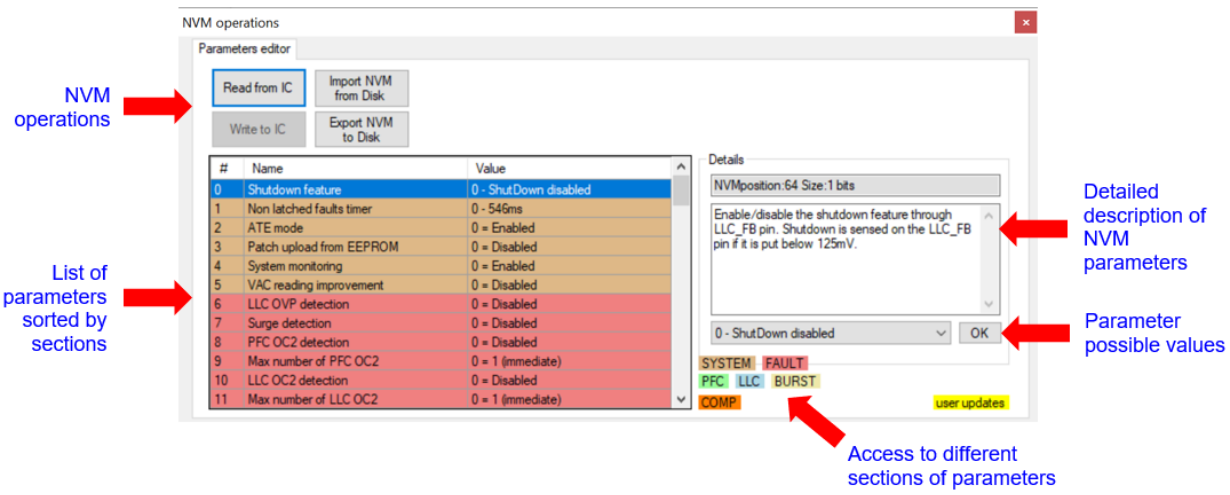
1. using the [**NVM r&w**] tab, mainly to write the complete NVM without knowing the parameters;
2. using the [**NVM editor**] to edit specific parameters.

6.5 NVM editor

The NVM editor provides an intuitive way of changing the STNRG012 system parameters, which are about 80.
These parameters are stored in the 32-byte NVM on-chip memory.
The GUI maps the NVM parameters to the 32-byte memory.

Danger: *Take care when changing the NVM parameters. Improper settings can lead to the offline converter destruction.*

Figure 19. **NVM editor window**



The NVM options are:

- [**Read from IC**]: reads the NVM content from the chip;
- [**Write to IC**]: writes the current NVM content to the chip;
- [**Import NVM from disk**]: reads an NVM file (.INIT format) previously stored;
- [**Export NVM to disk**]: writes the current NVM on the disk (.INIT format).

Note: Writing the NVM parameters is possible only in the ATE mode.

6.5.1 How to change parameters

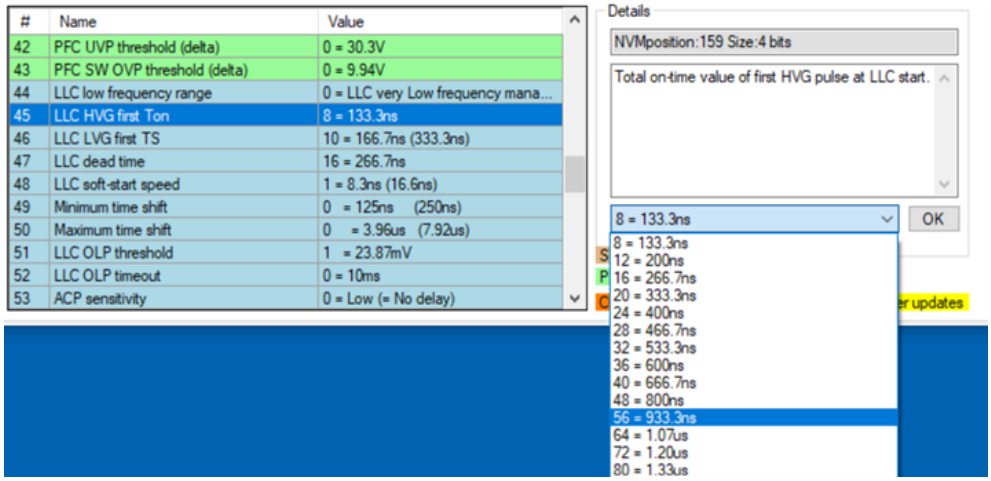
Step 1. Click on the parameter to edit.

To access quickly the parameters, you can click on the relevant section (system, fault, LLC, PFC, etc.), which displays a detailed description of that parameter.

Step 2. Click on the [**Combo**] box to choose the value.

For each value, the internal firmware value is also shown.

Figure 20. Parameter change



Step 3. Press [**OK**] to validate the change.

Once the parameter has been changed and is different from the current NVM, it is highlighted in yellow.

Step 4. Click the [**Write to IC**] button to program the NVM memory.

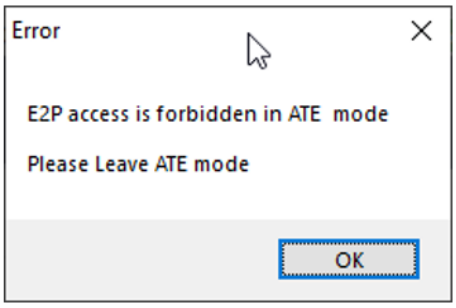
7 E²P operations

Important: *STNRG012 shares the E²P interface (SDA/SCL) with the UART interface to minimize pin count. During the normal operation (switching), the optional E²P is only accessed at boot and when a fault occurs.*

It is possible to access E²P in the normal mode, but this might cause conflicts due to simultaneous access by the STNRG012 metering information (UART) and the GUI accessing the E²P.

As a consequence, the STNRG012 metering messages must be disabled, but this is only possible if the UART uplink communication is enabled. Otherwise, E²P must be accessed while STNRG is disabled (that is, no mains nor an external V_{CC} are required thanks to the internal V_{CC} generation in the ATE mode).

