User manual

Getting started with the STEVAL-PCC020V1: USB to I²C UART interface board and associated GUI for STNRG products

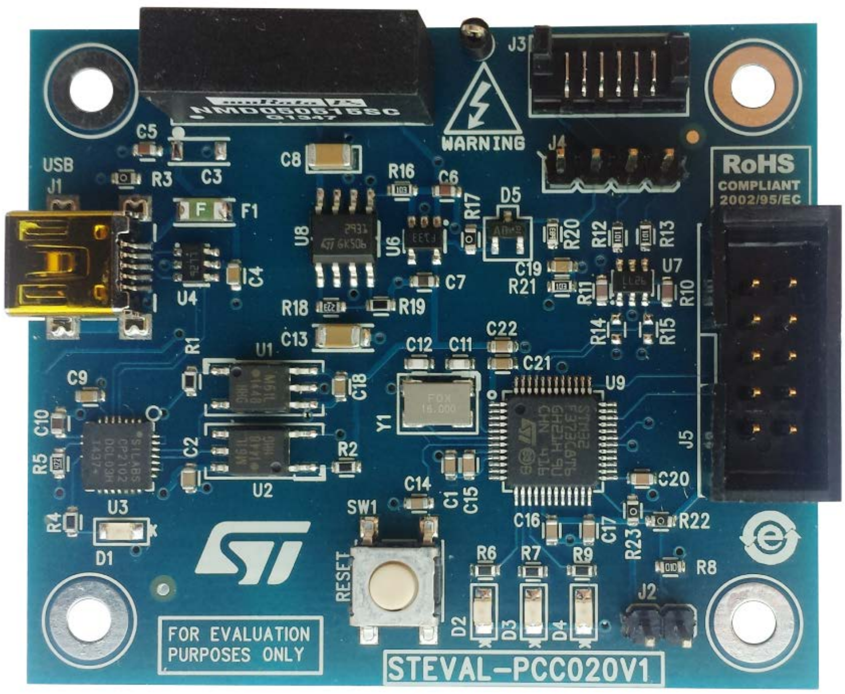
Introduction

The STEVAL-PCC020V1 USB to I²C/UART board interfaces a Windows®-based PC with STNRG digital power supply controllers such as STNRG011.

It is basically a bidirectional bridge between USB and I²C/UART buses and embeds an on-board power supply to communicate and program the STNRG IC without need of mains.

The associated GUI allows monitoring the status of the digital controller in real-time and tuning specific parameters according to customers' needs.

Figure 1. **STEVAL-PCC020V1 interface board**



1 Interface board aim

Figure 2. Customer typical application shows a customer typical application based on STNRG011 for the power supply section.

The host microcontroller receives information only from the STNRG011 using an opto-isolated connection: STNRG011 transmits metering information (instantaneous power) continuously, and the black box content at reset.

Hence, the host microcontroller does not have access to the STNRG011 optional E²PROM where the patch and black box history are stored.

Figure 2. Customer typical application

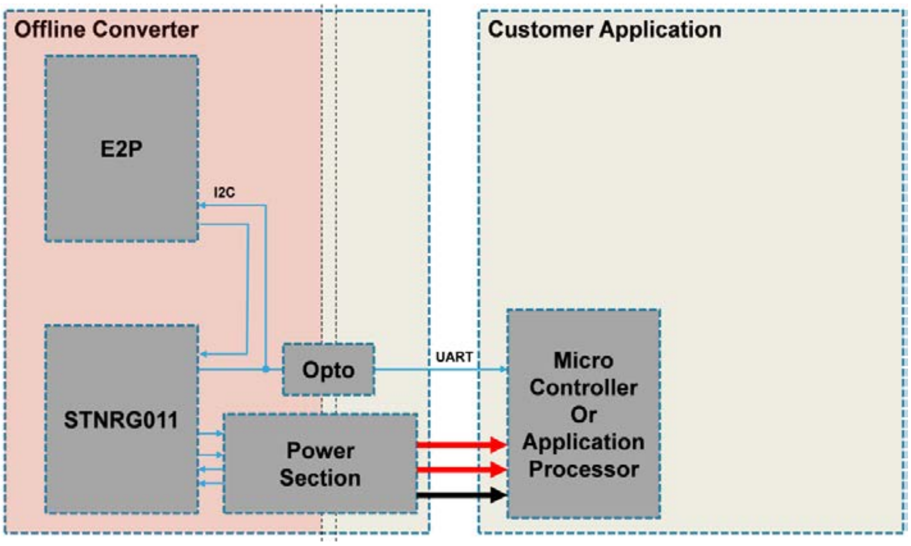


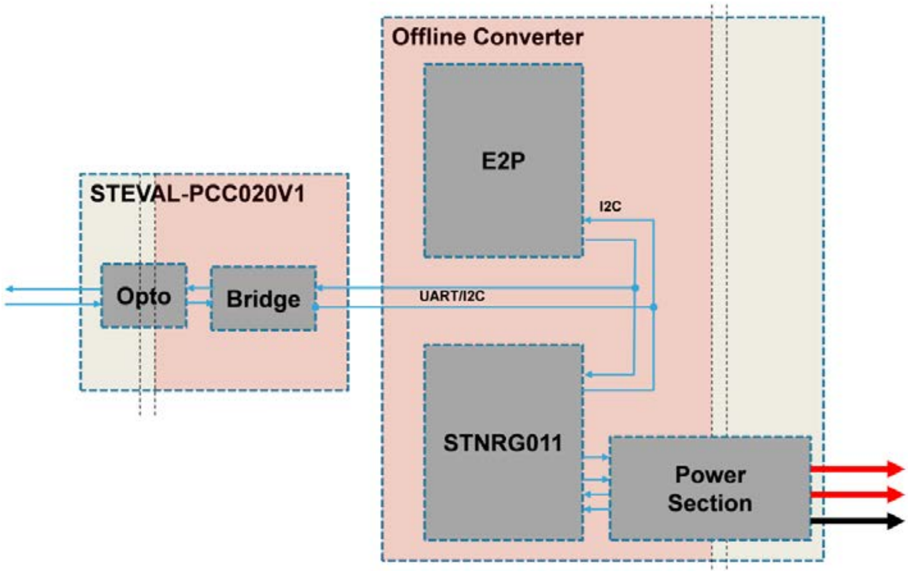
Figure 3. STNRG011 in demo/debug configuration shows the STNRG011 on the STEVAL-PCC020V1 interface board or during debug configuration.

In the latter case, you can access the external optional E²PROM using the I²C protocol to program the associated patches and reset the black box content.

You also have to access STNRG011 using UART bidirectional communication to:

- program the STNRG011 NVM content to change specific parameters according to customers' needs
- display the system specific parameters in real-time to check its behavior during the debug and integration phases.

Figure 3. **STNRG011 in demo/debug configuration**



To minimize STRGN011 pin count, UART and I²C interfaces share the same pins. The interfaces are not isolated from the mains as they are located on the offline converter primary side.

Important: *This adapter board is exclusively designed to interface with STNRG011 products.*

In the final customer application, the tasks performed by the interface would be handled directly by the host microcontroller or the application processor.

2 Getting started

2.1 STEVAL-PCC020V1 interface board overview

The STEVAL-PCC020V1 interface board key features are:

- Bidirectional communication between PC (USB) and STNRG011
- Self-powered from the USB line
- On-board 19 V generation for STNRG011 programming
- Electric Isolation between USB and other board electronics
- I^2C bus running at up to 1 MHz
- A UART bus running at 19200 bps
- UART and I^2C bus muxed together on the same interface
- On-board firmware upgrade through USB port
- Display power metrics (AC voltage, PFC power)
- Access to STNRG M24C32 optional E^2PROM (used to store patch, calibration and event history data)
- Program NVM settings
- RoHS compliant

2.2 GUI overview

The GUI key features are:

- Runs on Windows XP, Windows 7 (.NET 4.0 framework needed)
- Real-time monitoring of the digital controller status
- Access to STNRG011 NVM parameters
- Access to STNRG011 external E^2PROM for patch upload, calibration and event history
- Embedded PFC calibration wizard

2.3 Package contents

The STEVAL-PCC020V1 package includes:

- Hardware
  - the interface board
  - a 1.8 m USB A to USB mini-B cable
  - a 15 cm 6-wire flat cable for target connection to the STNRG011 device
- Software
  - USB drivers
  - PC GUI installation package

Note: The complete software package is available at www.st.com.

2.4 System requirements

To use the STEVAL-PCC020V1 interface board, you need a PC with Windows® operating system.

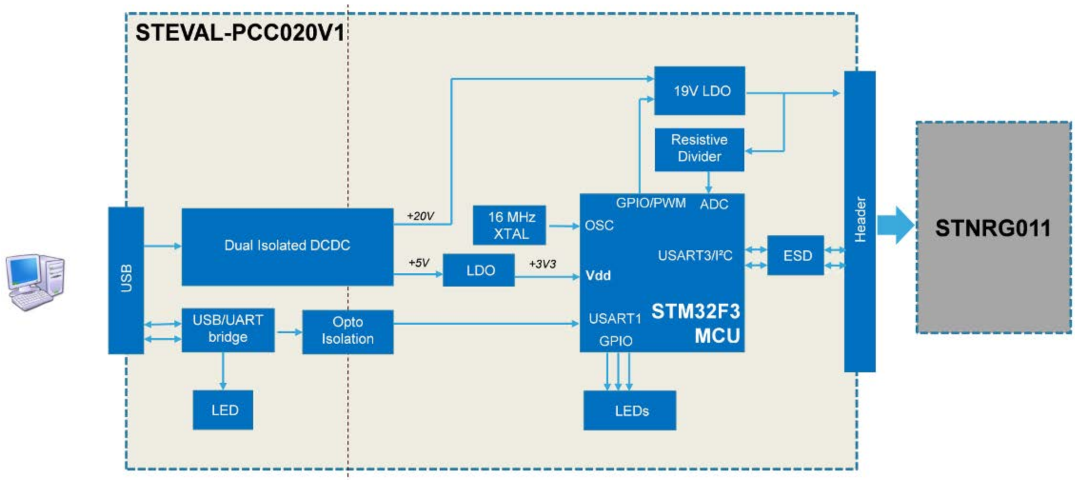
The graphical user interface (GUI) works with Microsoft Windows XP or later versions and .NET Framework 4.0.

Note: The .NET Framework 4.0 is not included in the Windows XP installation package.

3 Hardware description and setup

3.1 Block diagram

Figure 4. **STEVAL-PCC020V1 block diagram**



3.2 Galvanic isolation

The STNRG011 has to be placed on the offline converter primary side: the galvanic isolation between the USB and the remaining electronic of the board prevents any voltage from reaching the host PC and causing electrical damage or interference.

3.3 Power supply

The STEVAL-PCC020V1 interface board is self-supplied via the 5 V USB connector.

This voltage directly supplies U3 and the related circuitry.

A dual isolated DC-DC module (U5) is used to supply the remaining part of the board, maintaining the isolation among the PC and the target sides.

U5 generates two supplies, loosely regulated (+5 V and +20 V).

3.3.1 MCU subsystem supply (5 V)

The +5 V supply is later converted to a stable and clean +3V3 thanks to the linear regulator U6, which is always on.

3.3.2 VCC generation (20 V)

The +20 V is always generated from +5 V and +15 V cascaded together (VOUT2- is referenced to VOUT1+ in place of ground).

This voltage is later on supplied by the linear regulator U8 which has the following roles:

- to generate a stable +18.5 V;
- to act as a switch; U8 is enabled thanks to the MCU GPIO PA14 configured in open drain mode. When the MCU wants to enable the VCC generation, PA14 is driven low.

D5 provides an OR-ing diode which, by default, is short-circuited by R17 resistor (0 W).

3.3.2.1 *VCC soft start*

At VCC generation switch on, VCC is typically decoupled by a 100 to 200 µF capacitor on the STNRG011 device.

If the regulator is switched on abruptly, an inrush current is generated that cannot be sustained by the upfront DC-DC converter, which then enters current limitation.

Since the +20 V is generated by cascading +5 V and 15 V, the current limitation also impacts the +5 V supply (hence the MCU).

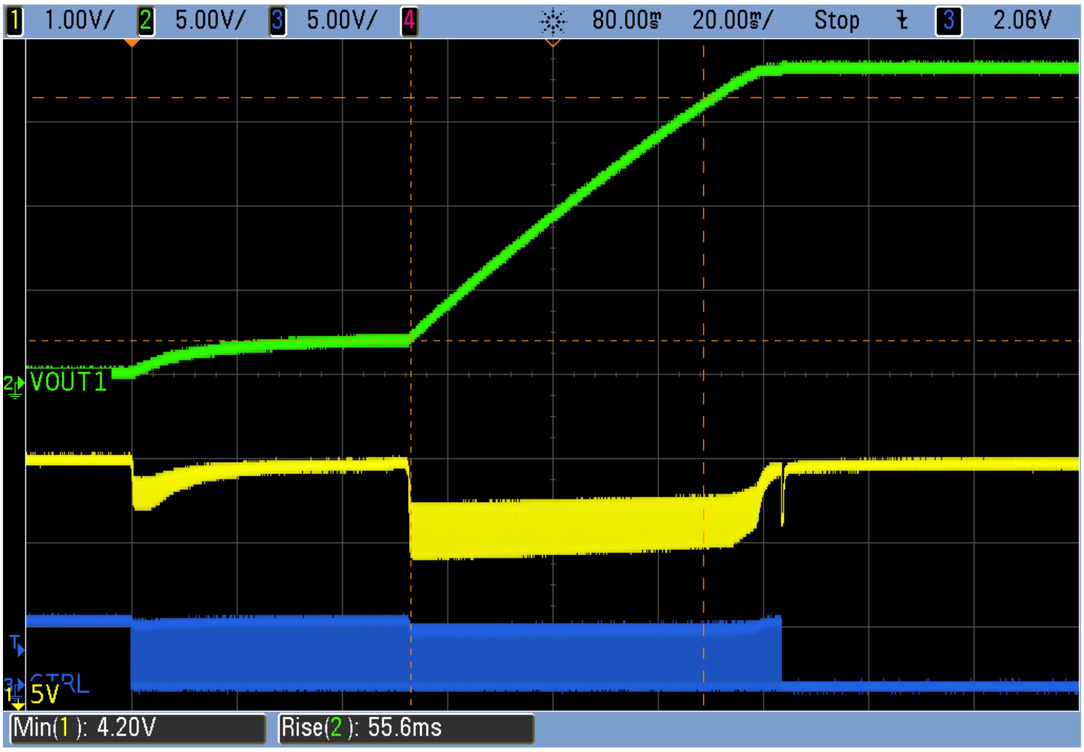
When the MCU supply drops below the **PowerOnReset** threshold, the MCU resets and the board reboots.

To avoid this behavior, the linear regulator U8 is switched on via a **soft start** using a PWM enable signal (which limits the current on the upfront DC-DC).

When VCC has reached a stable value (that is, the VCC capacitor is charged), the enable signal remains in the steady-state condition (always on, so always low).

Soft start phase usually lasts about 120 ms.

Figure 5. **VCC ramp-up typical waveform**



3.3.2.2 *NVM programming*

The STEVAL-PCC020V1 interface board provides a VCC voltage to the target device that is high enough for an NVM programming operation.

STNRG011 programming requirements are +18 V and 35 mA max. current.

If the VCC on the target device is < 17 V, the programming VCC can be simply connected to the target VCC through a couple of OR-ing diodes.

The 19 V supply current delivered is limited to 100 mA by the on-board LDO (U8).

3.4 USB bridge

The communication between the STEVAL-PCC020V1 and the PC is managed by the latter as a standard serial peripheral; the IC U3 converts the USB connection into a virtual COM port (refer to the electrical schematic).

By default, the virtual COM port operates at 921600 bps.

A yellow LED near the mini-B USB connector turns on when the CP2102 has been recognized (enumerated) by the host operating system.

The VCP RX and TX signals are then isolated thanks to the opto-couplers U1 and U2 and connected to the STM32F3 (U9) microcontroller USART1.

