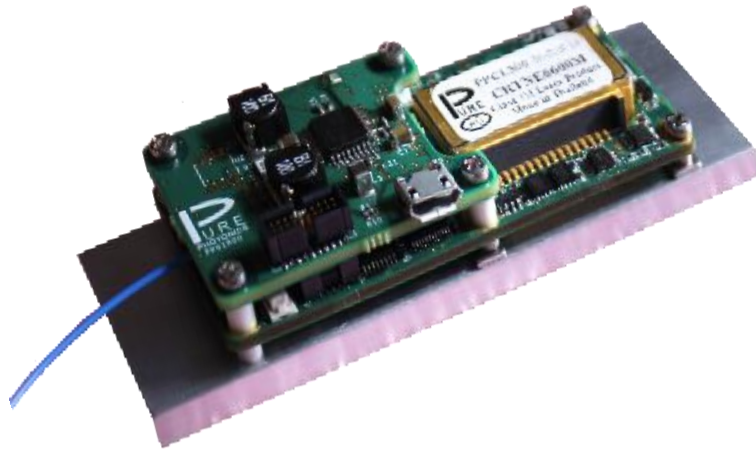# Low Noise Tunable Laser

Clean Light portfolio – Feature Guide

The **Pure Photonics** full-band tunable laser solution provides a very narrow linewidth (~10 kHz), significantly reduced low-frequency AM and FM noise and a range of operating modes in this low-noise setting. The product can access any desired frequency set-point in the tuning range (C-band, L-band or optionally up to 60nm). Output power can be set as low as 7dBm and as high as 18dBm.

This feature guide provides more detail about the features available on this product and how to use them, including interactions with the Graphical User Interface, Command Line Interface and the base RS-232 interface.

We recommend to use this guide as a starting point, with the need to use our more detailed implementation guides and application notes (mostly available on our website) for a full understanding.

# 1. Table of Contents

## 2. Features by Module Family

The features available on each model is listed in the below table. The 'S' designates a standard feature (available on all units) and the 'O' designates an optional feature.

| | PPCL100 | PPCL200 | PPCL300 |
|---|---|---|---|
| **Tuning range [1]** | S 38nm<br>O 60nm | S 38nm<br>O 60nm | S 38nm<br>O 60nm |
| **Maximum Power [1]** | S 13.5dBm<br>O 18dBm (C-band)<br>O 17dBm (L-band) | S 13.5dBm<br>O 18dBm (C-band)<br>O 17dBm (L-band) | S 13.5dBm<br>O 18dBm (C-band)<br>O 17dBm (L-band) |
| **In-operation power adjustment** | S (7dBm-MAX) | S (7dBm-MAX) | S (7dBm-MAX) |
| **In-operation frequency adjustment** | S (12 GHz) | S (30GHz)<br>O (100GHz) | S (30GHz)<br>O (100GHz) |
| **Fiber** | PMF /w short or long tubing | PMF /w short or long tubing | PMF /w short or long tubing |
| **Clean Sweep** | | Y | Y |
| **Clean Sweep Extended** | | O | O |
| **Clean Scan** | | O | O |
| **No Drift** | | O | O |
| **Low RIN configuration** | | O | O |
| **Clean Modulation FM** | | | O |
| **Clean Modulation AM** | | | O |
| **Clean Measurement** | | | O (1 or 2 channels) |
| **Enclosure [2]** | PPCL500 | PPCL500 | PPCL550 |
| **Enclosure fiber** | Adapter or pigtail (900um/2mm/3mm) | Adapter or pigtail (900um/2mm/3mm) | Adapter or pigtail (900um/2mm/3mm) |

Note 1: the customer can choose their desired tuning range (up to 60nm) in the wavelength ranges 1515nm – 1580nm (extended C-band) and 1560nm – 1625nm (extended L-band). Maximum guaranteed power in the tuning range (subject to the selected power level) is as below:

| **Max Guaranteed Output Power** | 1515-1520 | 1520-1525 | 1525-1570 | 1570-1575 | 1575-1580 |
|---|---|---|---|---|---|
| **Extended C-band** | 16.5dBm | 17.25 dBm | 18 dBm | 17.25 dBm | 16.5 dBm |
| | 1560-1565 | 1565-1570 | 1570-1615 | 1615-1620 | 1620-1625 |
| **Extended L-band** | 15.5 dBm | 16.25 dBm | 17 dBm | 16.25 dBm | 15.5 dBm |

E.g when a customer selects 18dBm for 1515-1575nm range, they are guaranteed at least 16.5dBm at 1515nm. Power levels above that are best-effort.

