

# New Python Shaping

## Login

```
ssh psdev (if remote)
ssh mec-laser
mecpython
import meclas
```

## Go to meclas level

```
%run /reg/g/pcds/pyps/apps/hutch-python/mec/mec/macros/meclas.py
```

## meclas.YFE

```
meclas.YFE.Get          meclas.YFE.OnCheck meclas.YFE.Trace
meclas.YFE.Off          meclas.YFE.Set
meclas.YFE.On           meclas.YFE.SetAll
```

## meclas.LPL

```
meclas.LPL.Off          meclas.LPL.get_curr_exp meclas.LPL.psefc10Hz meclas.LPL.pspostshot meclas.LPL.psrefrwvm
meclas.LPL.On           meclas.LPL.get_curr_run meclas.LPL.psloadwvm meclas.LPL.pspreshot meclas.LPL.pssavewvm
meclas.LPL.SHG_opt      meclas.LPL.get_curr_shape meclas.LPL.psmenu meclas.LPL.psrecipes meclas.LPL.psviewwvm
```

## meclas.EMeters.

```
meclas.EMeters.EG          meclas.EMeters.EGall meclas.EMeters.LPLInChamber
meclas.EMeters.EG1w2in     mec...rs.E_coeff_refresh meclas.EMeters.SPLEG
meclas.EMeters.EG1wYFE1in mec...rs.E_synth_refresh meclas.EMeters.gentec_refresh
```

## meclas.TTL\_shutter.

```
meclas.TTL_shutter.Refresh
meclas.TTL_shutter.Status
meclas.TTL_shutter.Toggle
```

- `TTL_shutter.Status()` tells you which TTL shutters are currently opened vs closed
- `TTL_shutter.Toggle('openall')` opens all TTL shutters

- `TTL_shutter.Toggle('closeABEF')` closes AB and EF TTL shutters
- `TTL_shutter.Toggle('openWW')` opens WEST TTL shutter
- `TTL_shutter.Toggle('closeXX')` closes EAST TTL shutter
- If you toggle the shutters using the button or the EPICS screen instead of Python, nothing is keeping track of the shutter state, so you will confuse the laser -- it will think it has one state when really it has another! As such, never touch the button or the GUI unless absolutely necessary!
- If things get out of whack, you can always come to the hutch, visually verify that ALL SHUTTER ARE OPEN, and then type in the command: `TTL_shutter_refresh()` This will reset the counter/reference back to being all open.

## meclas.LPL.ps

```
meclas.LPL.psefc10Hz meclas.LPL.pspostshot meclas.LPL.psrefrwvfm
meclas.LPL.psloadwvfm meclas.LPL.pspreshot meclas.LPL.pssavewvfm
meclas.LPL.psmenu meclas.LPL.psrecipes meclas.LPL.psviewwvfm
```

## meclas.HWP.

```
meclas.HWP.ClearStart meclas.HWP.On
meclas.HWP.HWP_opt meclas.HWP.Status
```

`meclas.HWP.O('all')` = rotation to 0 degree for full transmission

## Pulse Shaping

- Parameters:
  - YSSs: start and stop heights of seed pulse
  - SSs: start and stop heights of amplified pulse
  - Psns: pulse duration
- Read and set these parameters
  - `LPL._YSSs_get()` → output: `[[ y1, y2 ]]`; `LPL._YSSs_set([[ y1, y2 ]])`
  - `LPL._SSs_get()` → output: `[[ s1, s2 ]]`; `LPL._SSs_set([[ s1, s2 ]])`

LPL.\_Psns\_get() → output: [8.25] ; LPL.\_Psns\_set( t )

- For example: to make 20% gradient with 30ns pulse
  1. LPL.\_Psns\_set([ 30.25 ])
  2. LPL.\_SSs\_set([[ 80, 100 ]])
  3. LPL.\_YSSs\_get() → [[ y1, y2 ]]
  4. LPL.\_YSSs\_set( [ [ y1 \*c1, y2 \*c2 ] ] ) c1& c2 are adjusting factors btwn 0 and 1
  5. LPL.pspresshot(), take a shot, LPL.pspostshot(display=True, save\_flag=False)
  6. If further adjustment is needed, go back step 4 changing c1 or c2
- How to remove spike: if you have a little spike in the entrance

[Automatically adjust the waveform: use Automatically adjust the entrance: LPL.psefc10Hz](#)

Lower the first 3 points by 0.97:

```
LPL.psefc10Hz(pwt='curr',numIterQ=50,AQQ=0.03,displayPlot=True,reloopPrompt=True,YFEbkgrdY=-.004,PtNumFront=3,PtNumBack=2,CorrFactorFront=.97,CorrFactorBack=1.0,avgfwhm=9,avgrange=1):
```

[Manually adjust the waveform](#): altering the Highland settings directly:

```
Highland = HAWG()
```

```
current_wvfm = Highland.ReadPulseHeights()
```

```
new_wvfm = current_wvfm[:]
```

adjust new\_wvfm points manually exactly however you'd like, e.g. new\_wvfm[0] = 10000 or whatever

To plot: **ep.l(new\_wvfm)**

```
Highland.WritePulseHeights(new_wvfm)
```