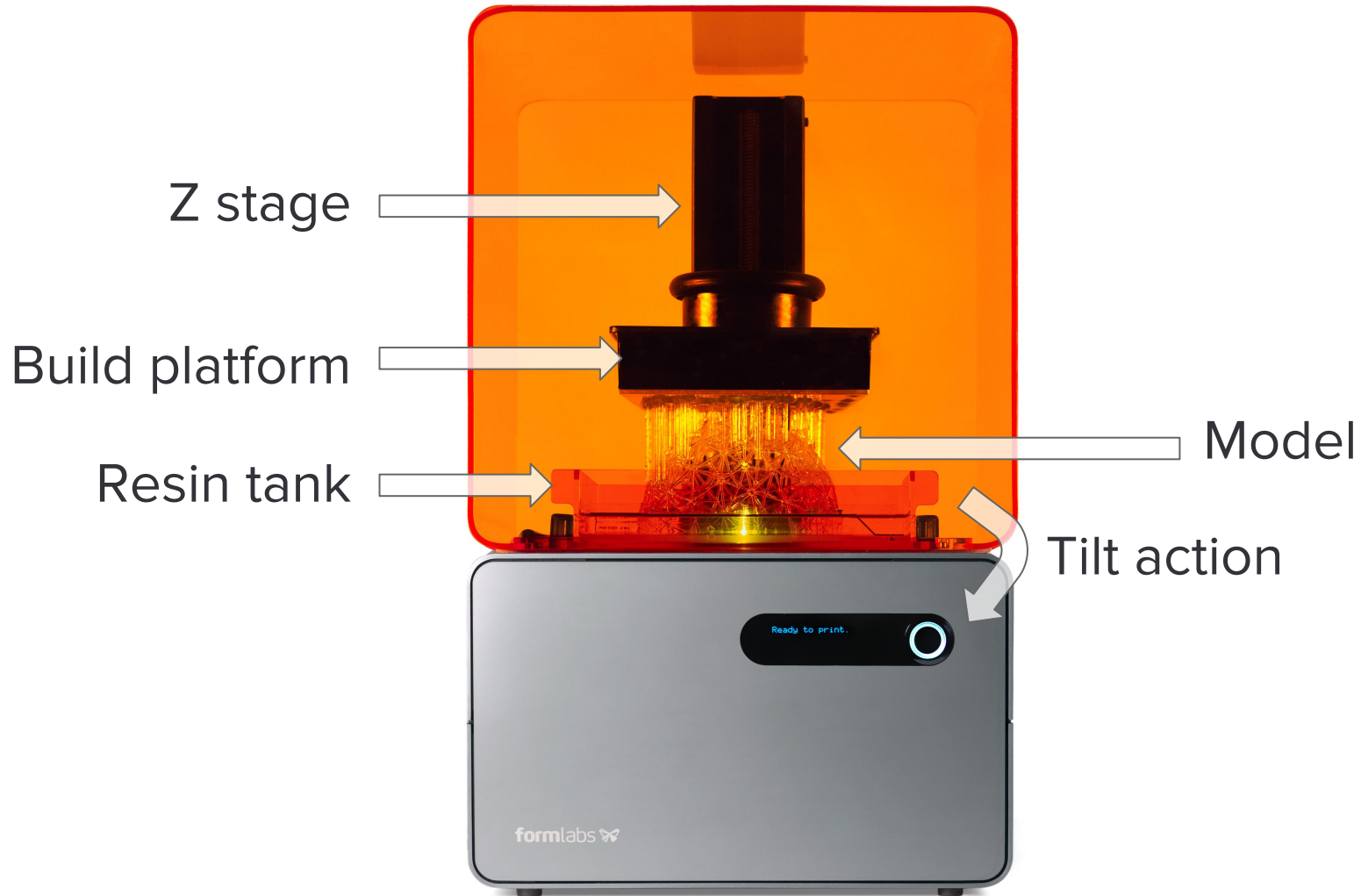


formlabs 

OpenFL: Hacking SL with the Form1+ API

Ben FrantzDale and Matt Keeter
{ben, matt}@formlabs.com

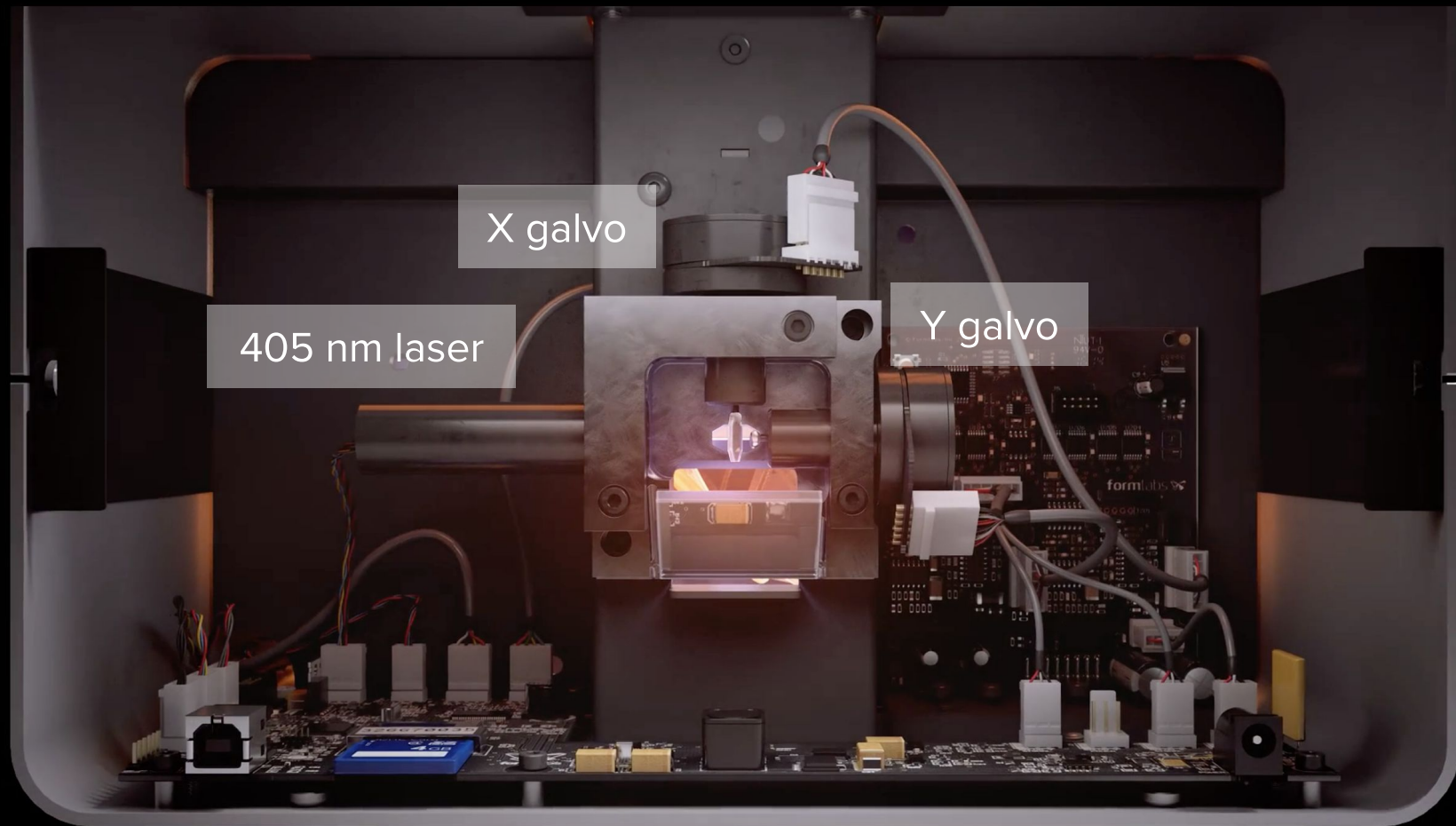