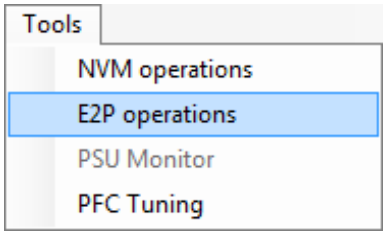
Figure 21. E²P access in the ATE mode error



E²P operations are accessible via the [**Tools**] menu, as detailed below:

- shut down the mains;
- press the [**GUI Force VCC**] button to apply the VCC without entering the ATE mode (the SDA is not pulled low);
- go to the [**Tools**] menu and select [**E2P operation**].

Figure 22. E²P operation menu



7.1 E²P dump

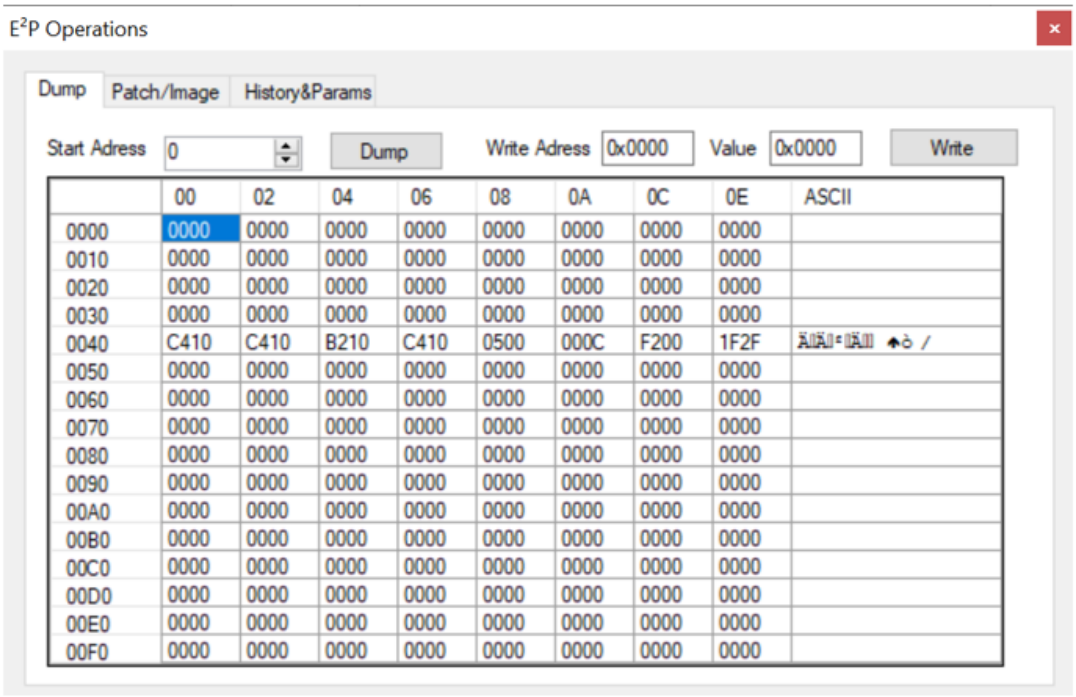
This feature allows displaying the content of the external E²P. It is mainly used for debugging.

It also allows changing a memory value by either clicking the address to be changed or by pressing the [**Write**] button in the [**Address**] and [**Value**] boxes.

Table 4. E²P mapping

Area	Meaning
0x0000-0x0022	Unused
0x0040-0x0047	Event history data
0x0048-0x004F	Running time, error counter, and power on/off cycle counter
0x0080-0x0089	Definition of cold/hot patch addresses
0x0090-0x0A80	Patch area

Figure 23. **E²P dump tab**



7.2 E²P patch and image upload/download

This feature allows manipulating the entire E²P images and also programming the patch.

7.2.1 Full E²P image operation box

This tab button allows:

- reading the full E²P image and save it to the disk
- computing the E²P checksum
- comparing E²P to an existing image
- writing the entire E²P using an image previously saved on the disk (performing the E²P parameters and E²P patch programming in a single step)
- erasing the entire E²P

*Note: If an E²P is connected, the STNRG012 firmware does not support an empty (FF) image. The E²P must be cleared using the [**All 0s pattern**]. Alternatively, a full image can be written (provided by ST).*

Both [**Erase**] and [**Write**] operations need confirmation to be saved.

7.2.2 Patch programming box

There are two different types of patches:

- cold, downloaded from E²P to XRAM just before the IC starts switching
- hot, downloaded after IC has started switching operations

We use only the cold patch in this example.

Note: Usually, you do not have to specify the patch type (hot/cold). Patches are delivered as a full E²P image.

Important: Do not change the XRAM and E2P address.

To program a patch, follow the steps below.

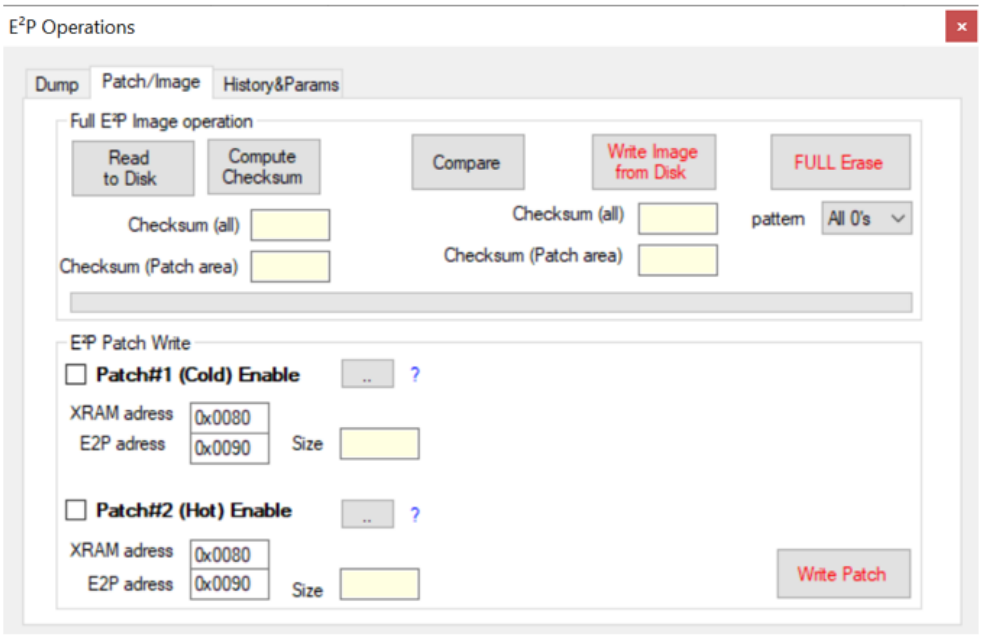
Step 1. Click on the [..] button to select the patch to be used.