Important: *The USB port and the remaining part of the board are isolated from the mains.*

The microcontroller performs:

- **Conversion between the host UART and I²C protocols**
  - The I²C speed can rise up to 1 MHz (maximum speed allowed by the STNRG011).
  - The STM32F3 allows bidirectional communication between the PC and the target device through the UART to I²C conversion.
- **Conversion between the host UART and the STNRG011 UART** .

  This is mainly baud rate matching: STNRG011 operates at 19200 bps, whereas the host UART operates at 921600 bps.

Note: *The microcontroller also manages the muxing of the UART and I²C protocols on the same interface.*

3.5 VCC monitoring

The MCU also monitors the STNRG011 VCC line voltage.

STNRG011 VCC is sampled periodically by the MCU via a simple resistive bridge divider plus a low-pass filter using R20, R21 and C19. The divider ratio is 10/78=1/7.8.

The divided voltage is then sent to STM32F3 PA0 pin on a regular 12-bit ADC.

For instance, this allows preventing the use of the on-board VCC when the STNRG011 is already operating.

Note: *This feature accuracy is ±100 mV.*

4 Using the board

4.1 Board connectors, LEDs and buttons

Figure 6. STEVAL-PCC020V1 interface board connectors

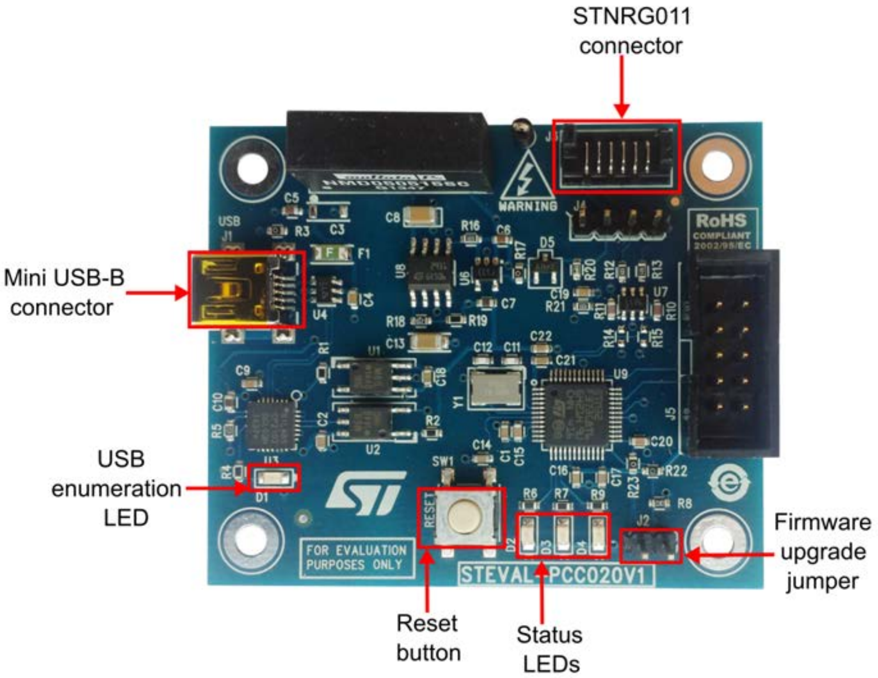


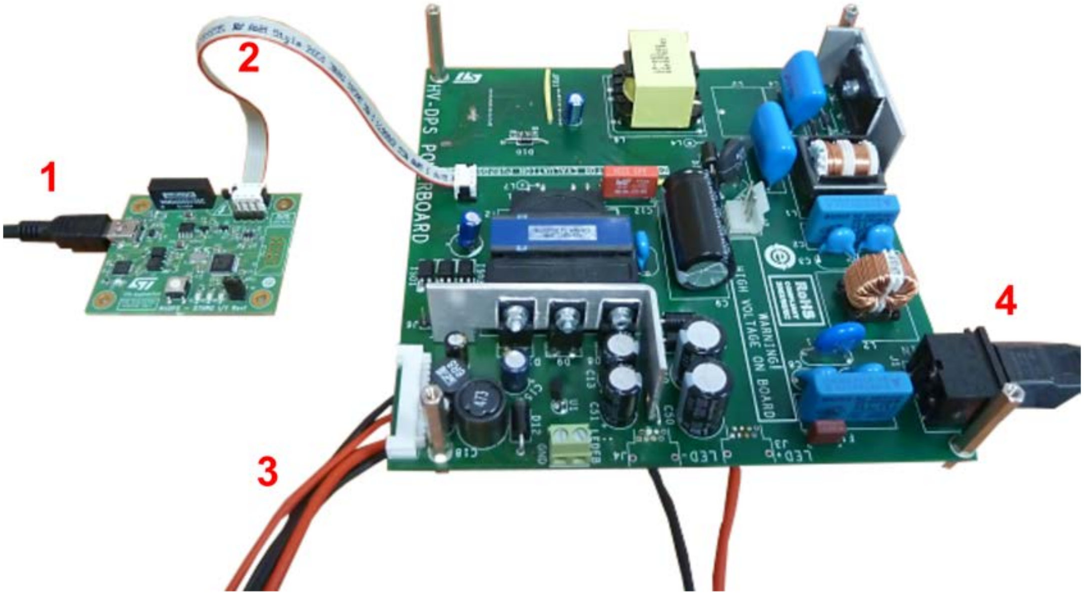
Figure 7. STEVAL-PCC020V1 interface board status LEDs

Table 1. STEVAL-PCC020V1 LEDs (ON, OFF, blinking state)

D1		D2		D3		D4	
ON	OFF	ON	Blinking	ON	Blinking	ON	OFF
VCP recognized by the PC	VCP not recognized/ inactive	Normal operation	Firmware error	Waiting for the STNRG011 frames	Receiving STNRG011 frames	Internal VCC enabled	Internal VCC disabled

4.2 How to connect the STEVAL-PCC020V1 interface board to the offline converter

Figure 8. STEVAL-PCC020V1 interface board typical connection



Procedure

- **Step 1.** Connect the STEVAL-PCC020V1 interface board to a PC via a USB cable
- **Step 2.** Connect the interface board and the offline converter board together through the 6-wire flat cable
- **Step 3.** Connect the offline converter to the load
- **Step 4.** Connect the mains

Caution:
You should never plug or unplug the interface board while the connection is running (for example, when the offline converter is running). If the 5 V UART signals and +15 V VCC (typ.) are connected when the GND is not yet connected, the STNRG011 or the interface board might be damaged.

5 Software installation

You have to install the USB driver and the PC GUI before using the STEVAL-PCC020V1 interface board.

5.1 Virtual COM port driver installation (SiLabs CP2102)

To use the STEVAL-PCC020V1 interface board, first install one of the USB drivers located in the CD folder **Driver \CP210x_VCP_Windows**:

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

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

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

5.2 GUI installation

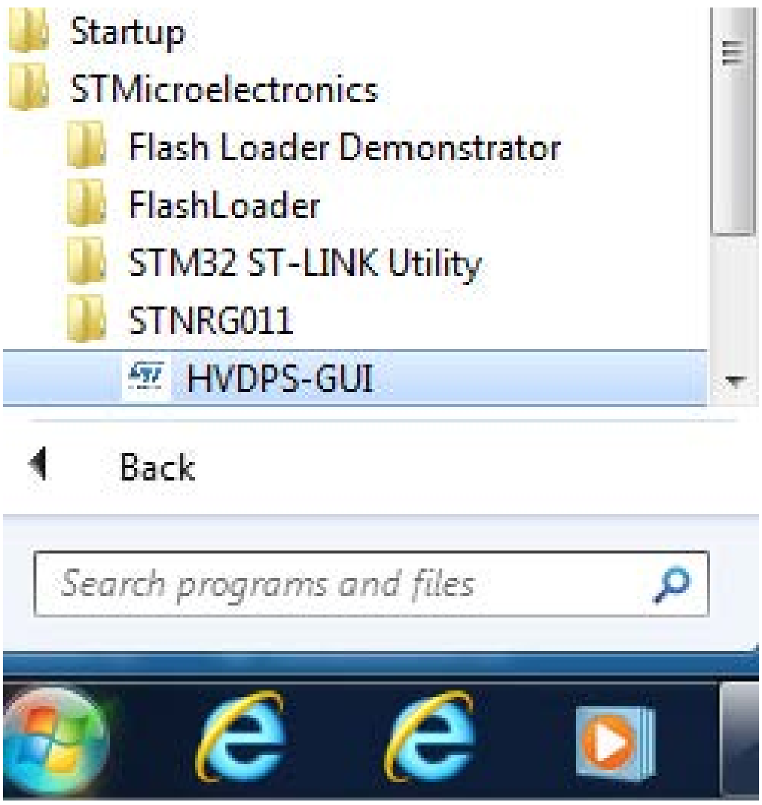
Procedure

Step 1. Launch **HVDPS-STNRG011-Setup.msi**

Step 2. Follow the installation wizard instructions.
By default, the GUI is installed under *C:\Program Files (x86)\STMicroelectronics\STNRG011 GUI*.

The GUI installer creates an icon in the Start menu, under STMicroelectronics\STNRG011.

Figure 9. HV-DPS GUI icon



*Note: If a previous version of the software has already been installed, it **must** be uninstalled through the **Windows Control Panel Uninstall** option.*

6 GUI introduction

6.1 GUI features

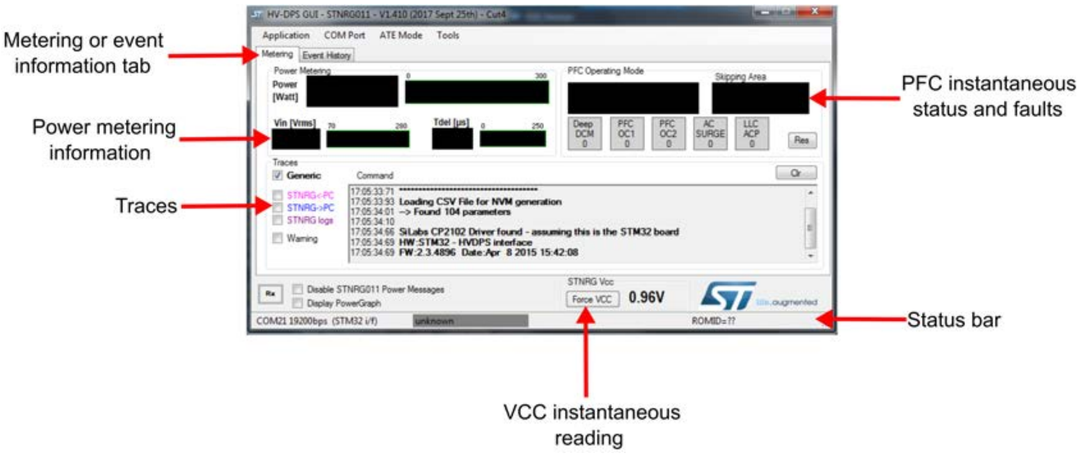
The STNRG011 GUI is designed for debugging power supply applications.

It allows:

- reading instantaneous power metering information and PFC operating modes;
- reading and modifying STNRG011 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).

6.2 GUI startup screen

Figure 10. STNRG011 GUI startup screen



The GUI is split in the following areas:

1. **Menu Bar**: used to select the operation mode, i.e. to communicate with STNRG011, display logs, access E2P directly, NVM programming, etc.
2. **Metering or Event tabs**: displays power metering information or event history
3. **Traces and Status**: internal debug traces and status bar showing the STNRG011 current status

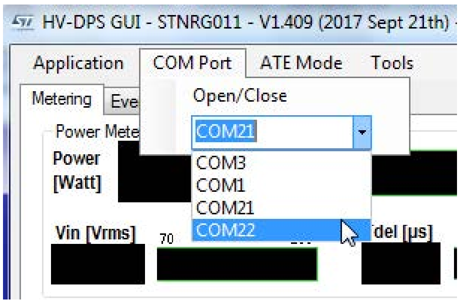
6.3 Connection management

At startup, the GUI detects automatically the COM port to be used (the GUI selects the CP2102-based VCP).

In case of multiple CP2102, you have to manually select the right COM port via the **COM Port** menu.

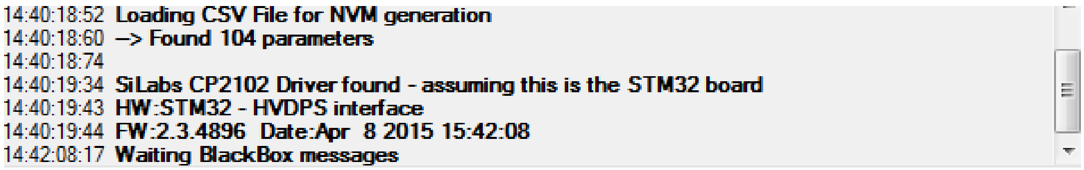
You can also open or close the COM port via the menu shown below.

Figure 11. **GUI COM port selection**



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

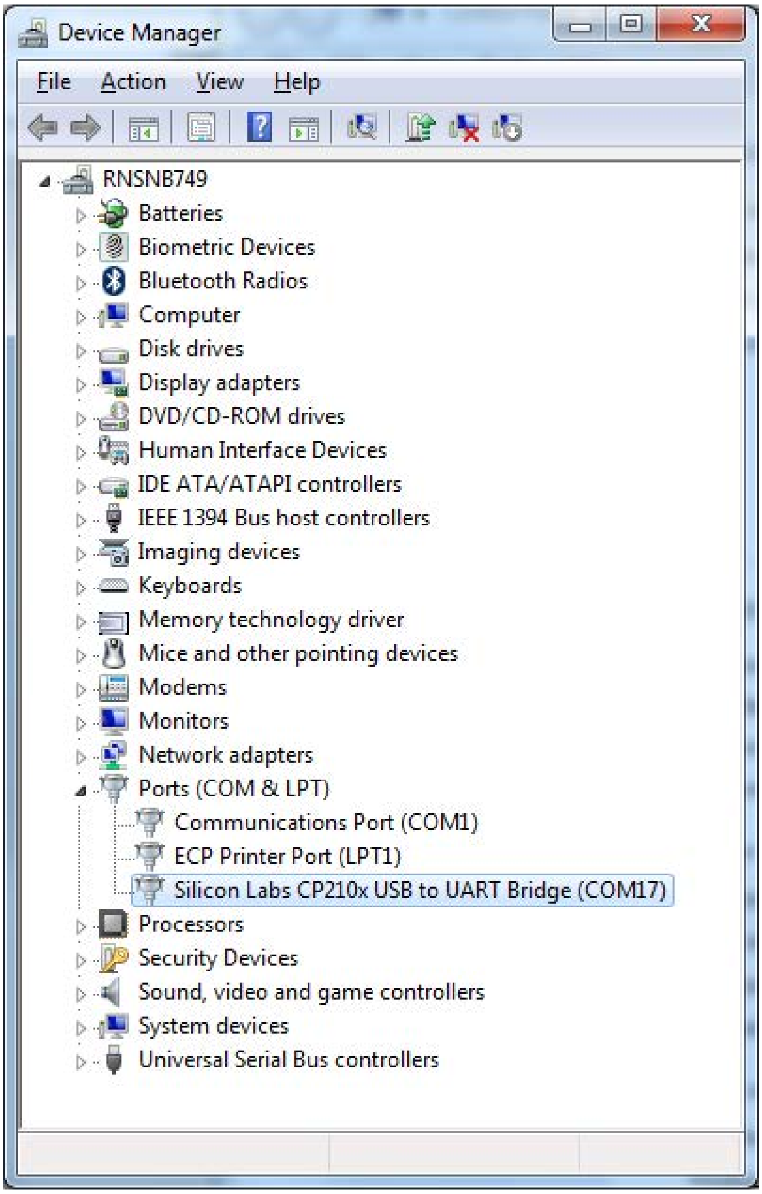
Figure 12. **Traces during GUI connection**