Note 2: The enclosure solution integrates the ITLA (PPCL100/200/300) into a metal enclosure with power supplies and a convenient communications interface. The customer will need to plug in a barrel plug from a power supply (included) and a micro-USB cable (included). If analog inputs are included, SMA connectors are added (PPCL550 only). The customer can choose between a fiber adapter in the wall of the module or a fiber pigtail (900um buffer, 2mm cable or 3mm cable).

At times, special configurations may have been created for customers that deviate from the above descriptions. Customers that have purchased such units were made aware of the differences.

## 3. How to Use this guide

This guide provides a description for each feature, along with how to use it and its limitations. Then more details are provided on which Graphical User Interface (GUI) to use with this feature, what calibration files are required and how to communicate through the Command Line Interface (CLI) and standard RS-232. For the use of the CLI and RS-232, please refer to the PPCL200 operating guide.

The Graphical User Interface is a set of tools to demonstrate and control the different features. It uses interfaces that are completely accessible to the customer and hence any feature in the GUI can be fully implemented by the customer in their own system.

The core file in the GUI is the "PP GUI.exe" file. Some of the features are loaded as their own executable (for convenience), but any other feature can be accessed also by creating a shortcut to the "PP GUI.exe" file and then adding the appropriate keyword. Please see below image as an example for the basic GUI.
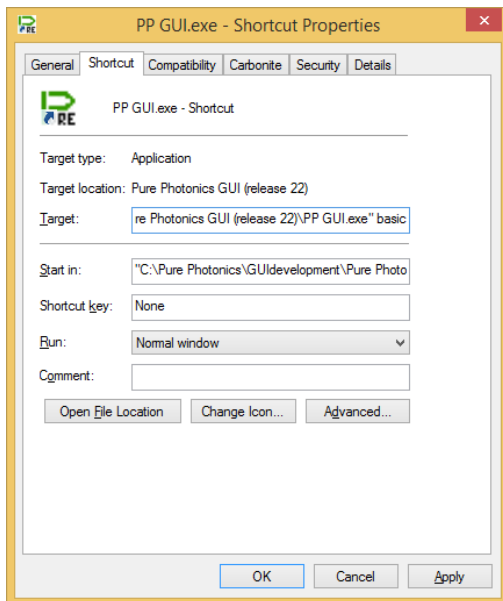
# 4. Configuring (micro) USB interfaces

The native communication with the module is through a low-voltage RS-232 interface. Several options exist where the customer-facing interface has been changed to a USB interface. For example, the PPCL300 comes with a micro-USB interface option, the CoBrite package comes with a standard USB interface and for computers without a COM port an RS-232 – USB dongle is used.

The standard USB interface is not optimized for serial communications on the RS-232 interface. Due to differences in protocol and optimization algorithms for the USB port, it installs with non-optimal settings. For most operations, this is not a problem, however for a firmware upgrade a proper configuration is required.

We recommend to make the below changes when the USB interface gets registered. On every computer this is only needed once for each USB interface/device.