Note: Only the .bin format is supported.

Step 2. Tick the associated [**Check**] box.

Step 3. Press the [**Write Patch**] button.

Figure 24. E²P full import/export and patch operation tab



7.3 E²P parameter editor

This feature allows editing the factory data parameters and clearing the event history data.

Step 1. Press the [**Read**] button to read the content of the E²PROM.

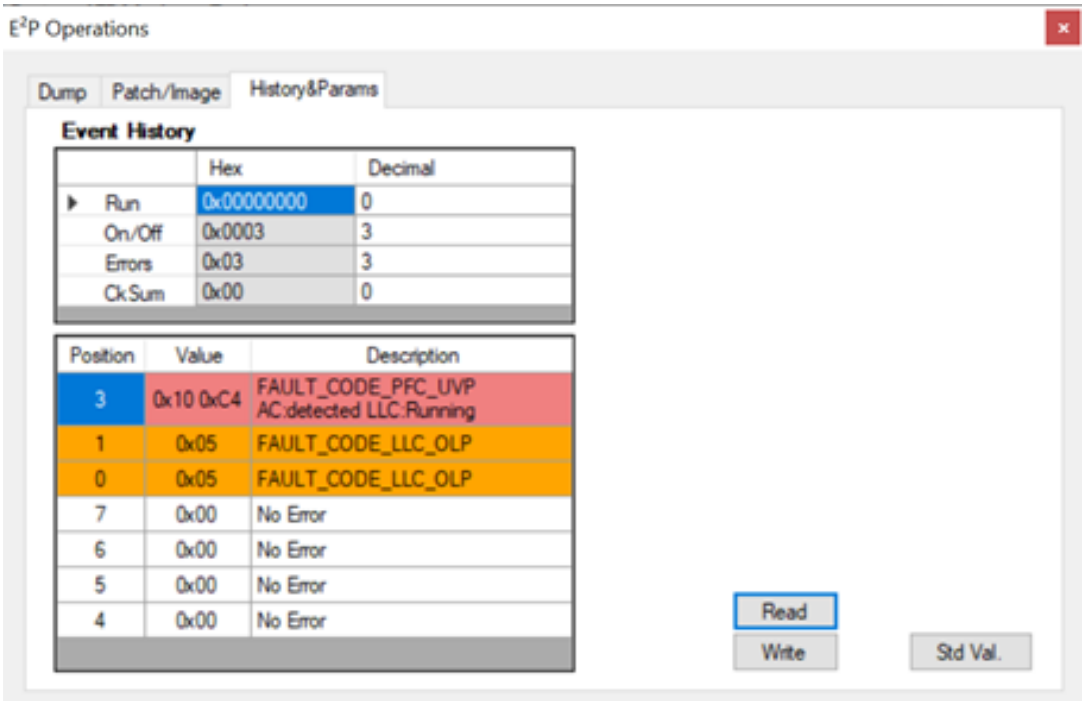
Step 2. Press the [**Write**] button to write the displayed values to the E²PROM.

Step 3. Press the [**Std Values**] button to fill the table with the default values.
If you want to write the values to the E²P, you have to press the [**Write**] button.

Note: The hex fields are only given for reference: they cannot be edited.

Important: It is recommended to edit these field with the STNRG012 in ATE mode or via a VCC externally powered.
If the parameters are written while the STNRG012 is running, they are overwritten when STNRG012 is shut down.
When the STNRG012 is powered up, it makes a copy of the event history in its RAM, which changes during the active phase. At shutdown, the RAM content is overwritten in the E²P, hence the E²P content is overwritten.

Figure 25. E²P parameter editor tab



8 Additional tools

The GUI embeds some additional tools useful during the power supply integration phase.

8.1 PSU monitor

Note: A specific patch is needed to access this feature. Contact STMicroelectronics sale office for details.

The PSU monitor provides similar information to the power metering, but only on a specific request by the GUI (different from the power metering information sent continuously by STNRG012).

In this case, the GUI reads the STNRG012 memory directly and additional information can be retrieved.