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

Note: *If the GUI does not find a SiLabs-based VCP, an error message appears. Check in the* ***Device Manager*** *if the SiLabs VCP is correctly recognized by pressing plus* ***Pause*** *and selecting* ***Device Manager****, as shown in the following figure.*

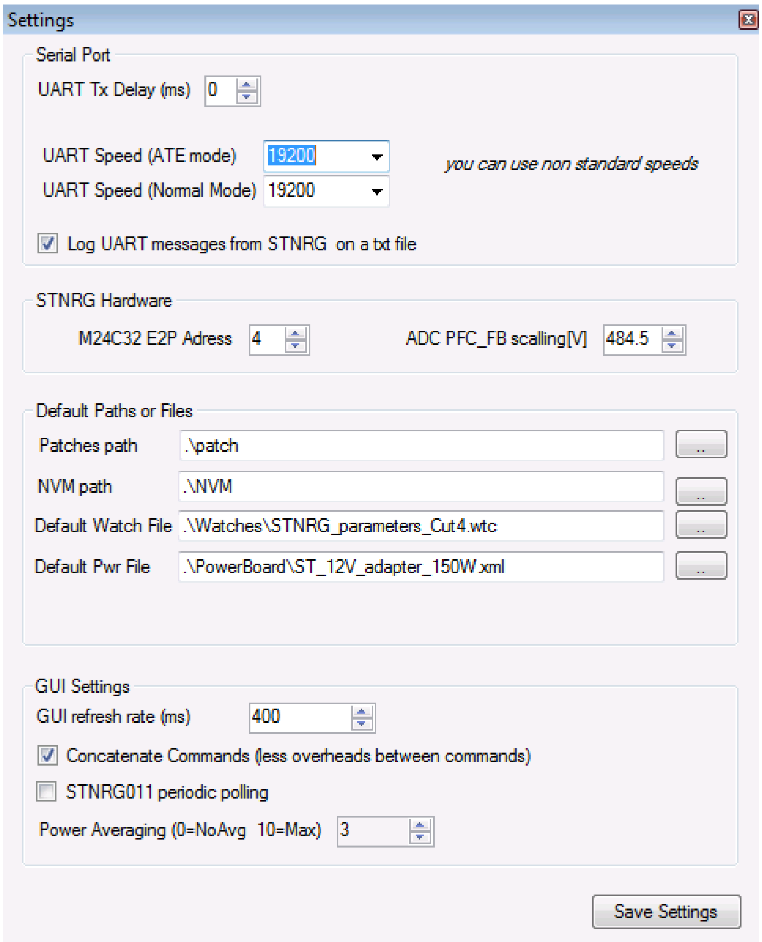
Figure 13. **SiLabs VCP in the Device Manager**



6.4 GUI settings

You can access GUI settings through the Application→Settings menu.

Figure 14. **GUI Settings menu**



Some settings (e.g. GUI refresh rate or power averaging) can be changed in real-time.

Pressing the **Save Settings** button saves the settings into the **config.xml** file.

Table 2. **GUI setting parameters**

Serial Port	
UART Tx delay (ms)	Optional Tx Delay. Leave it to 0ms
UART speed (ATE mode)	STNRG011 UART speed during logging phase
UART speed (Normal mode)	STNRG011 UART speed during ATE mode
Log UART messages from STNRG011	Option to log the UART exchange on a file (uart_trace.txt on the GUI executable directory)
STNRG011	
M24C32 E2P address	Hardware address of the external E2P. STNRG011 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

Serial Port	
DefaultWatch files	Default path for the Watch settings, used to monitor internal variables of the firmware
Default Pwr file	Calibration file for the power metering
GUI settings	
GUI refresh rate[1]	Delay in ms between each GUI refresh
Concatenate commands[1]	Messages sent to STNRG are concatenated to avoid USB overhead (Write1-Write2..Read1-Read2 instead of Write1-Read1-Write2-Read2….)
STNRG011 periodic polling	Periodically polls STNRG011 status
Power averaging	Averaging filter for real-time power display 0 = No averaging 10 = Maximum averaging

1. *Used only when Power Monitor window is active*

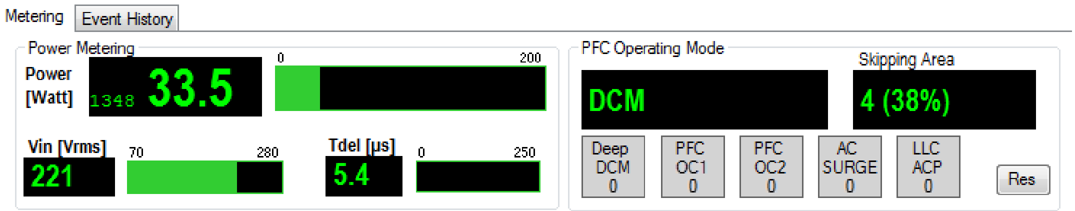
7 GUI normal mode

7.1 Power metering

During normal mode, STNRG011 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 :
  - Vin (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 15. **GUI metering information panel**



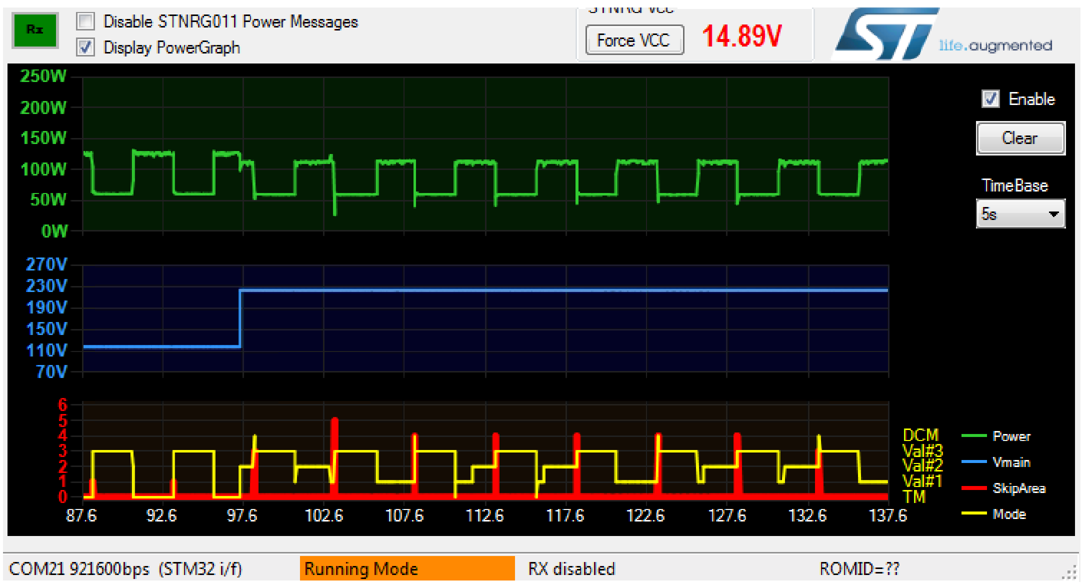
Note: If STNRG011 power messages are disabled, the boxes shown above are empty.

Important: Power metering is not available in ***Burst*** *mode (to save MCU power energy, hence efficiency at low power).*

7.2 PowerGraph report

By checking the box **Display PowerGraph**, the history of PowerGraph event is displayed, to see the long-term stability of the power supply or mode changes versus load.

Figure 16. **GUI PowerGraph report**



7.3 Event history and factory data pre-requisite: E²P

Event history and factory data are available only if an external E²PROM (M24C32) is connected to the STNR0G11. This allows retrieving some information in case of system failures.

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

However, if UART uplink communication is enabled (patch needed), it is possible to send a request to the STNRG011 to immediately send the event and fault history by pressing the **Get Factory Data** button.

Another option is to directly read the E²P content (refer to Section 10.3 E²P parameter editor)

7.4 Faults history

The fault history is stored in the optional external E²P.

Faults are stored in an 8-position circular buffer, hence only the last 8 faults are stored.

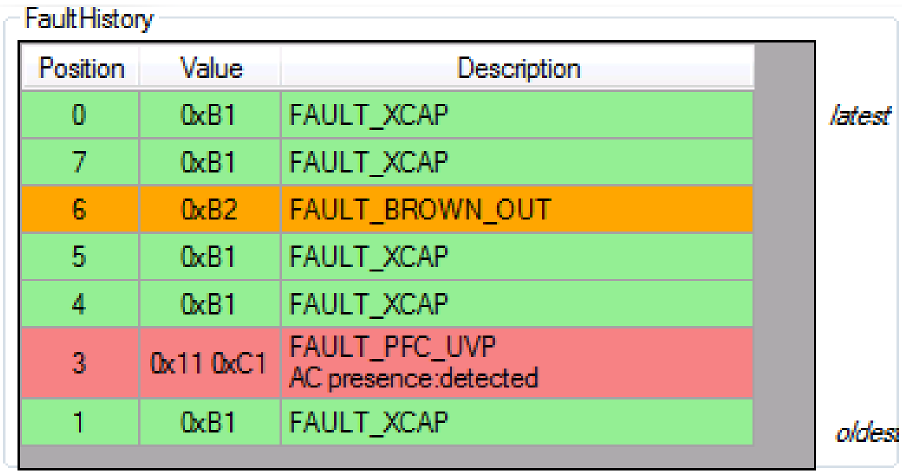
At each power up, the STNRG011 sends the content of the fault history (which is a sort of black box) to the host.

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

Note: There are two types of faults:

- *Standard fault→ **1** position per fault in the circular buffer*
- *Fault with debug information→ **2** positions per fault in the circular buffer; it provides more firmware information.*

Figure 17. GUI fault history



Fault History

Position	Value	Description
0	0xB1	FAULT_XCAP
7	0xB1	FAULT_XCAP
6	0xB2	FAULT_BROWN_OUT
5	0xB1	FAULT_XCAP
4	0xB1	FAULT_XCAP
3	0x11 0xC1	FAULT_PFC_UVP AC presence:detected
1	0xB1	FAULT_XCAP

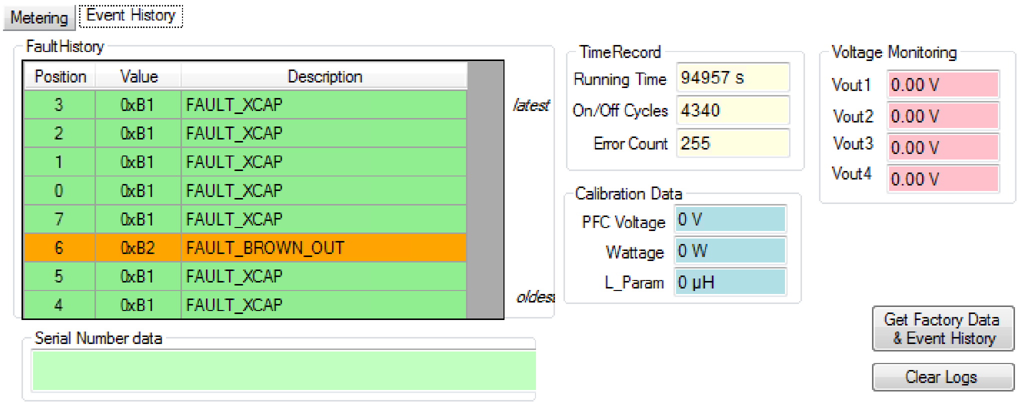
The number of faults stored depends on the fault type (between 4 and 8).

For instance, the figure above shows:

- a fault **PFC_PFC_UVP with AC presence** (position **3**), which is using two positions in the buffer, and the previous fault is at position **1**
- two shutdown events (XCAP discharged→positions **4** and **5**)
- a **BrownOut event** (Vac < 70 Vrms→position **6**)
- two shutdown events (positions **7** and **0**)

7.5 Factory data display

Figure 18. **GUI factory data panel**



Like faults history, factory data are sent at each STNRG011 power up.

The table below shows the factory data parameters and lists some fields as examples for a possible user application.

Table 3. **GUI E²P parameter description**

Voltage monitoring	
Vout1	Customer factory field, not used by firmware nor GUI
Vout2	
Vout3	
Vout4	
Calibration Data	
PFC Voltage	Customer factory field, not used by firmware nor GUI
Wattage	
L_Param	PFC inductance, used by the power computation algorithm
Time Record	
Running Time	Power supply cumulated active time
On/Off Cycles	Number of power supply restart events
Error Count	Number of errors
Serial Number Data	
Serial Number, 20 char	Customer factory field, not used by firmware nor GUI

8 Power metering calibration

8.1 Background

The STNRG011 provides to the host continuous power metering information about:

- PFC integrated power in raw format (resulting from the PID integrator)
- Input voltage
- PFC operating mode and skipping area
- PFC fault status

The instantaneous raw power estimation can be computed by:

$$P_{Raw}[W] = PFC_{lsb} \times \left[128 \times \left(\frac{FSR_{Vin}}{256}\right)^2 \times \frac{t_{smed}}{L}\right] \quad (1)$$

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 the input voltage, it is possible to correct the raw power to deduce the actual power consumption.

As a lot of parameters have to be taken into account, a very simple approach is to use a calibration method for each PFC operating mode.

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

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

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

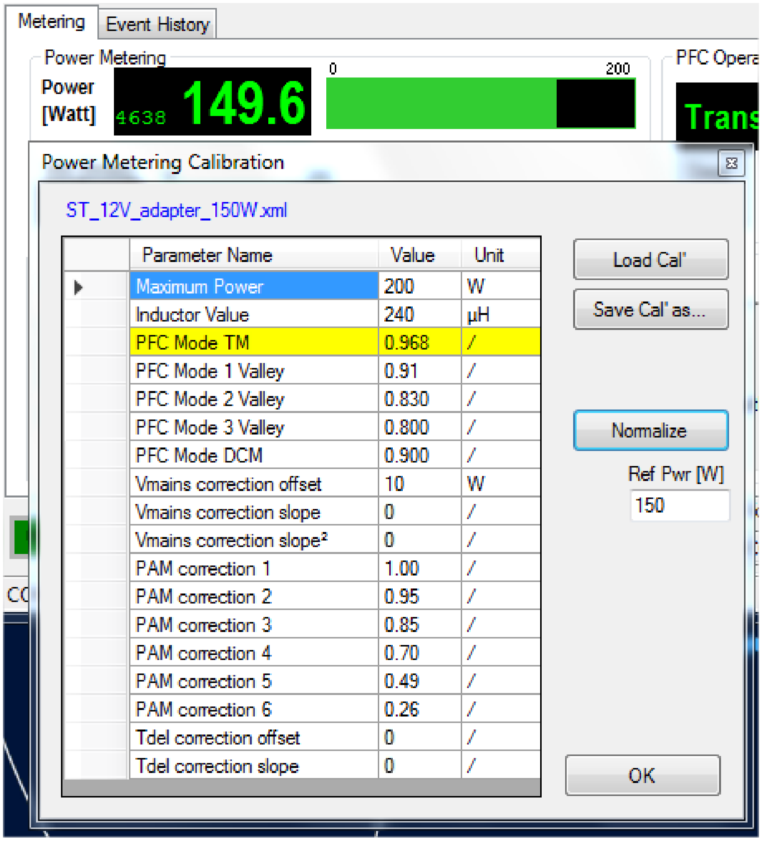
C_{PAM} is the skipping area correction factor (1 for no PAM, 0.26 for minimum PAM)

C_{Tdel} is the DCM mode correction using a first order equation (a+b*Tdel)

8.2 Metering calibration

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

Figure 19. **GUI power metering calibration window**



To calibrate the PFC mode parameters:

Procedure

Step 1. Set a load to make the systems enter the transition mode (near nominal power, e.g. 150 W)

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.