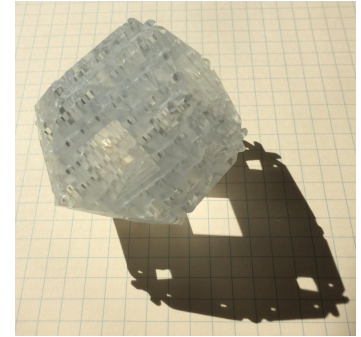
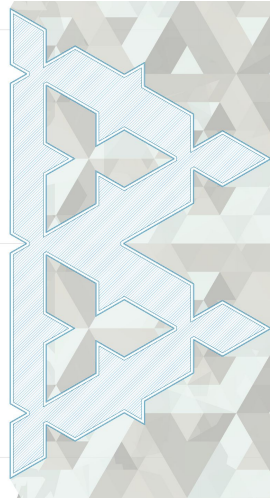
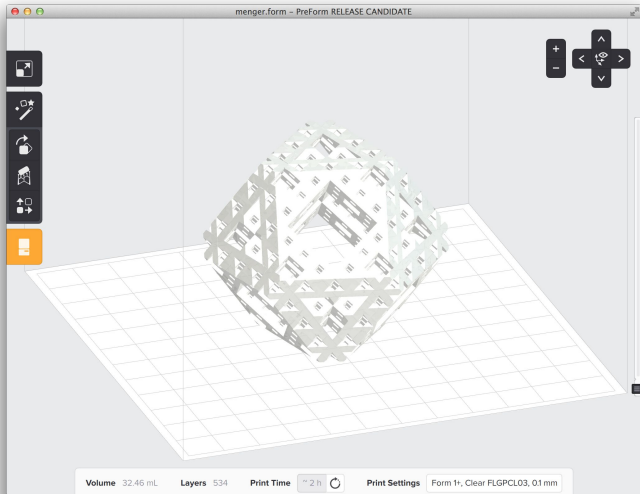
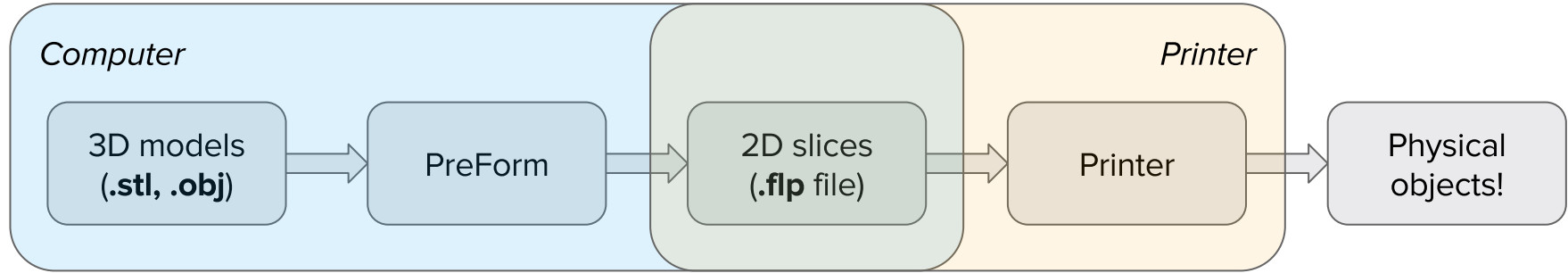
X galvo

405 nm laser

Y galvo



Print pipeline

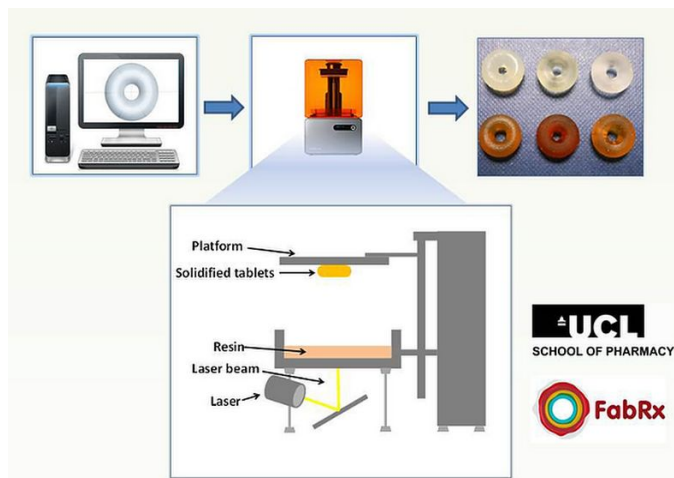


OpenFL is about supporting cases that fall outside the usual Form 1 / 1+ printing pipeline.