Figure 26. PSU monitor window

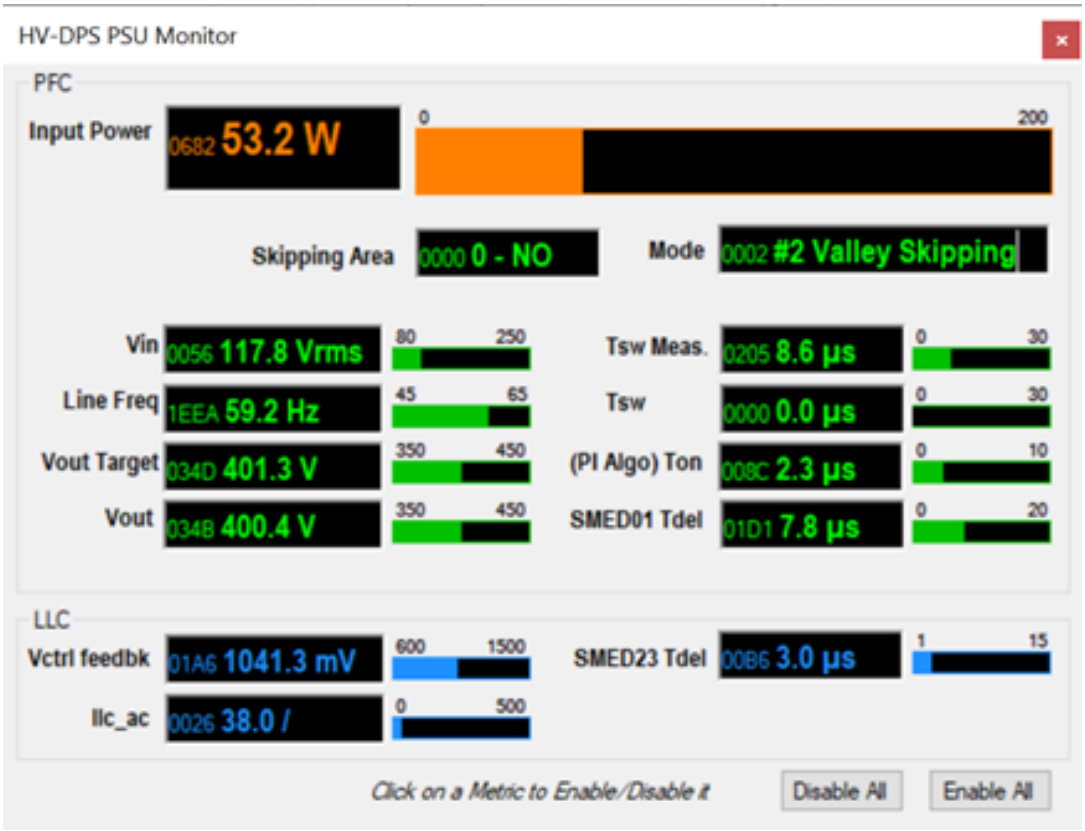


Table 5. PSU monitor: PFC/LLC features

PFC	
Input power	Estimated PFC power corrected by V_{IN}/PAM/mode/Tdel parameters
V_{IN}	Mains voltage in RMS value
Line Freq	Mains frequency
V_{OUT} target	PFC output voltage (target)
V_{OUT}	PFC output voltage (measured)
Tsw measurement	PFC switching period measured at the input line peak
Tsw	PFC switching period estimated in DCM for T_{off} and T_{on} calculation
(PI Algo) T_{on}	Ton added computed by the power integration algorithm
Smed01 T_{del}	Duration between PFC pulsed (DCM mode only)
Skipping area	PWM applied to PFC during a semi-cycle at low power
Mode	PFC mode of operation
LLC	
Vctrl feedback	Output voltage of the error amplifier after opto-coupler, used for the LLC feedback loop
llc_ac	LLC anti-capacitive mode indicator
SMED23 Tdel	Value of the LLC time shift

9 PFC calibration

The STNRG GUI can calibrate the PFC parameters thanks to the embedded wizard, which features:

- PFC parameter semiautomated calibration
- THD improver manual tuning
- Possibility to change manually some parameters to test the overall behavior
- Graphical representation of the mode switch
- PFC parameters are updated in the RAM; when the calibration results are satisfactory, you can store the parameters in the NVM for permanent use
- Manual or automated working mode

Note: Use the automated mode (a GPIB adapter is required). The embedded driver only supports the Chroma equipment. The SCPI commands issued are very generic, so it should work using other tools from other manufacturers (for example, Agilent/Keysight) but this is not guaranteed.

Otherwise, it is possible to use the manual mode, which is semiautomated. In this case, the GUI detects some events and tries to minimize the user's actions on the tools.

9.1 Principle

The calibration scope is to manage correctly:

- the PFC mode change (DCM, valley skipping, TM)
- the skipping area threshold (also called phase angle modulation)
- the THD improver parameters

The PFC supports five different modes: DCM (low power), 3/2/1 valley skipping, and TM (high power).

The mode change is based on the estimated output power and the switching period.

Table 6. PFC parameters to be calibrated

Parameter name	Typical value	Description
PFC THD improver base	5 (10 mV)	Base current of the THD improver (ReCOT functionality on PFC_CS pin)
PFC THD improver gain	0 (no gain)	Gain of the THD improver ramp (ReCOT functionality on PFC_CS pin)
PFC Min Pin Vskip	2176	Minimum PFC power to force a mode switch to DCM
PFC Max Pin Vskip (delta)	2560	Maximum PFC power to force a mode switch towards TM (offset with respect to PFC Min Pin Vskip)
PFC delta Pin Vskip	352	Correction factor to minimize discontinuities while switching among different PFC modes
PFC maximum DCM power	3072	PFC power threshold to switch from DCM to valley skipping mode. It also sets the PFC on-time during DCM. Above this threshold, the system goes into valley skipping
PFC Min Tsw Vskip	448 (134 kHz)	Minimum PFC switching period (max. frequency) to force a mode switch to DCM
PFC Max Tsw Vskip	608 (87 kHz)	Maximum PFC switching period (min. frequency) to force a mode switch to TM
Skipping area threshold	1536	PFC power threshold for the skipping area

Note: The PFC calibration has to be performed for a given design (based on the component choice, supply output power, etc.). It is not necessary to perform the calibration for each unit created.

9.2 PFC protection

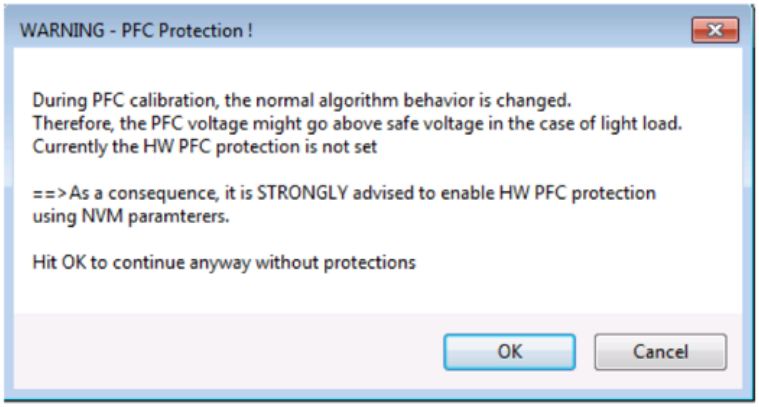
During the PFC calibration, the normal algorithm behavior is changed. In particular, the PFC mode change algorithms can be disabled to adjust some parameters.

Danger: *Consequently, the PFC bulk voltage can rise above the normal voltage at light load, leading to component (for example, bulk capacitors, MOS) damage and even destruction.*

To prevent any damage, the GUI performs a preliminary safety check by reading the NVM content and checking if the PFC hardware protection is set. If not, a warning is displayed (as shown in the figure below), inviting the user to update the NVM accordingly.

Important: You can override the warning, but it is under your own responsibility.