9 ATE mode

9.1 Normal and ATE mode: differences

The STNRG011 supports two main modes of operation:

- **Normal (or running)**: is the GUI normal operation mode (described in the previous chapter)
- **ATE**: is mainly used to program the STNRG011 NVM section and is based on the ATE protocol

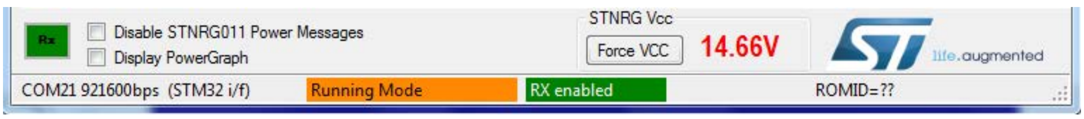
Table 4. **Differences between normal and ATE modes**

Mode	Running (normal) mode	ATE mode
Metering information	Available through the metering protocol (STNRG011 to host)	Not available
ATE protocol	No (default)	Yes
Communication	STNRG011→Host only (default)	Bi-directional
PFC/LLC operations	Yes	No
STNRG011 supply	Self-supplied	External supply
NVM write	No	Yes

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

Note:
*If the ATE protocol is enabled during 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 Asynchronous STNRG011 Power messages** box.*

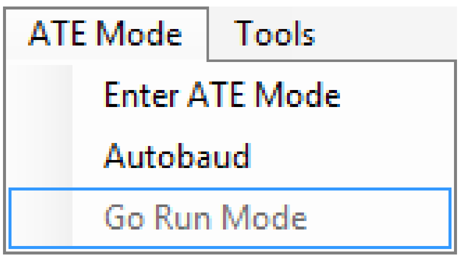
Figure 20. **GUI: disable STNRG011 power messages**



9.2 Entering ATE mode

The ATE mode menu is used to set the STNRG011 in ATE mode.

Figure 21. **GUI ATE mode menu**

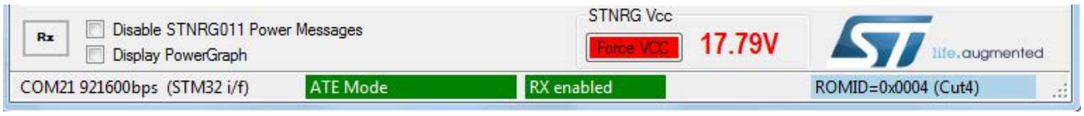


The available options are:

1. **Enter ATE Mode**: attempts ATE mode procedure.
   The GUI:
   a. asserts the SCL pin low (request to enter ATE mode)
   b. switches the internal VCC generation on
   c. checks the ATE mode is effectively working
2. **Auto baud**: is used for internal debug purposes but also to check UART speed. Basically, it performs auto baud and computes the optimal UART link speed
3. **Go Run Mode**: switches off Internal VCC generation

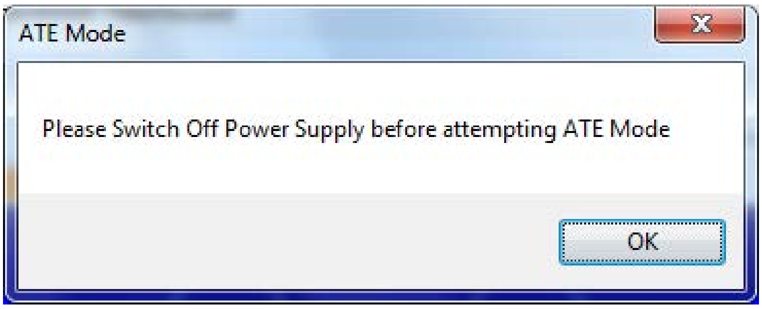
Once the device has successfully entered ATE mode, the status bar is updated and the ROM ID is displayed (here 0x04 for STNRG011 Cut4).

Figure 22. **GUI ATE mode status bar**



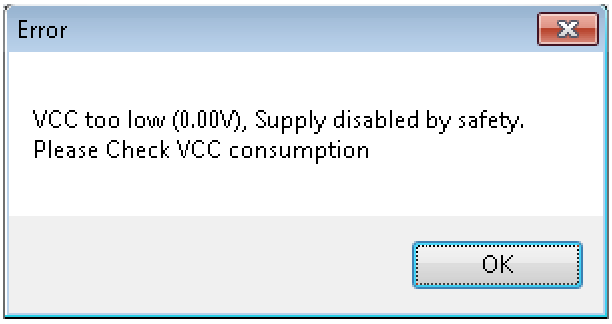
Note: The GUI prevents accessing ATE mode or forcing VCC when it detects the STNRG011 is running in normal mode. The main power supply must be switched off before entering ATE mode.

Figure 23. ***GUI error when trying to enter ATE mode whilst in normal mode***



Once 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 VCC is automatically disabled.

Figure 24. ***GUI error when VCC is applied***



9.3 Autobaud

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

STNRG011 timing relies on an internal calibrated oscillator, but it does not have the accuracy of an external crystal.

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

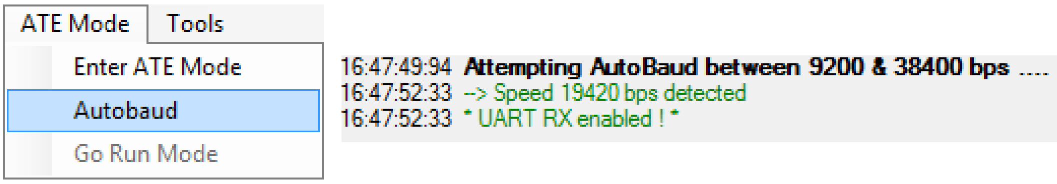
However, if the accuracy is outside UART tolerances, the GUI might not be able to communicate with STNRG011 and ATE mode would fail.

To avoid this failure, the GUI has an **Autobaud** function which tries to communicate with the STNRG011 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 25. Autobaud menu and results

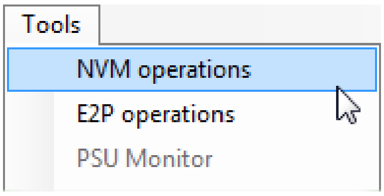


*Note: The computed speed is not updated in the **Application Settings** menu.*

9.4 NVM options

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

Figure 26. NVM operations menu



All power supply customization parameters are stored in the STNRG011 **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

9.5 NVM editor

The NVM editor provides an intuitive way of changing the STNRG011 system parameters, which are about 80.

Those parameters are stored in the 32-byte NVM on-chip memory.

The GUI maps the NVM parameters to the 32-byte memory.

Danger:

You must take care when changing the NVM parameters. Improper settings can lead to the offline converter destruction.

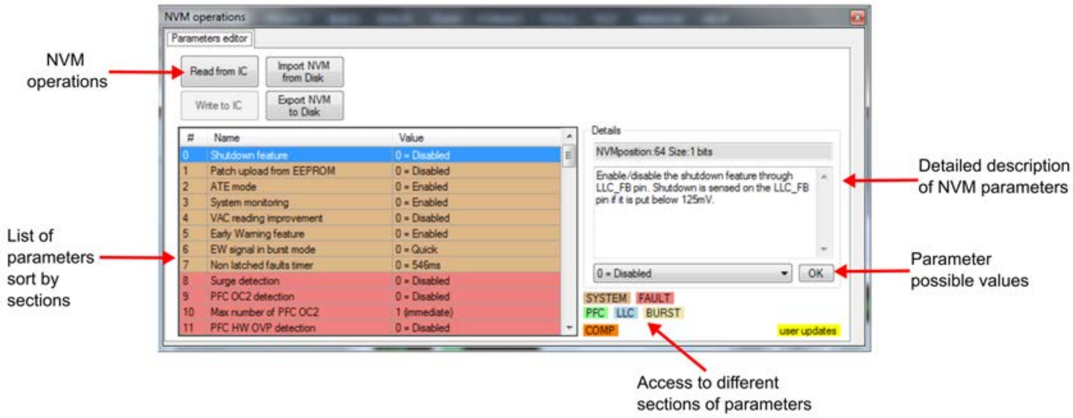


Figure 27. NVM editor window

9.5.1 NVM options

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

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

9.5.2 How to change parameters

Procedure

Step 1. Click on the parameter to edit.

To quickly access parameters, you can click on the relevant section (System, Fault, LLC, PFC, etc.) to display a detailed description of the parameter.

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

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

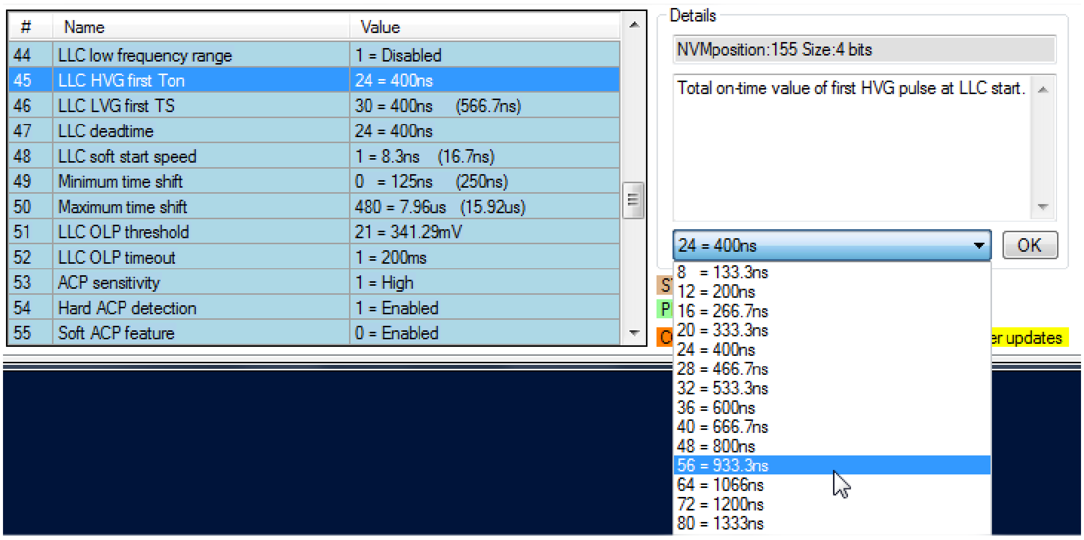


Figure 28. Parameter change

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.

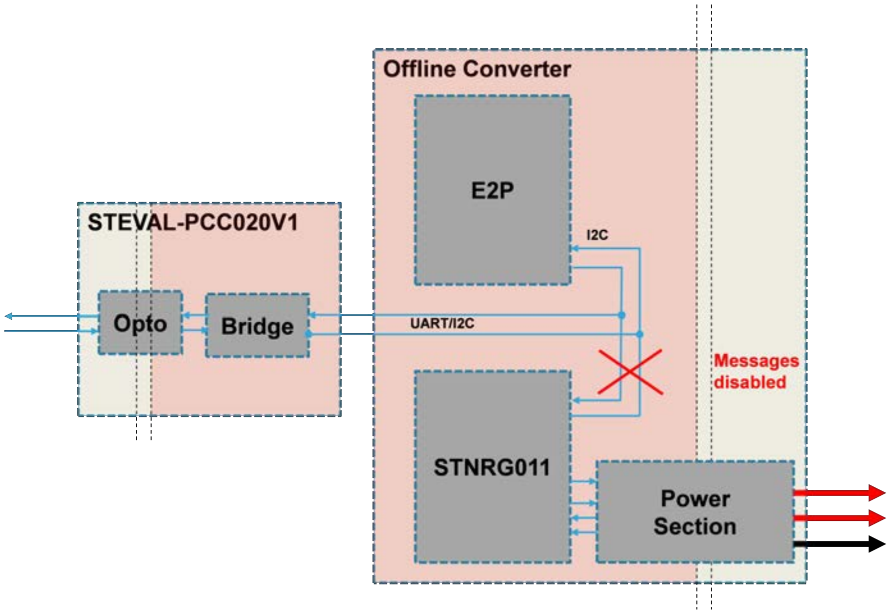
Step 4. Click the **Write to IC** button to program the NVM memory.

10 E²P operations

Important:
STNRG011 shares the E²P interface (SDA/SCL) with the UART interface to minimize pin count. During normal operation (switching), the optional E²P is only accessed at boot and when a fault occurs.

It is possible to access E²P in normal mode, but this might cause conflicts due to simultaneous access by the STNRG011 metering information (UART) and the GUI accessing the E²P.

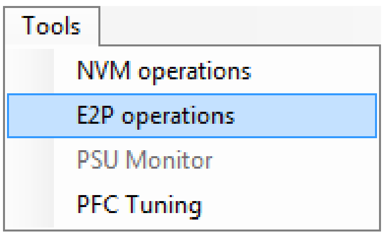
Figure 29. E²P simultaneous access conflicts on UART/I²C link



As a consequence, STNRG011 metering messages must be disabled, but this is only possible if UART uplink communication is enabled. Otherwise, E²P must be accessed while STNRG is disabled (e.g. no mains or external VCC required thanks to the internal VCC generation in ATE mode).

E²P operations are accessible via the **Tools** menu.

Figure 30. E²P operation menu



10.1 E²P dump

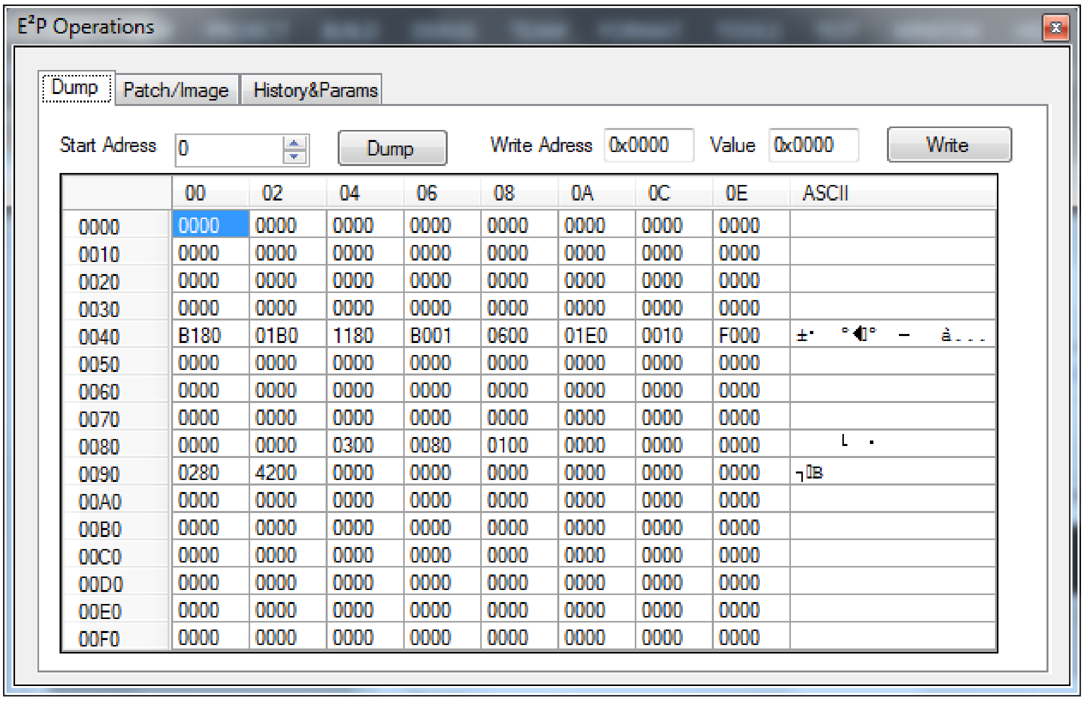
This feature allows displaying the content of the external E²P and is mainly used for debugging.

It is also possible to change a memory value by either clicking the address to be changed or by pressing the **Write** button in the **Address** and **Value** boxes.

Table 5. E²P mapping

Area	Meaning
0x0000-0x0022	Serial number and calibration data
0x0040-0x0047	Event history data
0x0048-0x004F	Running time, error counter, power on/off cycle counter
0x0080-0x0089	Cold/hot patch addresses definition
0x0090-0x0A80	Patch area

Figure 31. E²P Dump tab



10.2 E²P patch and image upload/download

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

10.2.1 Full E²P image operation box

This tab button allows:

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

Note: If an E²P is connected, the STNRG011 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.

10.2.2 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

Only the cold patching is used here.

Note: Normally, 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 & E2P Address.