a.  Open the 'Windows Control Manager'
b.  Open 'Hardware and Sound'
c.  Open 'Device Manager'
d.  Find the COM ports
e.  Right-click the USB serial port and select 'Properties'
f.  Select the tab 'Port Settings' and click the 'Advanced' button
g.  **Set the latency timer value to 1msec.**
    a. You can also select the COM-port designation here (we recommend port # <10).
    b. We recommend to set the 'USB transfer sizes' to the lowest possible setting (do this for both receive and transmit)
h.  Close the windows and start using the device
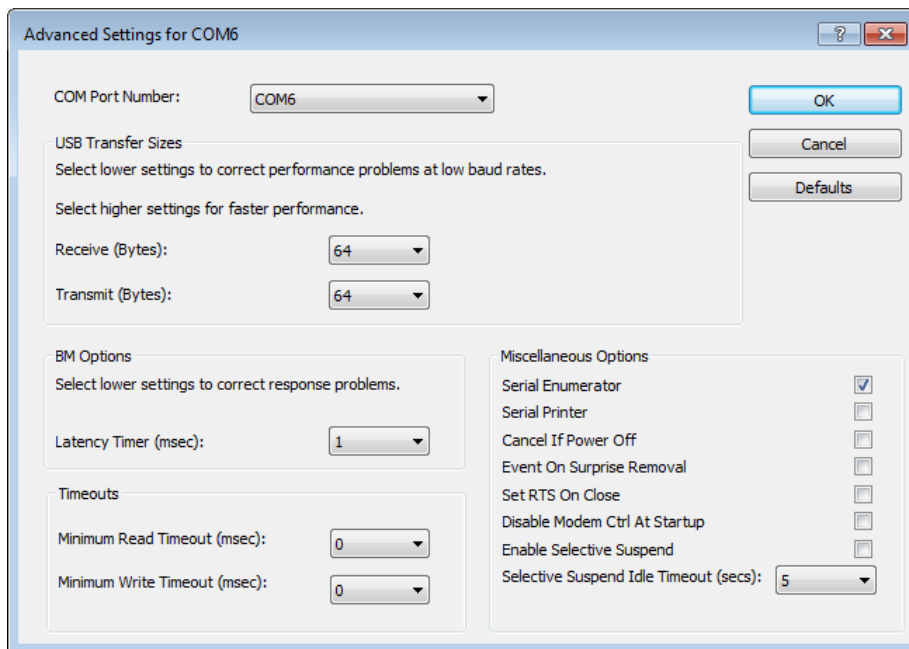
# 5. Electrical pin-out

The product comes configured either with a 14-pin female connector (PPCL100 and PPCL200; Samtec CLM-107-02-F-D) or a 20-pin male connector (PPCL300; Samtec FTMH-110-02-L-DV-ES). The first 14 pins of the 20-pin connector match the 14 pins on the 14-pin connector, defined by the OIF MSA document:

The 14-pin and 20-pin connector are defined as follows:

| | | | | |
|---|---|---|---|---|
| +3.3V supply | 1 | 2 | DIS* |
| | 3 | 4 | SRQ* |
| GND | 5 | 6 | MS* |
| | 7 | 8 | TxD (1) |
| -5.2V supply | 9 | 10 | RxD (1) |
| | 11 | 12 | RST* |
| Not used | 13 | 14 | DitherAA |
| PZT Noise Reduction | 15 | 16 | AM Modulation |
| Analog output | 17 | 18 | Clean Measurement 1 |
| Clean Measurement 2 | 19 | 20 | FM Modulation |

(1) not used when unit has micro-USB connector

Looking onto the top of the ITLA, with the goldbox on the right-side, the fiber going to the left and the connector closest to you. The pin 1 is the closest pin to the center of the ITLA. The top row increments as 1, 3, 5, etc. The bottom row (along the edge of the PCB) counts pins 2, 4, 6, etc.

# 6. Low-noise modes

| | |
|---|---|
| **Description** | The standard operating mode for the laser is the dither-mode. In this mode all the control loops are running and creating noise (especially in the 1-10,000Hz range). By disabling these control-loops, either completely or partially, a lower noise behavior can be achieved. |
| | In the whisper mode, essentially all control loops are disabled, to the extent possible. This results in the removal of the 888Hz tone (and its overtones) and this significantly reduced the AM and FM noise in the below 100Hz range. If in any way possible, we recommend to once in a while switch back to the dither mode for the laser to relock. Such a switch-back could be done in less than 10 seconds. |
| **GUI** | Basic GUI (keyword: 'basic') |
| **Calibration Files** | Not required. All calibration data is internal of the unit. |
| **Firmware Version** | 8.0.x, 8.1.x, 8.2.x (not on 8.3.x) |
| **CLI Commands** | *it.cleanMode(x)* to adjust the low-noise mode, with 0 the standard dither-mode and 2 the whisper-mode (a value of 1 defaults to 2). |
| **RS-232 Commands** | Register *0x90*: switch between the different low-noise modes, with 0 the standard dither-mode, 2 the whisper-mode (a value of 1 defaults to 2). |

# 7. In-operation Frequency and Power Adjust

**Description**            After reaching the target power and frequency, the user can adjust the power and frequency, while maintaining lasing operation in the same optical mode for a certain operating range. Beyond this range the laser would need to shut down and restart at the new set-point.