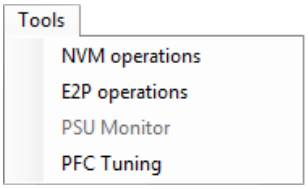
Figure 27. PFC protection warning window



9.3 Step by step calibration example

To start calibration, go to the [**Tools**] menu and select [**PFC tuning**]. The following window pops up.

Figure 28. PFC tuning menu



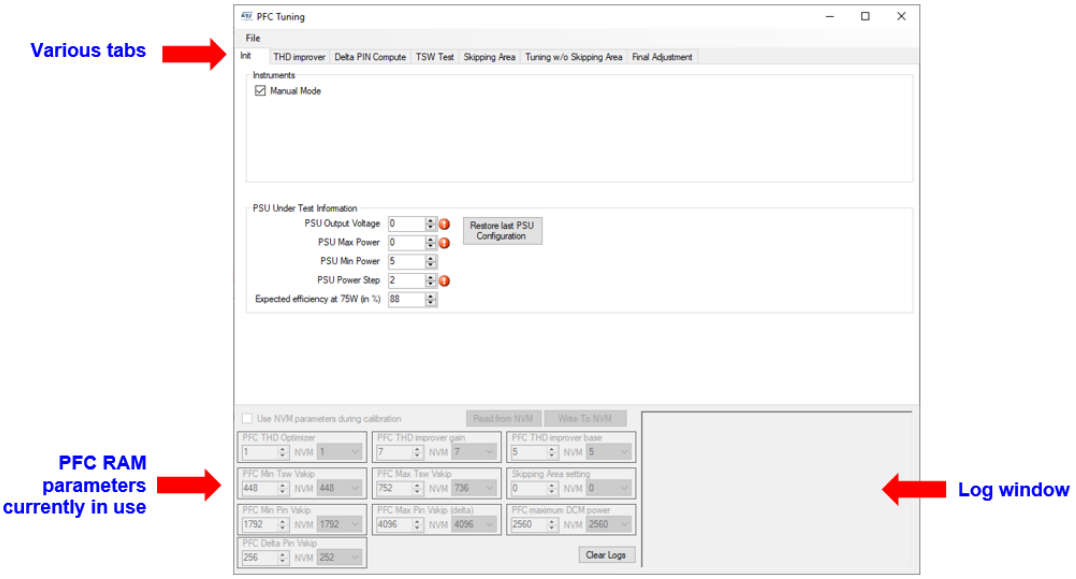
Note: To perform the calibration, the GUI has to communicate with the STNRG012. Since the UART communication is disabled by default in the running mode, a special procedure (similar to the ATE mode) is applied to enable the UART communication, which requires switching the power supply on/off, as prompted by the GUI.

As previously mentioned, during the tuning operation phases, the GUI modifies the RAM parameters but writes nothing in the NVM (so, changes are not persistent).

9.3.1 Tuning calibration tabs

The following picture shows the PFC tuning main window with tabs that represent the various steps to follow during the calibration.

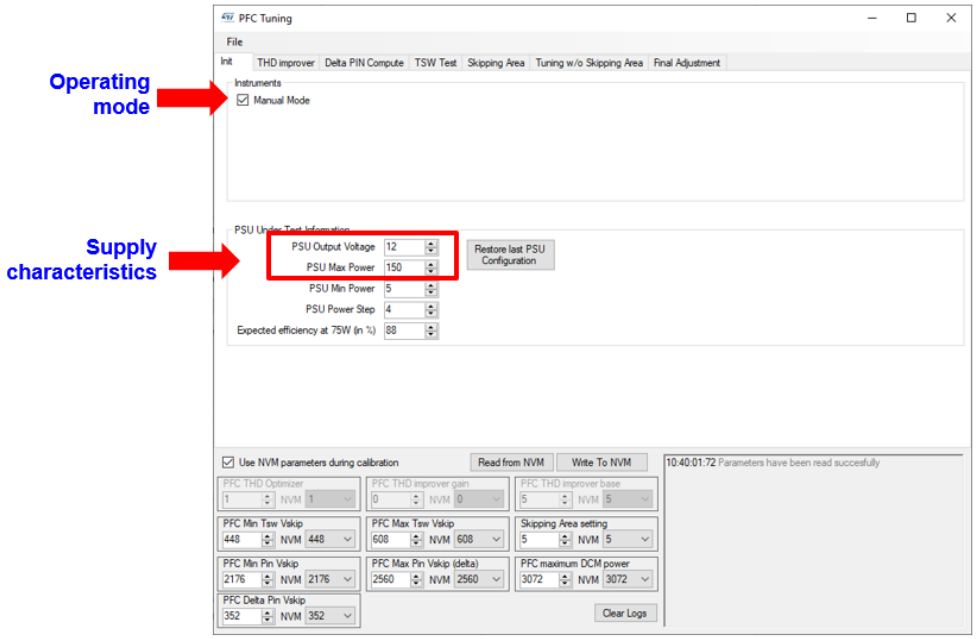
Figure 29. **Main PFC tuning tab**



9.3.2 Mode of operation

You have to choose the operating mode (manual or not) and the power supply characteristics.

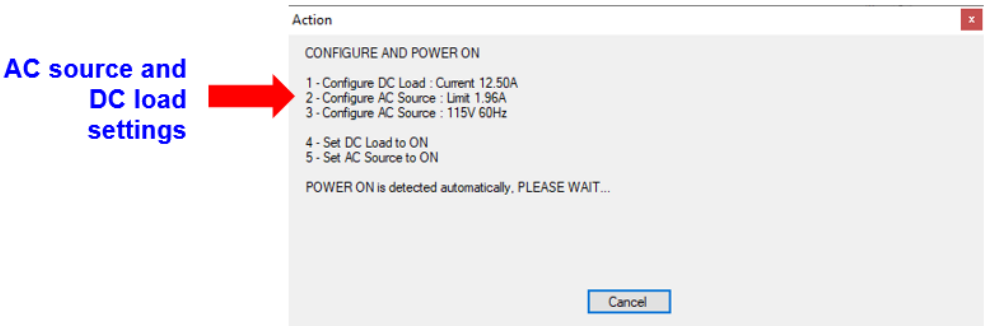
Figure 30. **PFC tuning parameter tab**



If manual mode is selected, you have to change manually the AC source voltage and DC load current: a window pops up each time an action is necessary.