To program a patch:

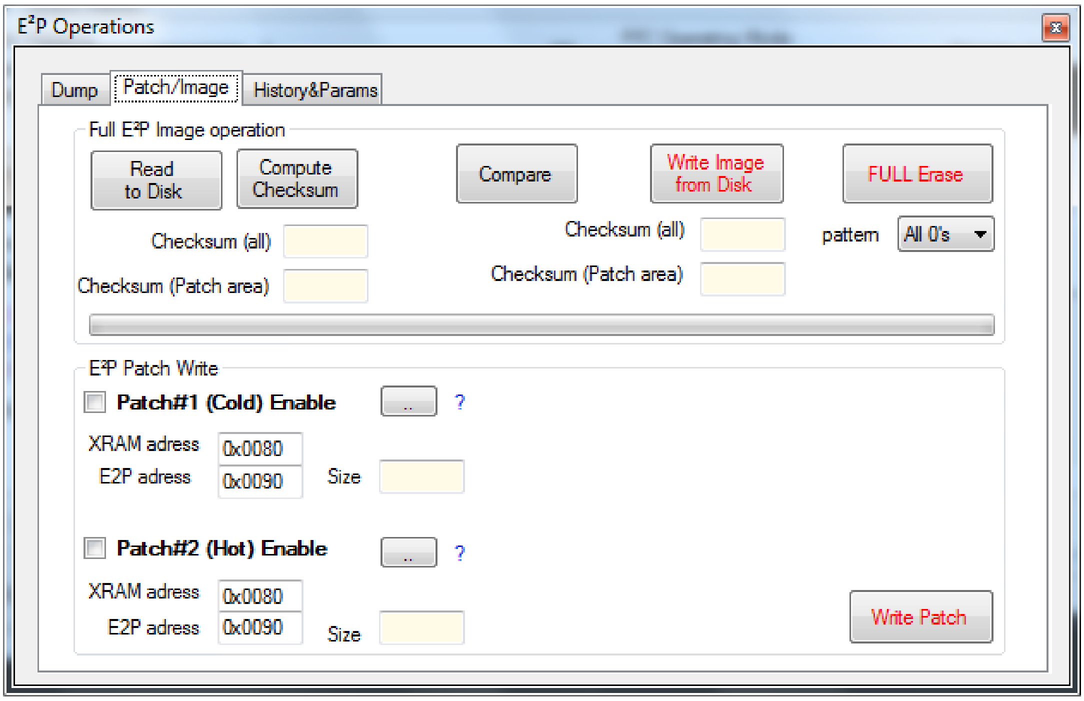
Procedure

Step 1. Click on the **..** button to select the patch to be used.
Only .bin format is supported.

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

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

Figure 32. E²P full import/export and patch operation tab



10.3 E²P parameter editor

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

Procedure

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 default values
If you want to write the values to E²P, you have to press the **Write** button.

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

Important:
It is recommended to edit these field with STNRG011 in ATE mode or via VCC externally powered.

If the parameters are written whilst STNRG011 is running, they will be overwritten when STNRG011 is shut down.

When STNRG011 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 will be overwritten in the E²P, hence the E²P content will be overwritten.

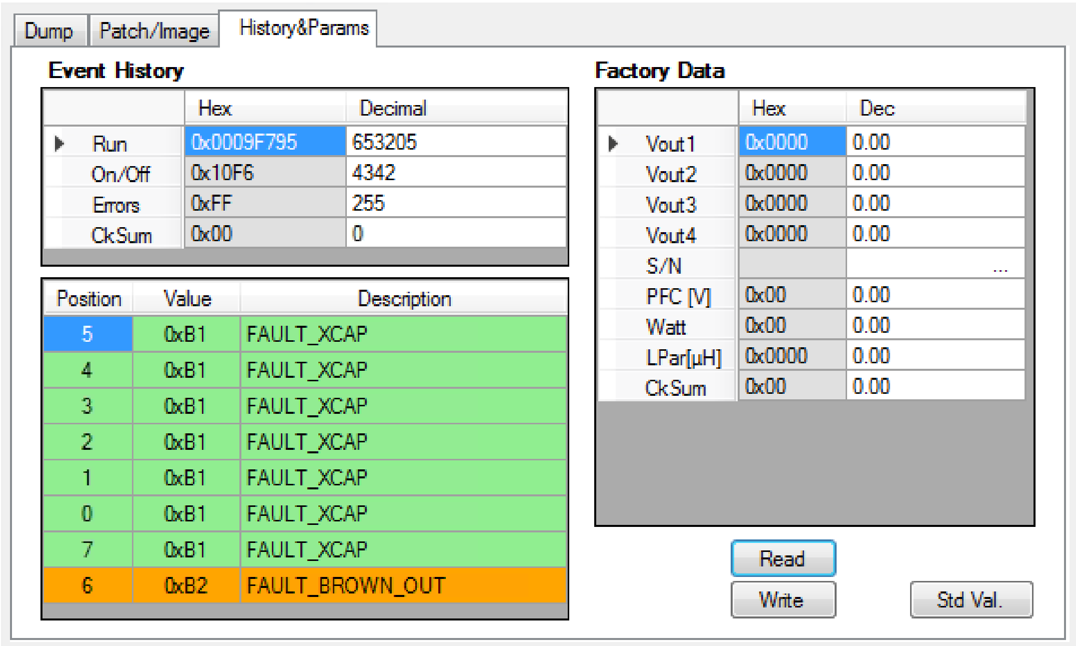


Figure 33. E²P parameter editor tab

10.4 Additional tools

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

10.4.1 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 STNRG011).

In this case, the GUI reads the STNRG011 memory directly and additional information can be retrieved.

Figure 34. **PSU monitor window**

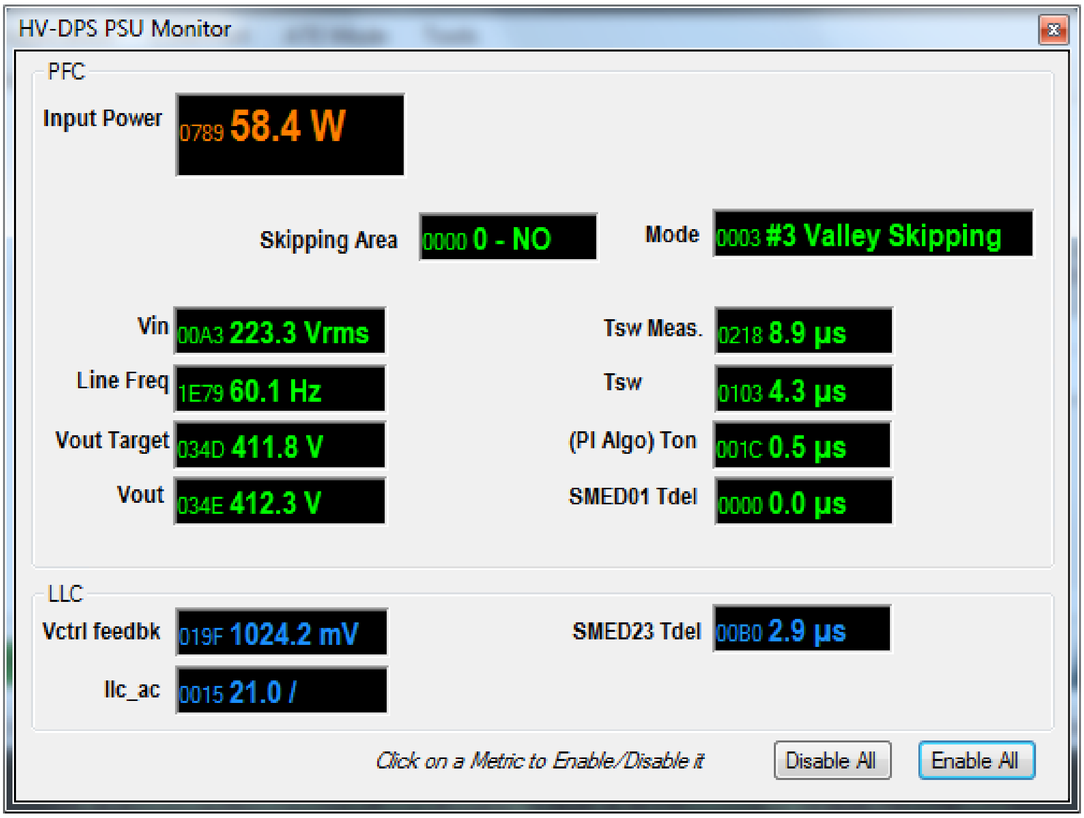


Table 6. **PSU monitor: PFC/LLC features**

PFC	
Input Power	Estimated PFC power corrected by Vin/PAM/Mode/Tdel parameters
Vin	Mains voltage in RMS value
Line Freq	Mains Frequency
Vout Target	PFC output Voltage (target)
Vout	PFC output Voltage (measured)
Tsw Meas	
PI Algo Ton	Ton added computed by power integration algorithm
Smed01 Tdel	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 optocoupler, used for the LLC feedback loop
llc_ac	LLC anti-capacitive mode indicator
SMED23 Tdel	Value of the LLC TimeShift

11 PFC calibration

11.1 Introduction

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

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

Note: *It is recommended to use the automated mode (GPIB adapter is required). The embedded driver only supports Chroma equipment. The SCPI commands issued are very generic, so it should work using other tools from other manufacturers (e.g., Agilent/Keysight) but this is not guaranteed.*

Otherwise, it is possible to use the manual mode (which is semi automated - in this case, the GUI detects some events and tries to minimize user actions on the tools).

11.2 Principle

The calibration scope is to correctly manage:

- 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 ValleySkipping and TM (high power).

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

Table 7. **PFC parameters to be calibrated**

Parameter name	Typ. value	Description
PFC THD improver base	5 (10 mV)	Base current of THD improver (ReCOT functionality on PFC_CS pin).
PFC THD improver gain	0 (no gain)	Gain of 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 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.

11.3 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.

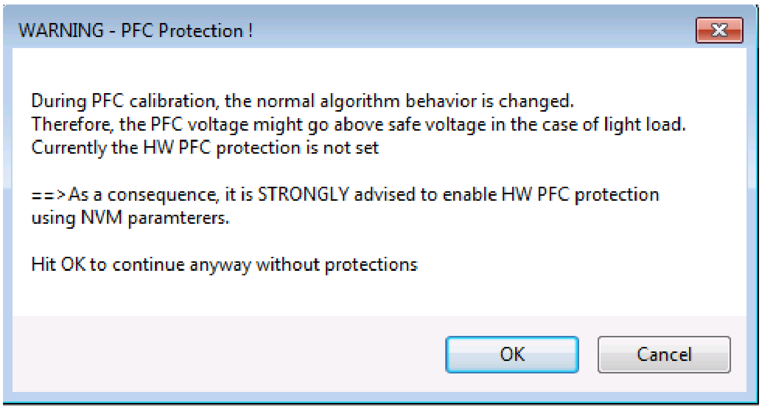
Danger:

Consequently, the PFC bulk voltage can rise above the nominal voltage at light load, leading to component (e.g. 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 NVM accordingly.

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

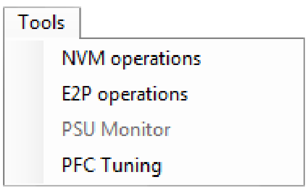
Figure 35. **PFC protection warning window**



11.4 Step by step calibration example

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

Figure 36. **PFC tuning menu**



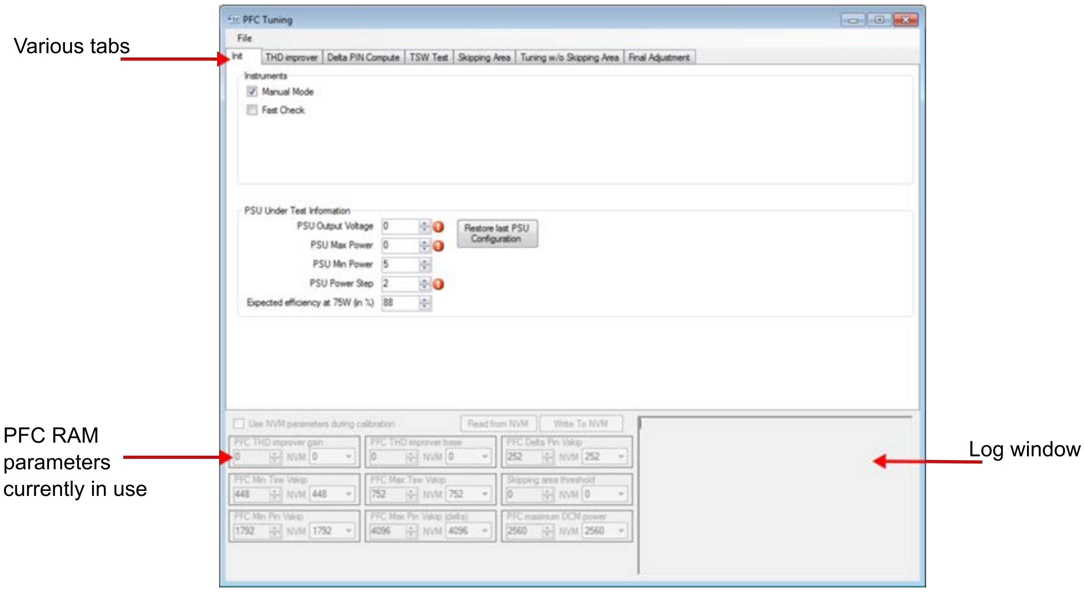
Note: To perform the calibration, the GUI has to communicate with the STNRG011. Since the UART communication is disabled by default in running mode, a special procedure (similar to the ATE mode) is applied to enable UART communication, which requires switching the power supply on/off, as prompted by the GUI.

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

11.4.1 Tuning calibration tabs

The following picture shows the PFC tuning main window with tabs representing the various steps to be followed during the calibration.

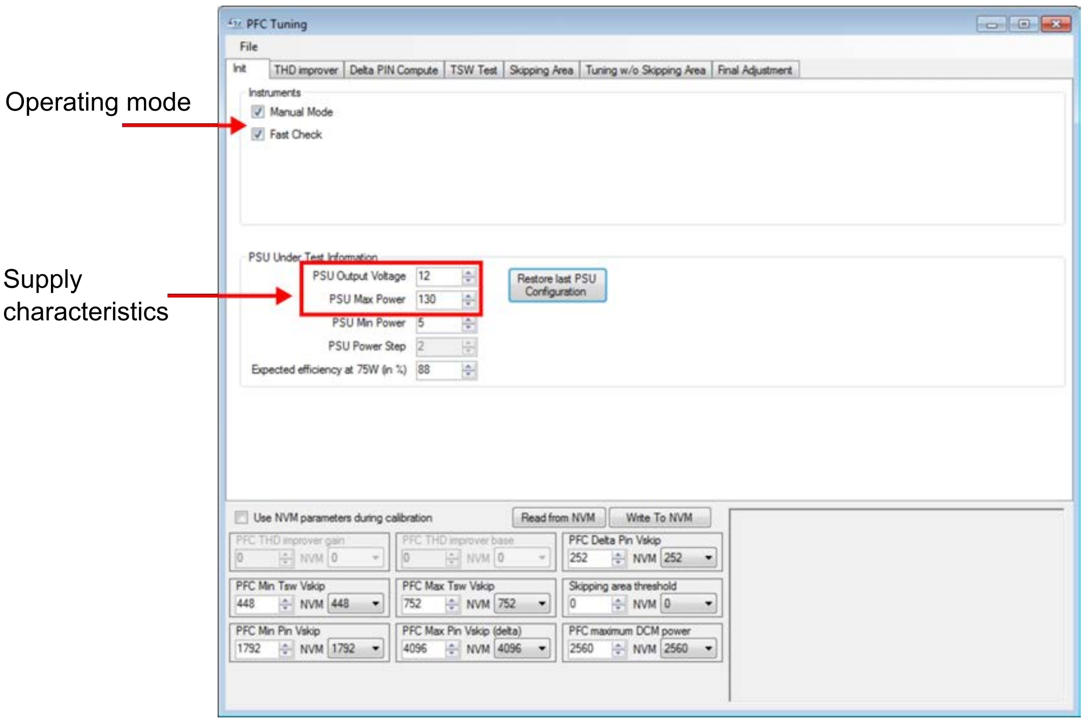
Figure 37. Main PFC tuning tab



11.4.2 Mode of operation

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

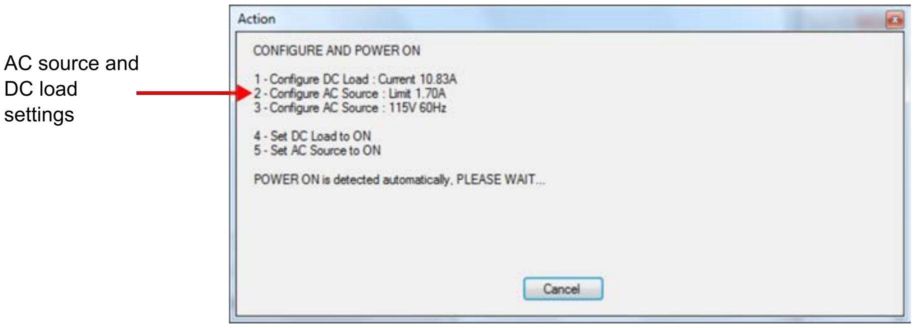
Figure 38. PFC tuning parameter tab



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

The STNRG011 is able to detect when the AC source is switched on/off, which limits the interactions.