In whispermode the frequency can be adjusted (though we recommend small steps). The Power cannot be adjusted.

**GUI**                    Basic GUI (keyword: 'basic')

**Calibration Files**      Not required. All calibration data is internal of the unit.

**Firmware Version**       This feature is available on every firmware version

**CLI Commands**           *it.pwr(x)* to adjust the power setting, with x the target power in units of 0.01dBm

*it.ftf(x)* to adjust the frequency setting, with x the offset from the target frequency (FCF + GRID * (CHANNEL-1) ) in units of 1MHz

**RS-232 Commands**        Register *0x31*: adjust power in units of 0.01dBm

Register *0x62*: adjust frequency in units of MHz

# 8. Clean Sweep (standard)

| | |
|---|---|
| **Description** | Performs a controlled frequency ramp (up and down) over a chosen frequency range. The laser will initially ramp to (+0.5 * range) and then reverse the ramp down to (-0.5 * range). This ramp-up and ramp-down is repeated until the user instructs to stop. |
| **GUI** | Clean Sweep GUI (keyword: 'cleansweep') |
| **Calibration Files** | Not required. All calibration data is internal of the unit. |
| **Firmware Version** | 8.0.9 and above |
| **CLI Commands** | *it.cleanSweepAmplitude(x)* to set the Clean Sweep range, with x the range in units of GHz |
| | *it.cleanSweepEnable(x)* to activate (x=1) and de-activate (x=0) the Clean Sweep |
| | *it.cleanSweepOffset()* to read the frequency offset in units of 0.1 GHz (signed integer). |
| | *it.cleanSweepRate()* to set the maximum sweep rate in MHz/sec |
| **RS-232 Commands** | Register *0xe4*: set the Clean Sweep range in units of GHz |
| | Register *0xe5*: activate (1) and de-activate (0) the Clean Sweep |
| | Register *0xe6*: read the frequency offset in units of 0.1 GHz (signed integer). |
| | Register *0xf1*: Set the maximum sweep rate in units of MHz/sec |

# 9. Clean Sweep Extended

| | |
|---|---|
| **Description** | Similar to the standard Clean Sweep, this feature provides a controlled frequency ramp up and down. The extended sweep is used for sweep ranges larger than 150GHz, where additional calibrated values are required for proper operation. |
| **GUI** | Clean Sweep Extended GUI (keyword: 'CleanSweepExtended') |
| **Calibration Files** | .csmap file |
| **Firmware Version** | 8.0.9 and above |
| **CLI Commands** | *it.cleanSweepAmplitude(x)* to set the Clean Sweep range, with x the range in units of GHz |
| | *it.cleanSweepEnable(x)* to activate (x=1) and de-activate (x=0) the Clean Sweep |
| | *it.cleanSweepOffset()* to read the frequency offset in 0.1 GHz with an offset of 200GHz (offset = (value – 2000) * 0.1 GHz) |
| | *it.cleanSweepRate()* to read the frequency offset in units of 0.1 GHz (signed integer). |
| | *it.cleanSweepConstants()* to upload calibration constants. Please refer to the **implementation guide for Extended Clean Sweep.** |
| **RS-232 Commands** | Register *0xe4*: set the Clean Sweep range in units of GHz |
| | Register *0xe5*: activate (1) and de-activate (0) the Clean Sweep |
| | Register *0xe6*: read the frequency offset in units of 0.1 GHz (signed integer). |
| | Register *0xf1*: Set the maximum sweep rate in units of MHz/sec |
| | Register *0xf8*: calibration constants derived from the .csmap file to ensure proper operation at the extremes of the sweep. |
| | Please refer to the **implementation guide for Extended Clean Sweep.** |