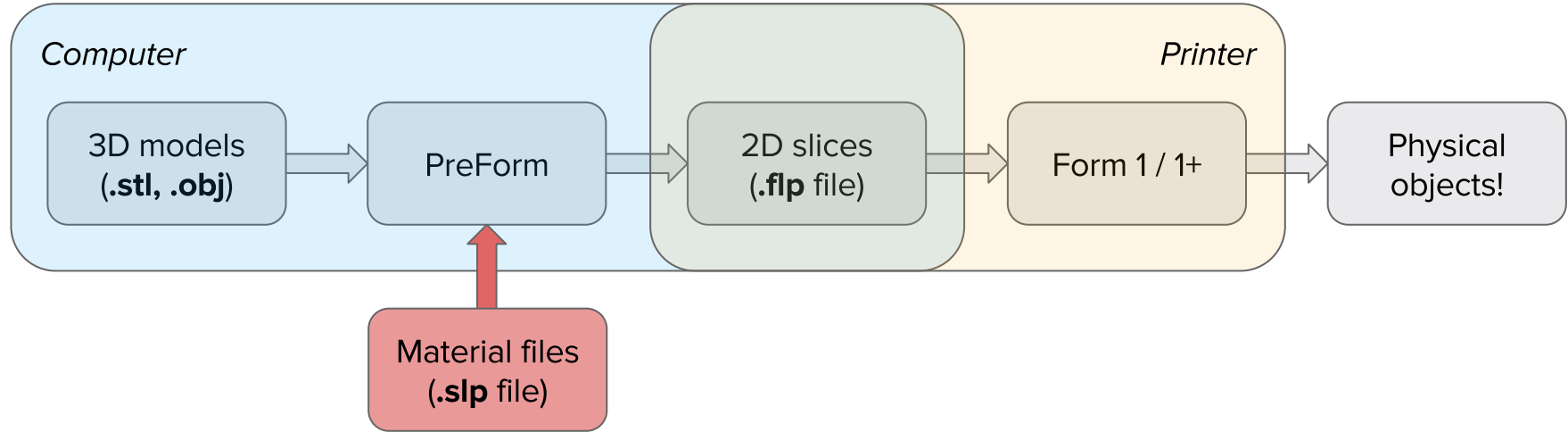
Stereolithographic (SLA) 3D printing of oral modified-release dosage forms

Jie Wang^{a, 1}, Alvaro Goyanes^{a, b, 1}, Simon Gaisford^{a, b}, Abdul W. Basit^{a, b},  



Customizing material files on Form 1 and Form 1+

User-defined material files



PreForm: Material Hacker Edition*

Use customized material files with PreForm and your Form 1/1+

```
[PrintSettings]
```

```
OuterBoundaryOffset = 0.03
```

```
InnerBoundaryOffset = 0.12
```

```
ScanlineBoundaryOffset = 0.03  
outlines
```

```
ScanlineSpacing = 0.09
```

```
SliceHeight = 0.025
```

```
OffsetsNum = 3
```

```
Xcorrectionfactor = 1.008
```

```
Ycorrectionfactor = 1.008
```

Distance from the model boundary - theoretically is laser's radius

Distance between any inner boundary offset

Distance from the innermost boundary offset - defines raster

Line positioning

The number of outlines

x shrinkage-correction scale

y shrinkage-correction scale

```
[perimeter]
```

```
modellaserpowermw = 20.7
```

```
modelxyfeedrate = 800 ; mm/s
```

```
supportlaserpowermw = 24.84
```

```
supportxyfeedrate = 800
```

```
baselaserpowermw = 24.84
```

```
basexyfeedrate = 800
```