The STNRG012 is able to detect when the AC source is switched on/off, which limits the interactions.

Figure 31. **Manual mode: AC source voltage and DC load current selection**



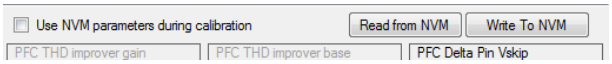
See Figure 32 for NVM options.

As the NVM storage space is limited, some PFC parameters have to be uploaded with a lower resolution than the NVM values. To ensure that the calibration reflects the real mode of operation, the calibrated values can be rounded to the closest one ticking the [**Use NVM parameters**] check box.

The [**Read NVM**] button transfers the content of the chip NVM memory to the GUI. This is useful to revert to the original calibration stored in the NVM, if the current calibration is not satisfactory.

The [**Write NVM**] button is used to write the current calibration in the chip NVM memory.

Figure 32. **NVM parameters: RAM and NVM values**



9.3.3 Step 1: THD improver

The first step is to calibrate the THD improver since the parameters directly affect the output power reported.

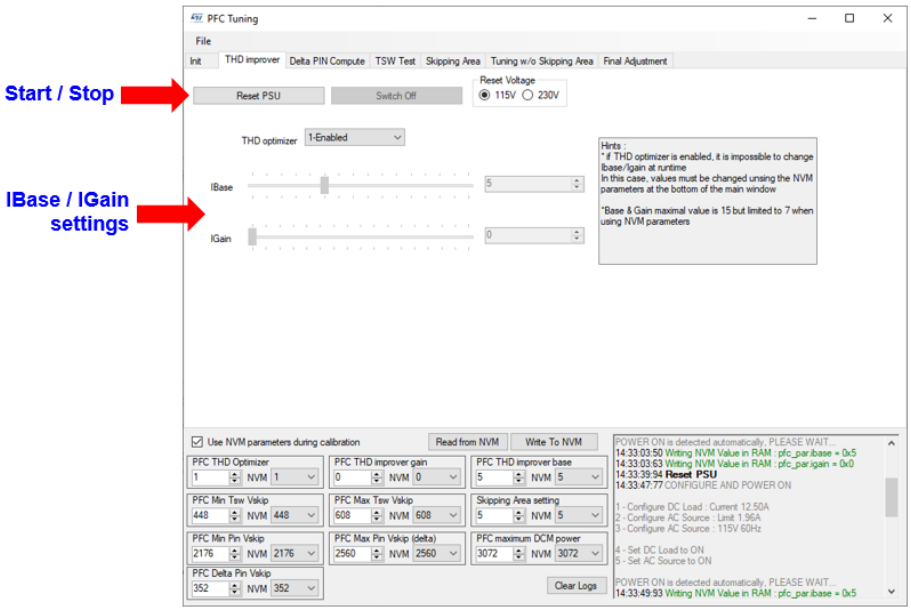
This feature helps to improve the AC current total harmonic distortion (THD) and the power factor (PF) value.

After pressing the [**Reset PSU**] button, follow the instructions.

Once the PSU starts, you can tune the IBase and IGain settings.

There is only one possible setting, so you have to try various loads and AC voltages to find the best trade-off across all the operation modes.

Figure 33. **PFC tuning: THD improver tab**



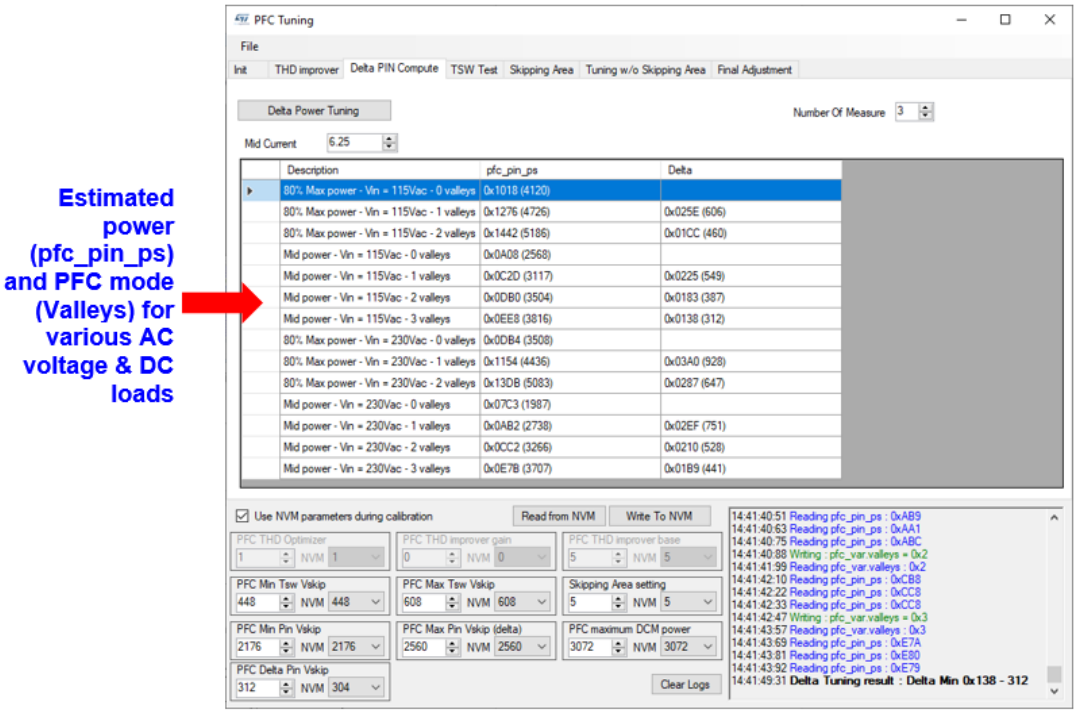
9.3.4 Step 2: delta pin calibration

The second step is to calibrate the delta pin correction factor when changing from one mode to the other.

To compute the delta pin calibration, the GUI:

- disables the PFC mode change on the basis of the frequency and power values
- forces the PFC mode and records the associated estimated power
- repeats the operation for 2x load current and 2x AC input voltage
- computes the delta pin correction factor

Figure 34. PFC tuning: delta pin calibration



9.3.5 Step 3: TSW test

After computing the delta pin, the third step is to determine the mode changes based on the PFC frequency.