# 10.  Clean Jump

| | |
|---|---|
| **Description** | Provides the ability to quickly jump from one frequency to another. Typically, with this laser type, to switch from one frequency to another will require disabling the laser and re-enabling it, resulting in 20-30 seconds switching time. With this feature an in-operation switch is performed (includes a modehop) which takes place is less than a second (0.1 seconds for specific frequency combinations). |
| **GUI** | Clean Jump GUI (keyword: 'cleanjump') |
| **Calibration Files** | .map file at the desired power level |
| **Firmware Version** | 8.1.x |
| **CLI Commands** | *it.cleanSweepOffset()* to read the frequency offset in MHz (signed integer) |
| | *it.cleanJumpEnable(1)* to activate (x=1) the Clean Jump |
| | *it.cleanJumpTHz(x)* to load the THz part of the next frequency, with x in units of THz |
| | *it.cleanJumpGHz(x)* to load the GHz part of the next frequency, with x in units of 0.1GHz |
| | *it.cleanJumpSled(x)* to load the sled temperature target of the next frequency, with x in units of 0.01C |
| | *it.cleanJumpCurrent(x)* to load the target current for the next frequency, with x in units of 0.1mA |
| **RS-232 Commands** | Register *0xe6*: to read the frequency offset in MHz (signed integer) |
| | Register *0xea*: Target frequency THz part (in units of THz) |
| | Register *0xeb*: Target frequency GHz part (in units of 0.1GHz) |
| | Register *0xec*: Target sled temperature (in units of 0.01C) |
| | Register **0xe9:** Target current (in units of 0.1mA) |
| | Register *0xed*: Execute (1) Clean Jump |
| | Please refer to the **CleanJump Implementation Guide** for further information. |

# 11.  Clean Scan

| | |
|---|---|
| **Description** | Performs a scan over a wider frequency range. The full range is broken up in individual sweeps with Clean Jumps in-between to provide coverage over the full frequency range. |
| | For reasons of speed and accuracy, we recommend a range of 100 GHz for each segment. Note that in the GUI, the segment size is multiplied by 1.2x to have 10% overlap on both extremes. |
| | The intended operation is to have a new center point provided to the laser after each sweep. If no new center point is available, the laser will continue with the next sweep around the same center-point (i.e. same as Clean Sweep). |
| **GUI** | Clean Scan GUI (keyword: 'cleanscan') |
| **Calibration Files** | .map file at the desired power level |
| | .sled file, .current2 file |
| **Firmware Version** | 8.2.x |
| **CLI Commands** | *it.cleanScanAmplitude(x)* to set the Clean Scan range, with x the range in units of GHz |
| | *it.cleanScanEnable(x)* to activate (x=1) and de-activate (x=0) the Clean Sweep |
| | *it.cleanScanOffset()* to read the frequency offset in 0.1 GHz (signed integer) |
| | *it.cleanScanCalibration(x)* sets a calibration factor |
| | *it.cleanScanSetSled(x)* load the target sled temperature for the next centerpoint (x in units of 0.001C) |
| | *it.cleanScanSetF1(x)* load the target filter1 temperature for the next centerpoint (x is 1000 * (temperature - 50)) |
| | *it.cleanScanSetF2(x)* load the target filter2 temperature for the next centerpoint (x is 1000 * (temperature - 50)) |
| **RS-232 Commands** | Register *0xe4*: set the Clean Scan range in units of GHz |
| | Register *0xe5*: activate (1) and de-activate (0) the Clean Scan. The read operation provides status information (bit 0 is set to 1 if the next setpoint has been loaded; bit 1 is set to 1 if the Clean Scan is ongoing; bit 3 and 2 are 01 if the sweep is going to higher frequency and 10 if the sweep is going to lower frequency. |

Register **0xe6**: read the frequency offset in 0.1 GHz (signed integer)

Register **0xe7**: sets a calibration factor from the .current2 file

Register **0xf0-0xf3**: for loading calibration constants derived from the .map and .current2 files.

Please refer to the **CleanScan Implementation Guide** for further information.