Figure 39. Manual mode: AC source voltage and DC load current selection



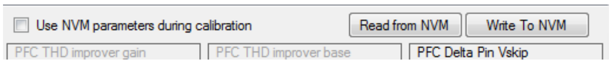
See Figure 40. NVM parameters: RAM and NVM values for NVM options.

As NVM storage space is limited, some PFC parameters have to be uploaded with a lower resolution than NVM values. To make sure 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 40. NVM parameters: RAM and NVM values



11.4.3 Step 1: THD improver

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

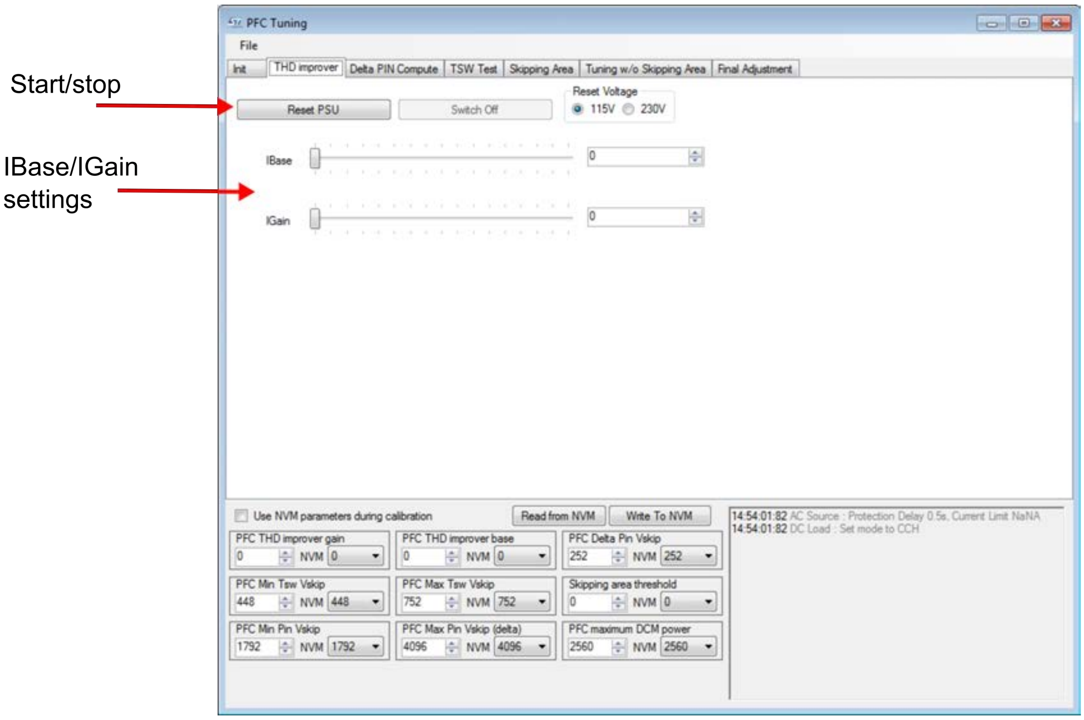
This feature helps to improve the AC current total harmonic distortion (THD) and the power factor (PF) value.

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

Once PSU has started, 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 modes of operation.

Figure 41. **PFC tuning: THD improver tab**



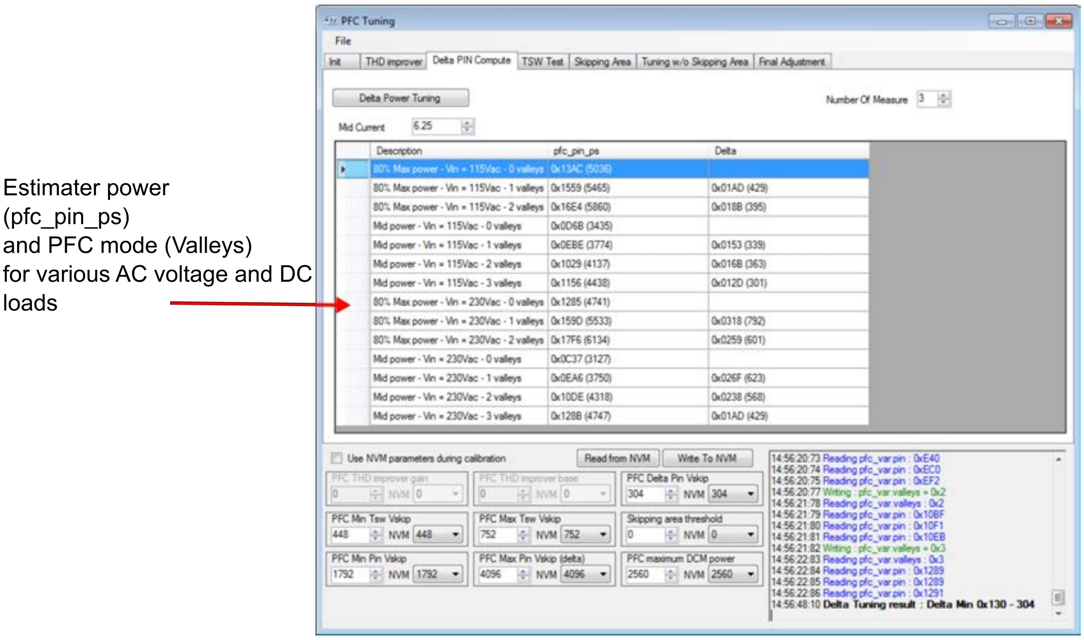
11.4.4 Step 2: delta pin calibration

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

To compute the delta pin calibration, the GUI:

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

Figure 42. **PFC tuning: delta pin calibration**



11.4.5 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 manually select the minimum/maximum bounds for the PFC frequency.

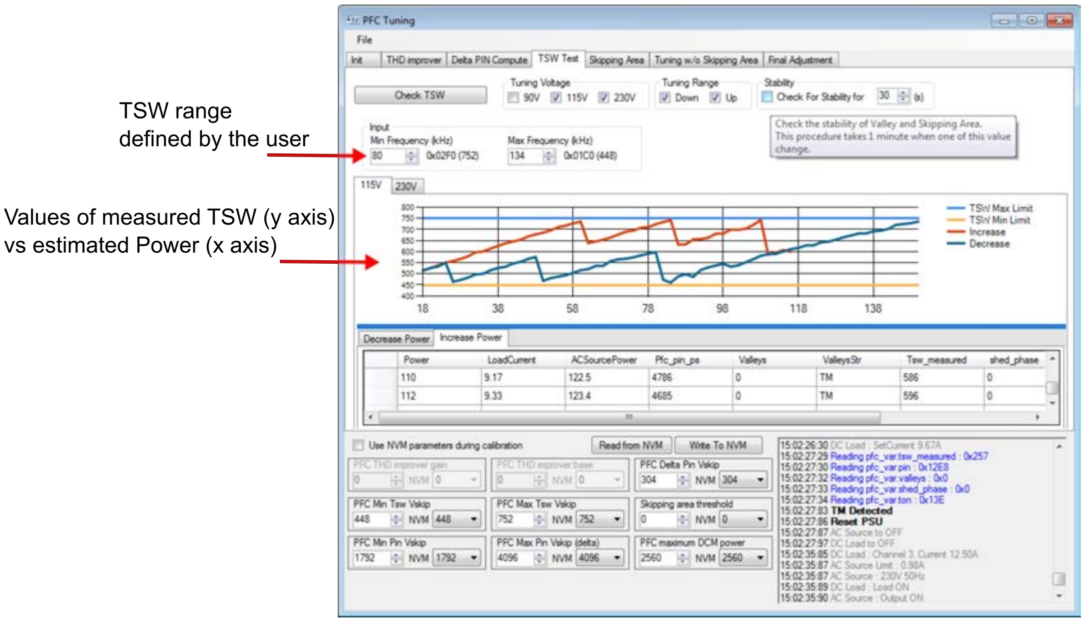
During normal operations, if the PFC actual frequency crosses this boundary, a mode change is required.

To check the **Tsw range**, in the GUI:

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

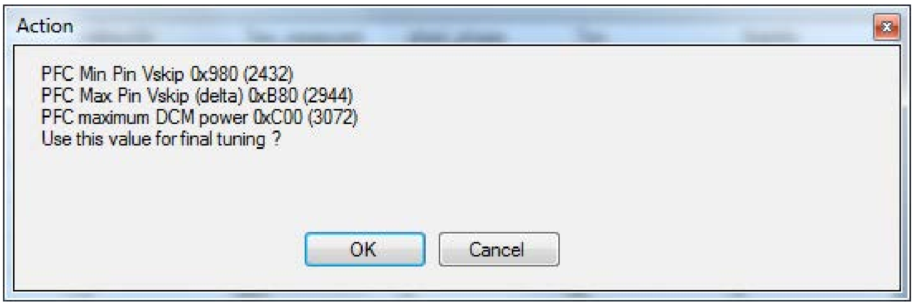
Note: If manual mode is selected, the GUI asks to smoothly ramp-up and down the DC load.

Figure 43. **PFC tuning: determining TSW limits**



Finally, the GUI computes the PFC parameters.

Figure 44. **PFC tuning: PFC min./max. pin and DCM power**



11.4.6 Step 4: skipping area threshold

The forth step consists in setting the skipping area power threshold (below this threshold, the system will apply power skipping, i.e. shutdown PFC operation before/after the peak of 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.

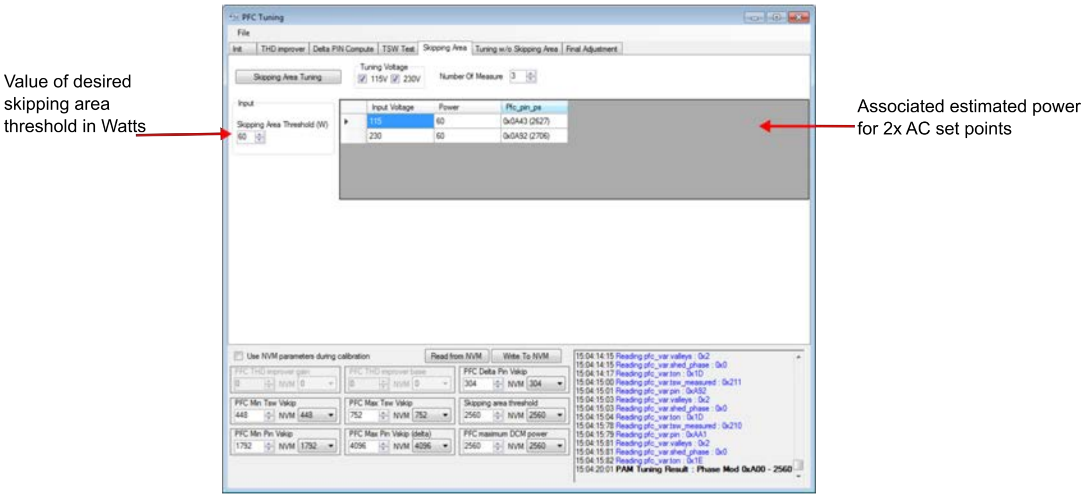
The threshold is up to the customer.

To determine the skipping area threshold, the GUI:

- enables PFC mode change (both 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 45. **PFC tuning: determining skipping area threshold**



11.4.7 Step 5: final verification

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

To perform the final verification, the GUI:

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

Figure 46. **PFC tuning: final verification without skipping area tab**

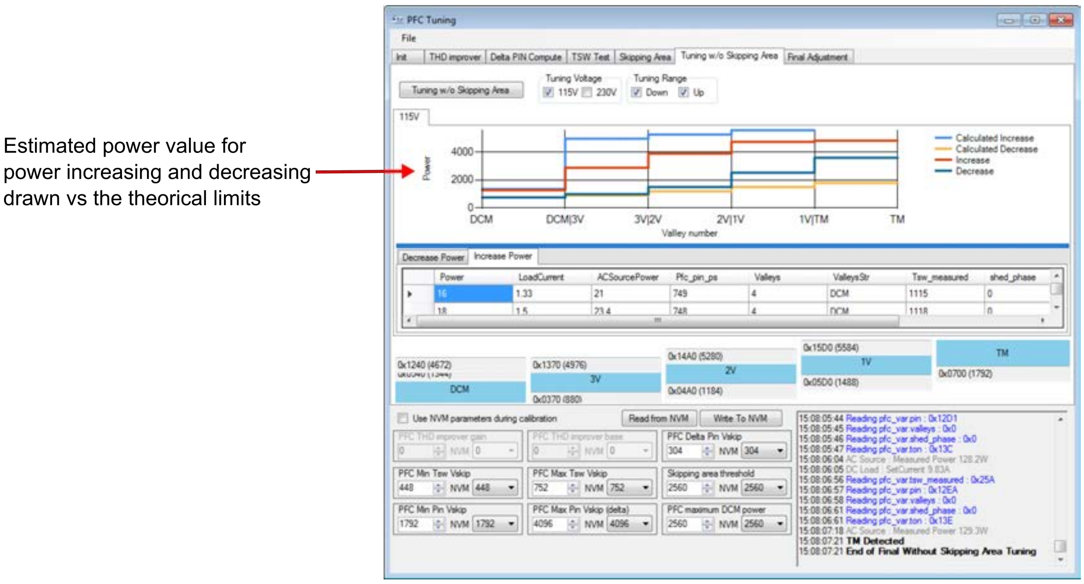
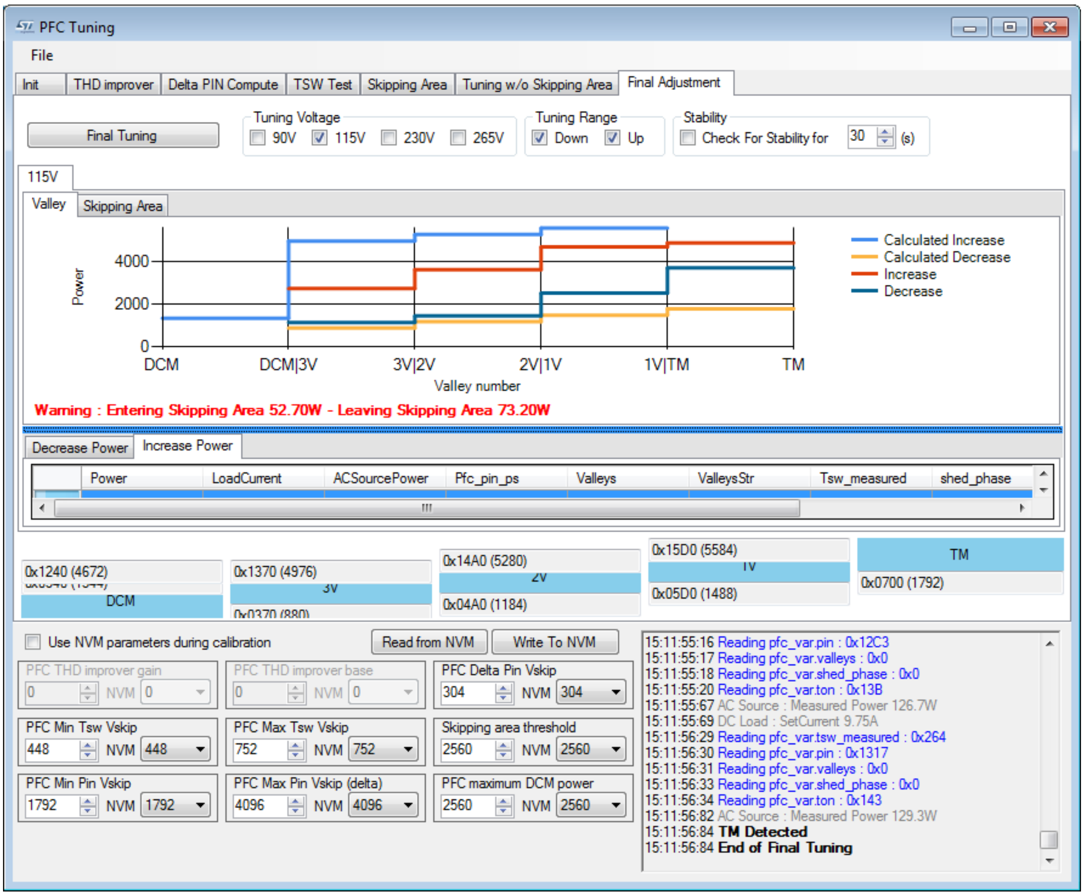