You have to select manually the minimum/maximum bounds for the PFC frequency.

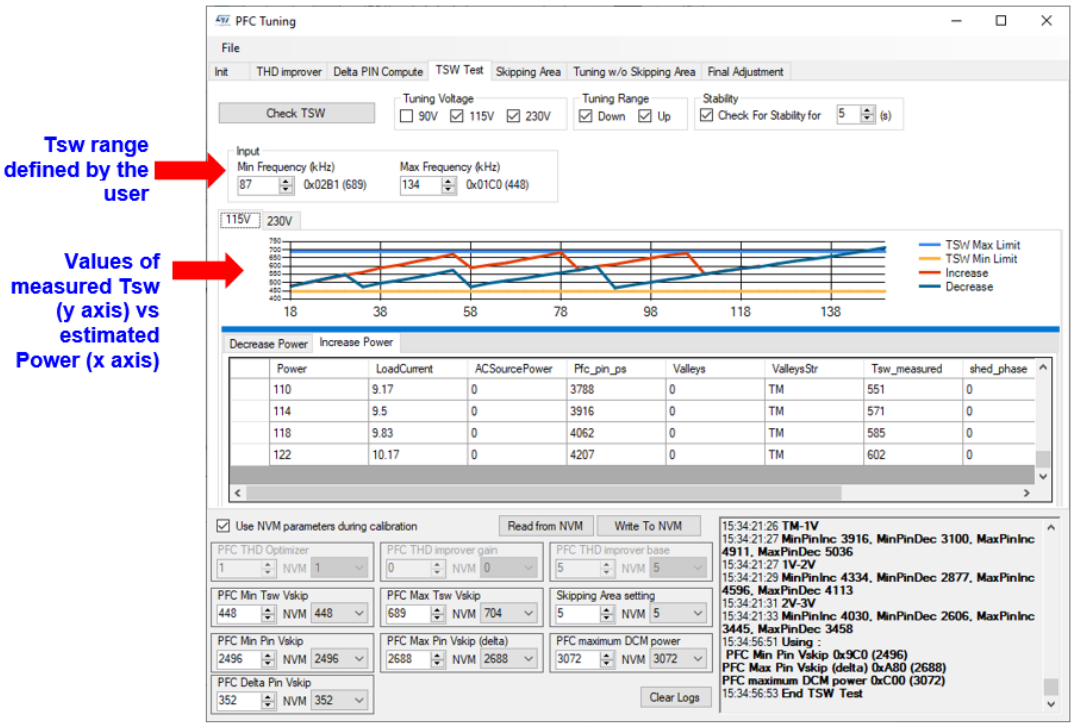
During the normal operations, if the PFC actual frequency crosses this boundary, a mode change is required.

To check the [**TSW range**], in the GUI:

- PFC mode change is enabled on the basis of the PFC frequency (the mode change based on the estimated power is disabled)
- the output power is swept across the operating range
- the switching frequency is monitored; check that the mode change is stable (that is, the IC does not jump continuously from one mode to another because it crosses always a frequency limit during the mode change), adjusting the limits if needed.

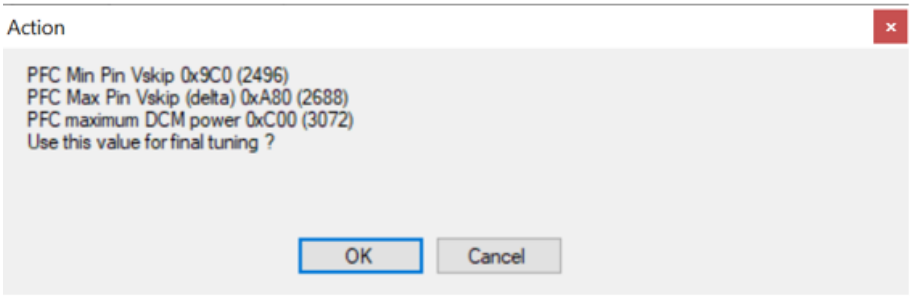
Note: If you select the manual mode, the GUI asks to ramp-up and down the DC load smoothly.

Figure 35. **PFC tuning: determining TSW limits**



Finally, the GUI computes the PFC parameters.

Figure 36. **PFC tuning: PFC min./max. pin and DCM power**



9.3.6 Step 4: skipping area threshold

The fourth step consists in setting the skipping area power threshold. Below this threshold, the system applies the power skipping, that is, it shuts down the PFC operation before or after the peak of the AC cycle, like PWM.

This improves the overall efficiency but obviously degrades the THD/PF figures. For this reason, it has to be performed at low power.

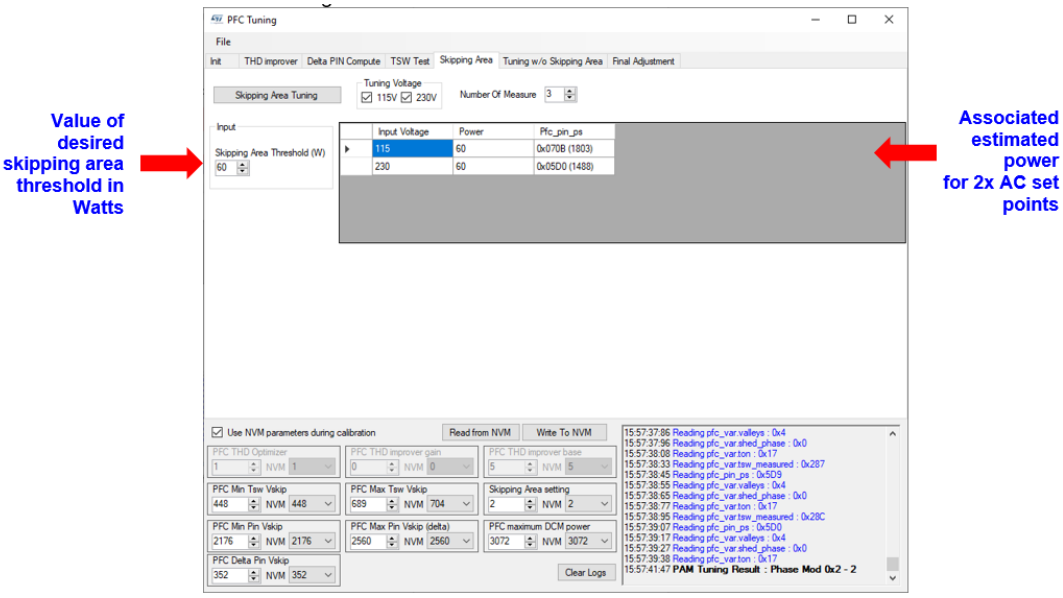
The threshold is up to the customer.