# 12. No Drift Calibration

| | |
|---|---|
| **Description** | The standard unit will show a frequency variation with ambient temperature that varies from unit to unit and can be as large as 500MHz over the temperature range. |
| | The No-drift calibration will test each individual laser over temperature and find its optimized correction constants. These constants are loaded to the permanent memory of the unit and accessed directly by the firmware. |
| | No user intervention is needed for this feature. If the calibration occurred, the firmware will apply it.. |
| **GUI** | Basic GUI (keyword: 'basic') |
| **Calibration Files** | Not required. All calibration data is internal of the unit. |
| **Firmware Version** | All versions |
| **CLI Commands** | None |
| **RS-232 Commands** | None |

# 13. No Drift Operating Mode

**Description**

In the standard telecom mode the laser has many control loops running to ensure stability over the long run. For sensor applications we recommend the whispermode, where these control-loops have been mostly disabled. This does result in drift over time, and the laser needs to be put back into standard mode regularly for a short period of time (10secs) for relocking. For critical applications, where long term stability is important (i.e. the laser needs to be locked), but where the drift needs to be limited, the no-drift mode has optimized the control loops to have a factor 10x less noise (some limits apply for the temperature ramp-rate etc.). The no-drift calibration involves the characterization of over temperature variation and compensation of the measured change rate based on ambient temperature.

The unit starts up in the no-drift mode.

**GUI**

Basic GUI (keyword: 'basic')

**Calibration Files**

Not required. All calibration data is internal of the unit.

**Firmware Version**

8.3.x

**CLI Commands**

*it.noDriftMode(x)* to enable(x=1) and disable (x=0) the no-drift mode

**RS-232 Commands**

Register *0x91*: enable (1) or disable (0) the no-drift mode

# 14.  Micro-USB interface (PPCL300 only)

**Description**            Optionally and without extra cost, the PPCL300 can be configured with a micro-USB interface, instead of an RS-232 interface.

**GUI**                    All

**Calibration Files**      Not required

**Firmware Version**       Not firmware specific

**CLI Commands**           Not applicable
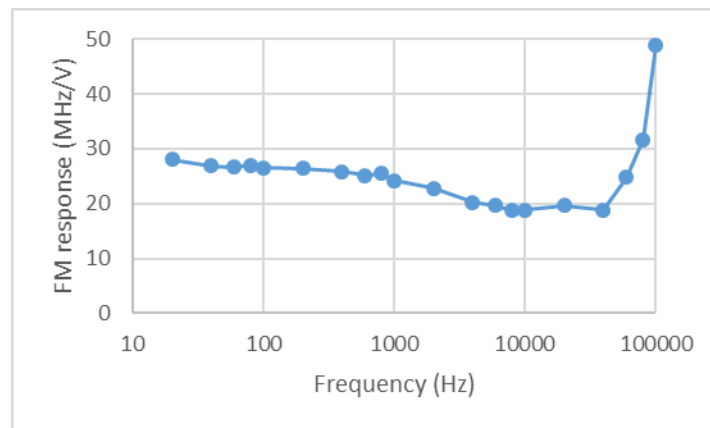
**RS-232 Commands**        Not applicable

# 15. Clean Modulation FM

**Description**
Provides access to the PZT to generate FM modulation without any associated AM modulation. This is a hardware modification.

**Excessive high or low (negative) voltage values and frequencies close to 888Hz should be avoided while the laser is operating in standard (dither) mode.**

The signal is DC coupled and the typical bandwidth is 100kHz. The response varies with frequency (reproduced below as example), but should not exceed 100MHz. The applied voltage should be between 0 and 6V (signal clipping occurs beyond these voltages).



**GUI**                     Basic GUI (keyword: 'basic')

**Calibration Files**       Not required

**Firmware Version**        Not firmware specific

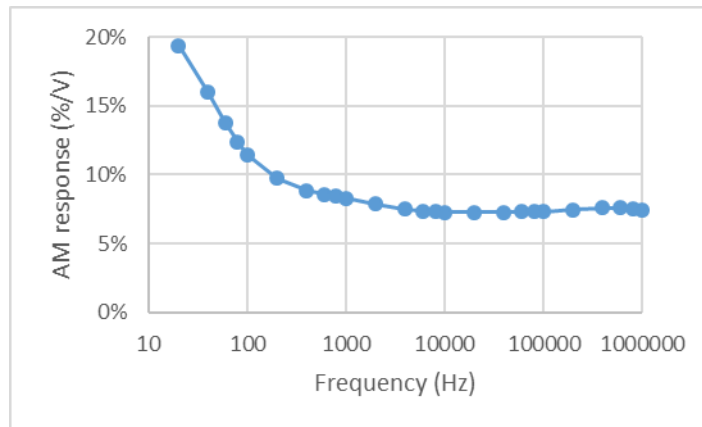**CLI Commands**            Not applicable