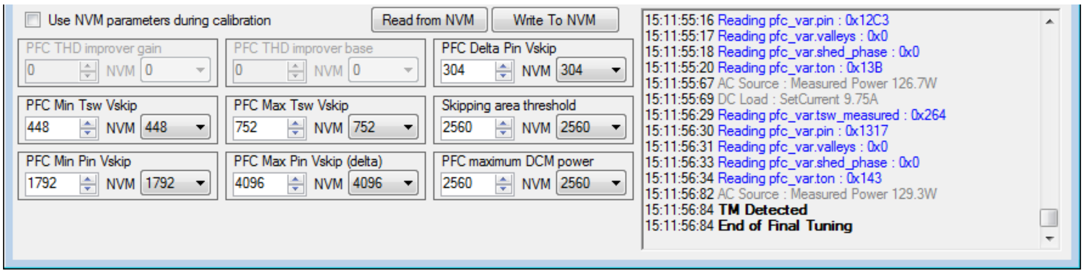
Figure 47. **PFC tuning: final verification**



11.4.8 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 the **Write to NVM** button.

Figure 48. **PFC tuning: writing NVM values**



12 Troubleshooting

12.1 No LED activity detected on the STEVAL-PCC020V1 interface board

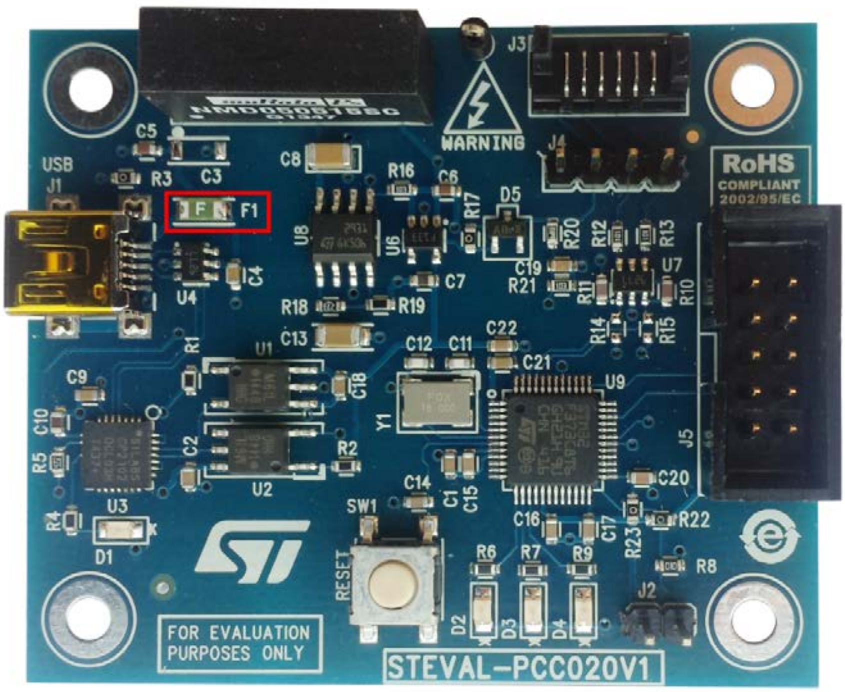
When the board is plugged to the system:

- LED D2 lights up, indicating the MCU is working, hence, the power supply is present;
- the yellow LED D1 starts blinking, indicating the USB port has been enumerated correctly.

If the LEDs are not working properly, it might be due to a power supply issue.

Procedure **Step 1.** Locate fuse F1 (close to the USB connector)

Figure 49. STEVAL-PCC020V1 interface board fuse



Step 2. Check the voltage between J1 ground (shield) and the right side of the fuse.
If it is not 5 V ±10%, it means an overcurrent occurred and the fuse has blown.

Step 3. Replace the fuse (0.5 A), after trying to fin out the root cause.

12.2 USB yellow LED shutdown in few seconds

The yellow LED D1 is wired to the USB suspend signal from CP2102; that is, it only lights up when the USB port is not in USB suspend mode.

Note: *By default (for Windows 7 and 8), the system forces external devices to enter suspend mode to save power (for example, when the COM port is not used). It does not mean the power supply is shut down, but the CP2102 goes into low power mode.*

To avoid this issue, select the SiLabs COM port, and go to the Power Management Tab and uncheck “Allow the Computer to turn off this device to save power”.

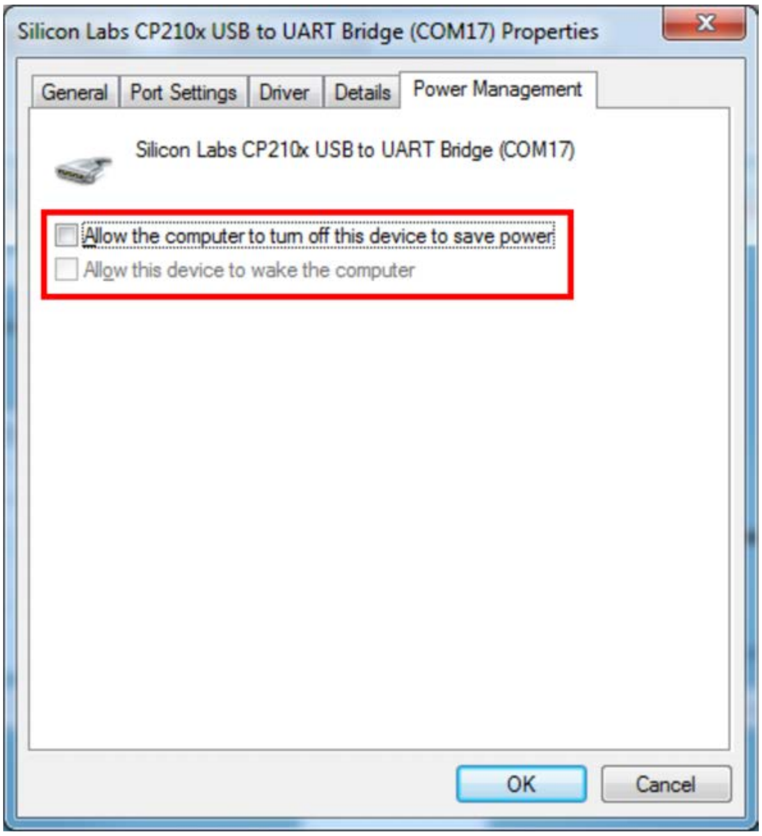
Procedure **Step 1.** go in the **Device Manager**

Step 2. select the SiLabs COM port

Step 3. go to the **Power Management** tab

Step 4. untick the **Allow the computer to turn off this device to save power** box

Figure 50. **Disabling CP2102 USB suspend mode**



13 STEVAL-PCC020V1 interface board hardware

13.1 Connector pinout

The connection between the supply and the interface board is made via a Molex 6-pin low profile connector (J3).

The signals are also available on a 4-pin HE10 header (J4).

Figure 51. STEVAL-PCC020V1 interface board signal connectors

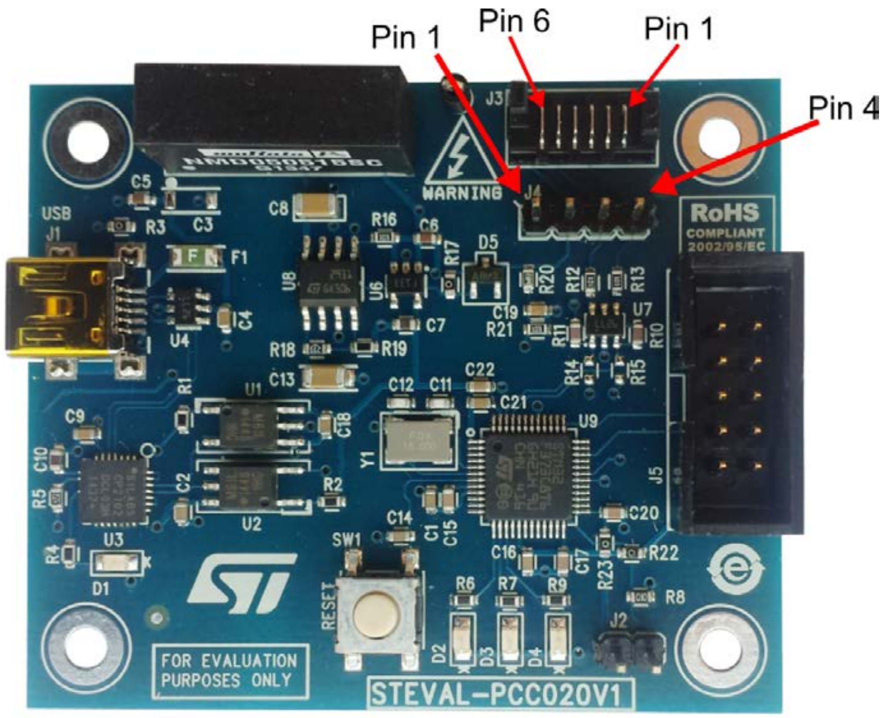


Table 8. STEVAL-PCC020V1 J3 and J4 pinout

Signal	4-pin header	6-pin Molex
GND	1	1
VCC	2	3
UART_TX / SDA	3	5
UART_RX / SCL	4	4
NC	none	2.6

13.2 Firmware upgrade

Figure 52. STEVAL-PCC020V1 interface board firmware update menu

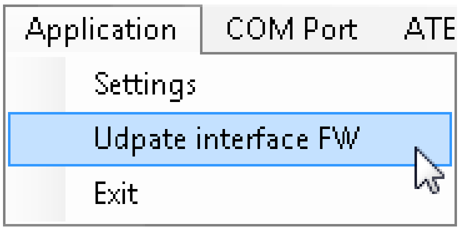


Figure 53. STEVAL-PCC020V1 interface board firmware update window

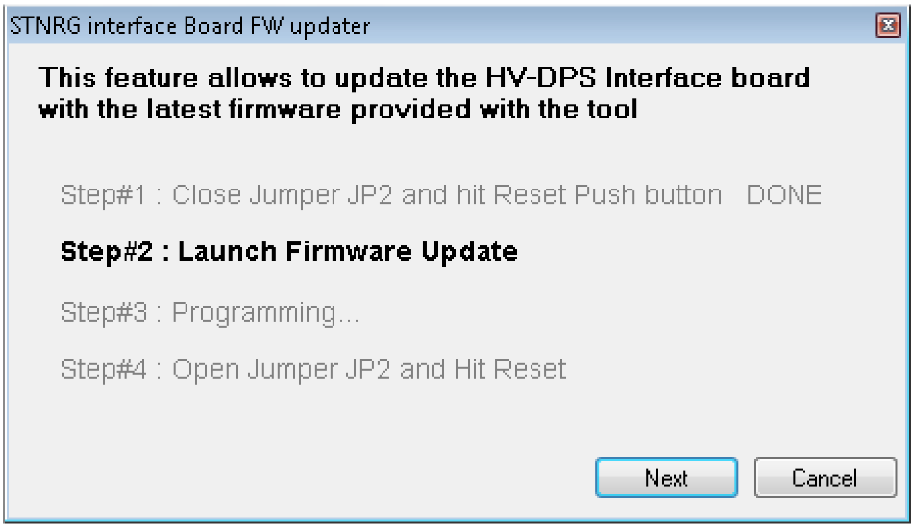
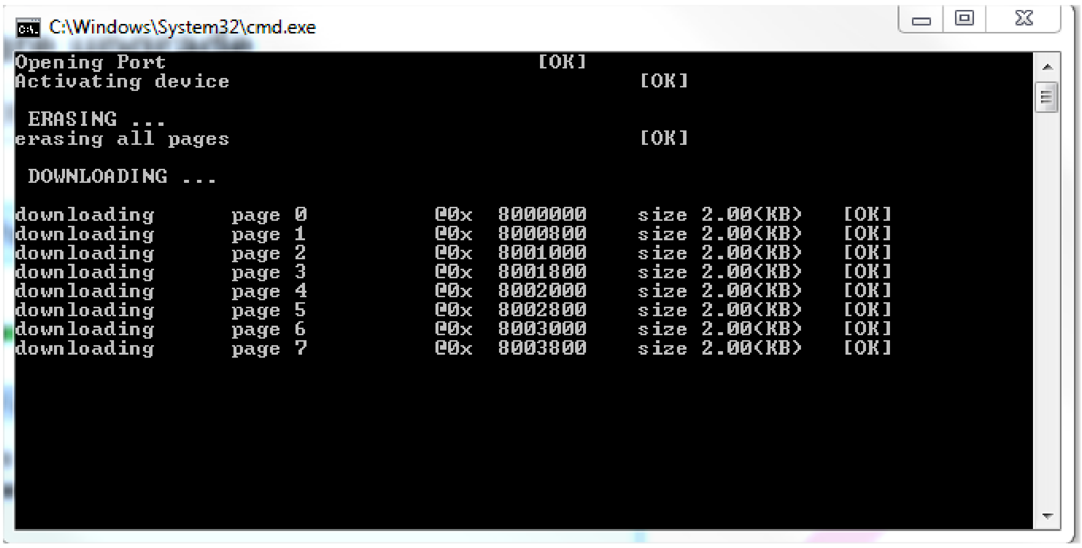