To determine the skipping area threshold, the GUI:

- enables the PFC mode change (both the PFC frequency and estimated power-based)
- sets the DC load to match the desired power
- measures the estimated power at various AC voltages

The threshold is the average of the measured values.

Figure 37. **PFC tuning: determining the skipping area threshold**



9.3.7 Step 5: final verification

Once all the steps have been performed, the GUI simply checks the supply behavior.

To perform the final verification, the GUI:

- enables the PFC mode change (both the PFC frequency and estimated power-based) as the normal mode
- disables the skipping area mode (the focus is on the mode changes)
- sweeps the DC load across all the operating modes and checks if it is among the limits

Figure 38. **PFC tuning: final verification without the skipping area tab**

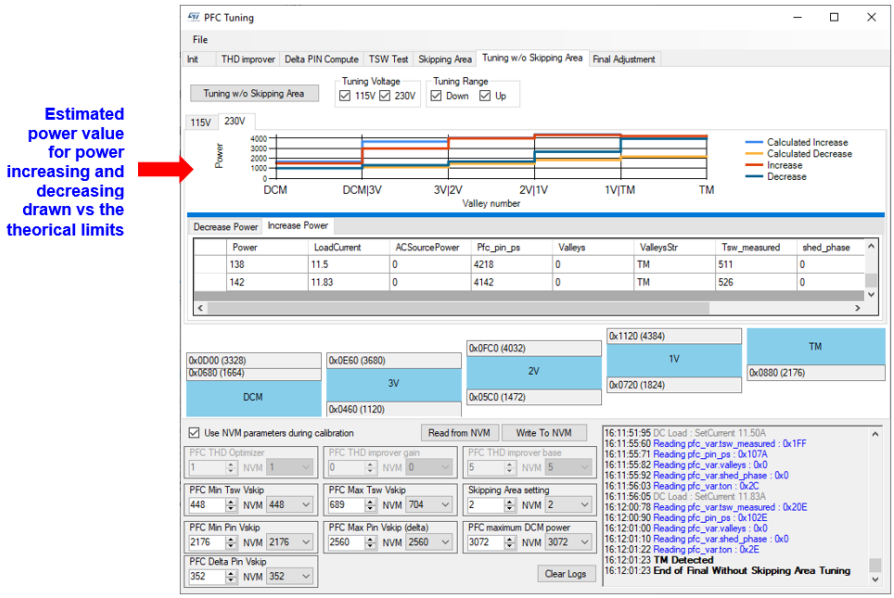
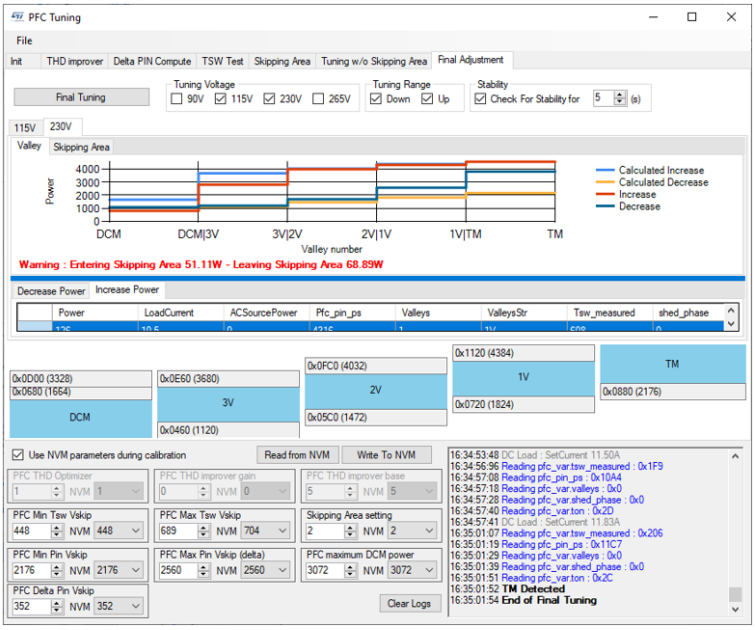


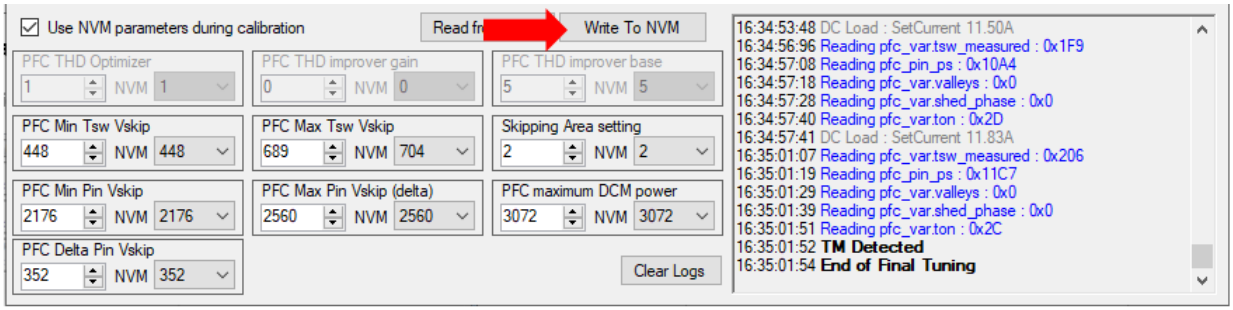
Figure 39. **PFC tuning: final verification**



9.4 PFC parameter update

Once you are satisfied with the PFC calibration performance, you can update the NVM to make the parameters permanent, by simply clicking on the [**Write to NVM**] button.

Figure 40. **PFC tuning: writing the NVM values**



Revision history

Table 7. **Document revision history**

Date	Revision	Changes
19-Apr-2022	1	Initial release.

Contents

List of tables

List of figures