**RS-232 Commands**         Not applicable

# 16.  Clean Modulation AM

| | |
|---|---|
| **Description** | Provides access to the current-injection circuit for the gain chip to generate AM modulation. There is associated FM modulation. This is a hardware modification. |

**Excessive high or low (negative) voltage values and frequencies close to 888Hz should be avoided while the laser is operating in standard (dither) mode.**

The signal is DC coupled. Bandwidth is 1MHz. Response depends on the operating power and varies from device to device. Below a typical response at 13.5dBm is provided (response is stronger at lower powers). Typical voltage range is 0-10V.



**This configuration is not compatible with the low RIN option**

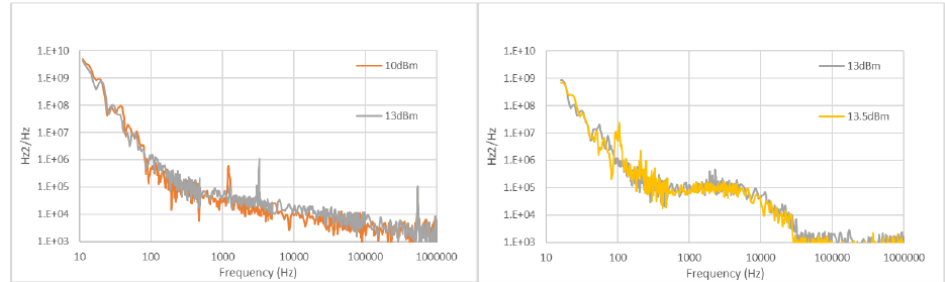| | |
|---|---|
| **GUI** | Basic GUI (keyword: 'basic') |
| **Calibration Files** | Not required |
| **Firmware Version** | Not firmware specific |
| **CLI Commands** | Not applicable |
| **RS-232 Commands** | Not applicable |

# 17.  Clean Measurement

| | |
|---|---|
| **Description** | The Clean Measurement input (2 channels possible) feeds into a micro-processor 12-bit ADC line. It can be used to feed information to the laser through an analog channel. The usage of that information is defined in the firmware. |
| | For example, one implementation is to map the measured voltage to a voltage offset ('analog FTF'). In this way a feedback loop for frequency stabilization can be created. Other usage might be to detect the signal of a photo-diode while the laser is scanning. This allows integration of laser control and PD read-out and should provide better accuracy. |
| **GUI** | Basic GUI (keyword: 'basic'), for implementation of the analog FTF |
| **Calibration Files** | Not required |
| **Firmware Version** | 8.0.9 for the analog FTF implementation |
| **CLI Commands** | None |
| **RS-232 Commands** | For the analog FTF implementation |
| | Register *0xf8*: set range associated with analog FTF in units of 10MHz (e.g. a value of 1000, means that 0V input is -10GHz and 10V is +10GHz) |

# 18.  Low RIN Configuration

**Description**            This hardware modification adds additional filtering on the current supply, lowering the RIN significantly in the 50kHz – 10MHz range. In addition, the FM noise is lowered to the Lorentzian noise level in that frequency range.

Phase Noise (before / after)



**This configuration is not compatible with AM modulation.**

**GUI**                    Basic GUI (keyword: 'basic')

**Calibration Files**      Not required

**Firmware Version**       All firmware versions

**CLI Commands**           None

**RS-232 Commands**        None

# 19.  PPCL5x0 Enclosure

**Description**          The PPCL5x0 enclosure solution takes a PPCL100/PPCL200 (PPCL500) or PPCL300 (PPCL550) and integrates it into an enclosure along with power supplies and a micro-USB interface.

The unit is delivered with wall-plug power supply (transformer) and micro-USB cable.

For units with analog inputs/outputs SMA connectors are used.

All features that are available on PPCLx00 are available within the enclosure solution.



| | |
|---|---|
| **GUI** | All GUIs |
| **Calibration Files** | Not required |
| **Firmware Version** | All |
| **CLI Commands** | None |
| **RS-232 Commands** | None |