Figure 54. STEVAL-PCC020V1 interface board firmware update in progress



13.3 Schematic diagram

Figure 55. STEVAL-PCC020V1 circuit schematic: USB power

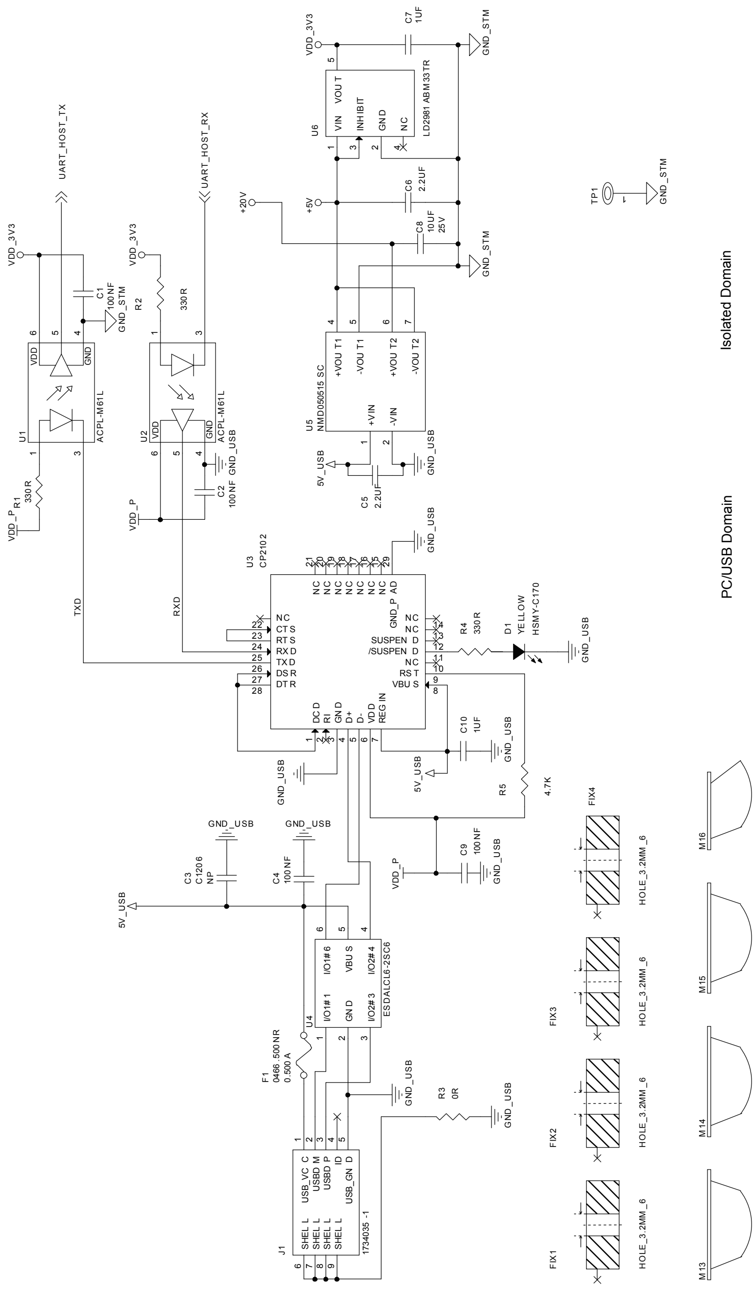
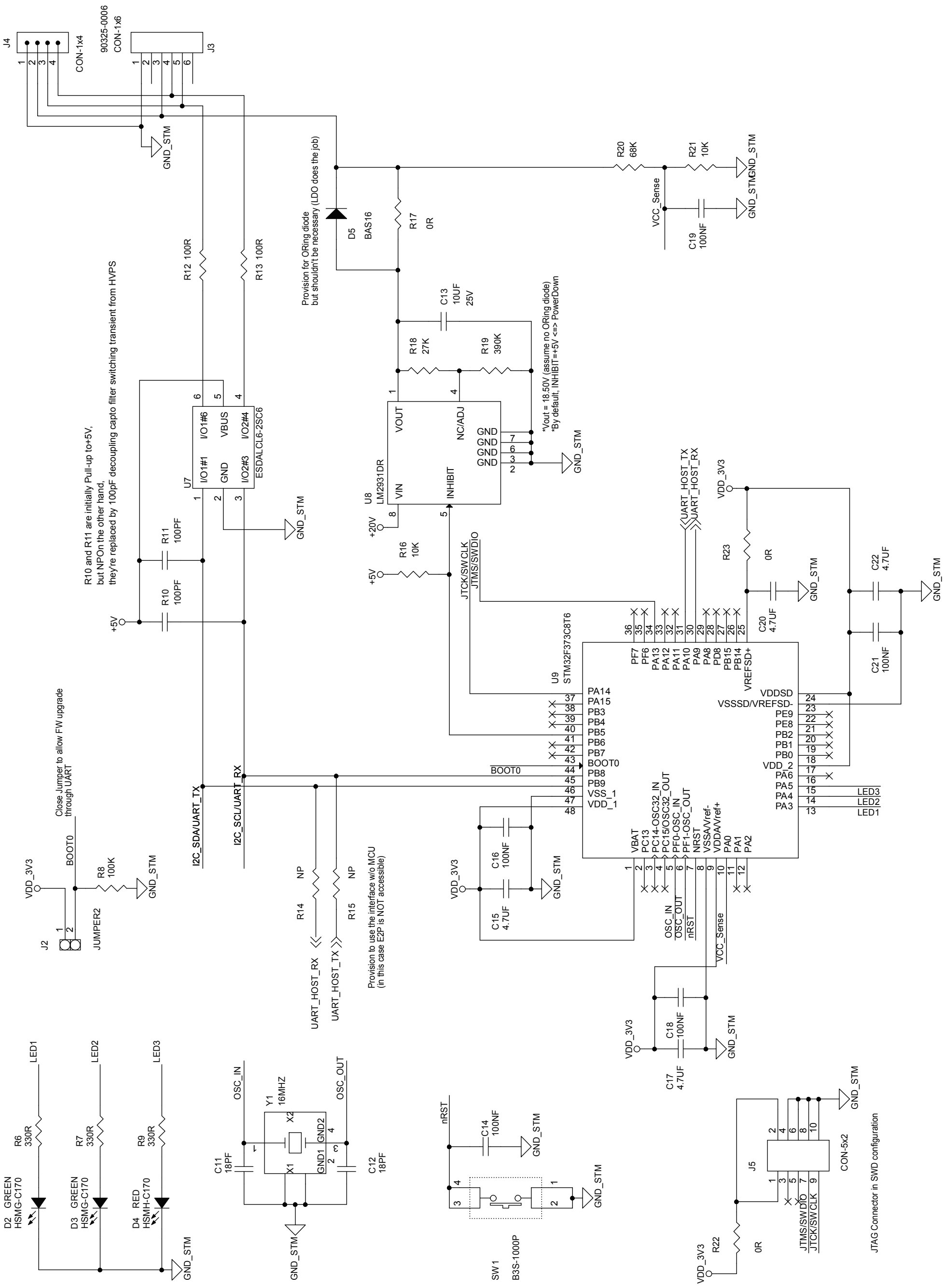


Figure 56. STEVAL-PCC020V1 circuit schematic: MCU



13.4 Layout

Figure 57. **STEVAL-PCC020V1 interface board layout (top view)**

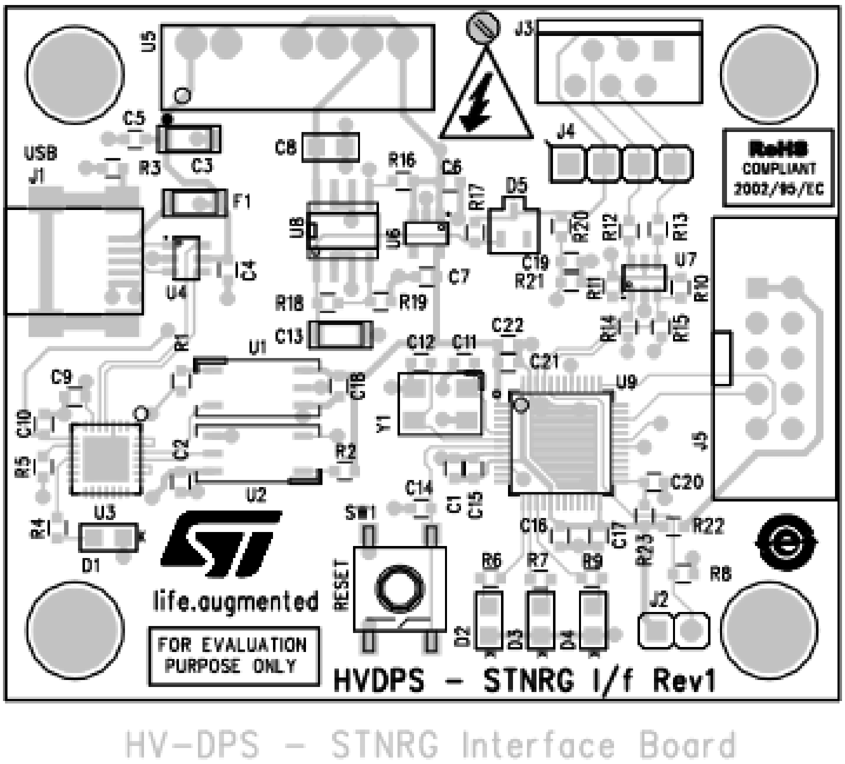
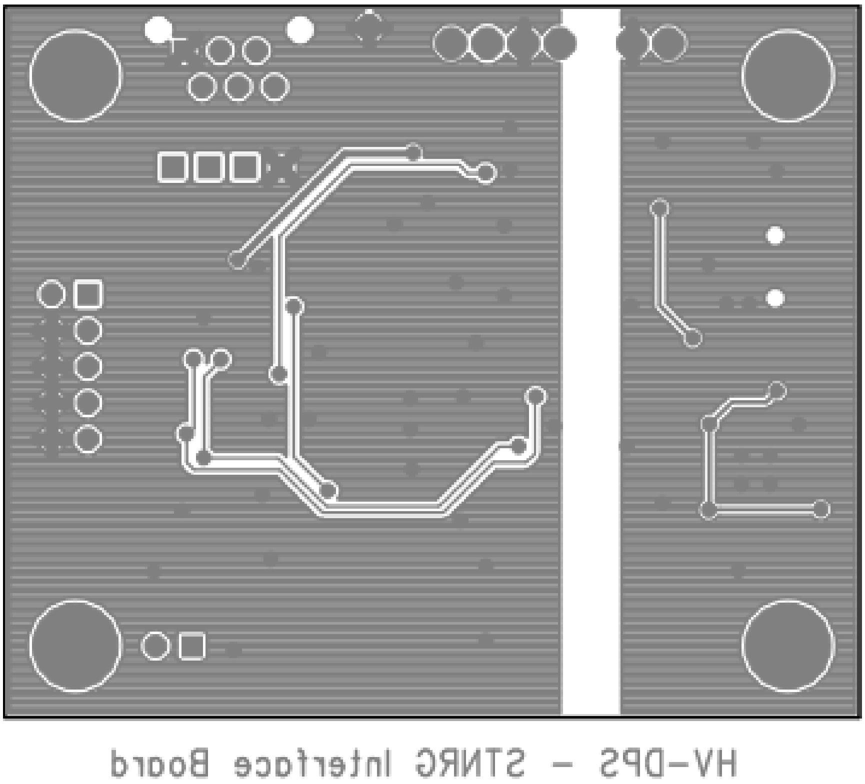


Figure 58. **STEVAL-PCC020V1 interface board layout (bottom view)**



13.5 Bill of materials

Table 9. **STEVAL-PCC020V1 bill of materials**

Item	Q.ty	Ref.	Part/Value	Description	Manufacturer	Order code
1	9	C1, C2, C4, C9, C14, C16, C18, C19, C21	100 nF, 50 V, 0603, X7R	Ceramic capacitors	Murata	GRM188R71H104KA93D
2	1	C3	NP 1206	Ceramic capacitor	Any	
3	2	C5, C6	2.2 µF, 6.3 V, X5R, 0603	Ceramic capacitors	TDK	C1608X5R0J225M080AB
4	2	C7, C10	1 µF, 16 V, X5R 0603	Ceramic capacitors	TAIYO YUDEN	EMK107BJ105KA-T
5	2	C8, C13	10 µF, 25 V, X5R, 1206	Ceramic capacitors	Murata	GRM31CR61E106KA12L
6	2	C11, C12	18 pF, 50 V, C0G, 0603	Ceramic capacitors	Any	
7	4	C15, C17, C20, C22	4.7 µF, 6V3, X5R, 0603	Ceramic capacitors	Any	
8	1	D1	YELLOW LED-0805	LED diode	Avago	HSMY-C170
9	2	D2, D3	GREEN LED-0805	LED diode	Avago	HSMG-C170
10	1	D4	RED LED-0805	LED diode	Avago	HSMH-C170

Item	Q.ty	Ref.	Part/Value	Description	Manufacturer	Order code
11	1	D5	85 V, 0.5 A, 4 nS, BAS16, SOT23	Diode	NXP Semiconductors	BAS16 /T3
12	1	F1	0.5 A, 63 V, SLO, SMD, 1206	Fuse	Littelfuse	0466.500NR
13	4	FIX1, FIX2, FIX3, FIX4	HOLE_3.2MM_6	Through hole	Any	
14	1	J1	CN-USB 1734035-1	USB_B_MINI_AMP_1734035-1	TE Connectivity	1734035-1
15	1	J2	STRIP254P-M-2 22-28-4023	Jumper	Molex	22-28-4023
16	1	J3	CON-1x6	PF-50 Header, 6 ways	Molex	90325-0006
17	1	J4	CON-1x4 1x4 - pitch 2.54	Through hole	Samtec	TSW-104-07-L-S
18	1	J5	CON-5x2	Connector	Molex	70246-1002
19	4	M13, M14, M15, M16	BUMPER SJ-5003 (BLACK)	Hemispherical bumper	3M	SJ-5003 (BLACK)
20	6	R1, R2, R4, R6, R7, R9	330 R, ±5%, 1/10 W, 0603	Resistors	Any	
21	4	R3 R17 R22 R23	0 R, 0 Ω, 1/10 W, 0603	Resistors	Yageo	RC0603JR-070RL
22	1	R5	4.7 K, ±5%, 1/10 W, SMD, 0603	Resistor	Any	
23	1	R8	100 K, 1/10 W, SMD, 0603	Resistor	Any	
24	2	R10, R11	100 pF, 50 V, COG, 0603	Resistors	Murata	GRM1885C1H101JA01D
25	2	R12, R13	100 R , ±5%, 1/10 W, 0603	Resistors	Panasonic Corp.	ERJ-3GEYJ101V
26	2	R14, R15	NP 0603	Resistors	Any	
27	2	R16, R21	10 K , , ±5%, 1/10 W, 0603	Resistors	Panasonic Corp.	ERJ-3GEYJ103V
28	1	R18	27 K, ±5%, 1/10 W, 0603	Resistor	Any	
29	1	R19	390 K , ±1%, 0603	Resistor	Panasonic Corp.	
30	1	R20	68 K, ±5%, 1/10 W, SMD, 0603	Resistor	Panasonic Corp.	ERJ3GEYJ683V
31	1	SW1	B3S-1000P B3S-1000P	Tactile switch	Omron	B3S-1000P
32	1	TP1	TP	Test point	Mouser	534-5001
33	2	U1, U2	ACPL-M61L SO-5	Optocoupler	Avago	ACPL-M61L-000E
34	1	U3	CP2102 QFN-28	Uart over USB bridge, QFN28	Silicon Labs	CP2102-GM
35	2	U4, U7	ESDALCL6-2SC6 SOT23-6L	Very low capacitance and low leakage current ESD protection	ST	ESDALCL6-2SC6
36	1	U5	NMD050515SC NMD_SIP	Isolated dual output DC-DC	Murata	NMD050515SC

Item	Q.ty	Ref.	Part/Value	Description	Manufacturer	Order code
37	1	U6	3.3 V, 100 mAh, LD2981ABM33TR SOT23-5	Ultra low drop voltage regulators with inhibit low ESR output	ST	LD2981ABM33TR
38	1	U8	100 mAh, LM2931D-R SO8	Very low drop voltage regulators with inhibit function	ST	LM2931D-R
39	1	U9	STM32F373C8T6 LQFP48	Mainstream mixed signals MCUs ARM Cortex-M4 core	ST	STM32F373C8T6
40	1	Y1	16 MHZ FQ5032B	XTAL	Fox Electronics	FQ5032B-16.000
41	1	Cable	6 circuits, 0.20m length	Picoflex® PF-50 IDT-to-IDT	Molex	92315-0620

Revision history

Table 10. **Document revision history**

Date	Version	Changes
01-Feb-2018	1	Initial release.

Contents

List of tables

List of figures