Exposure

```
[fill]
```

```
modellaserpowermw = 20.7
```

```
...
```

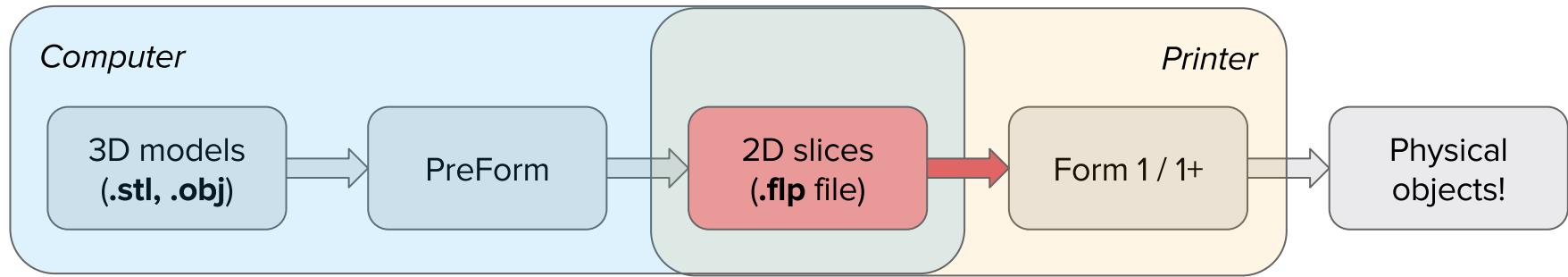


*name subject to change

Coming next week!

OpenFL API

What does it cover?



Two main modules:

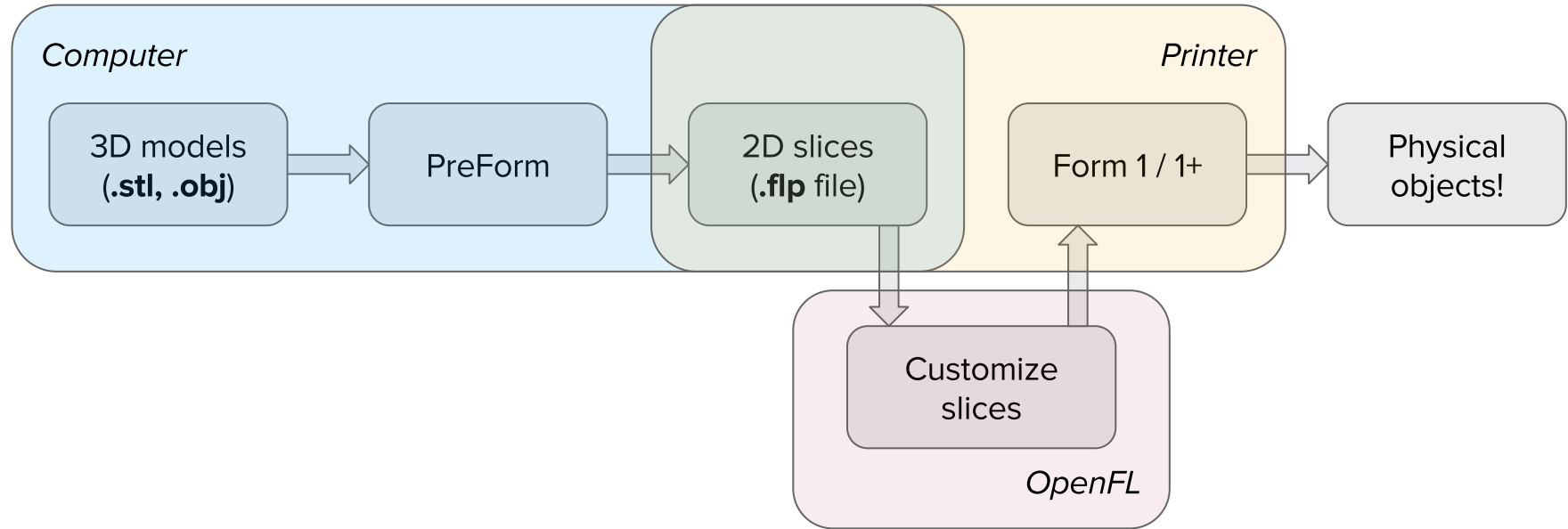
`OpenFL.FLP`
Read, manipulate,
and write 2D slice files

`OpenFL.Printer`
Interface between your
computer and a Form 1 / 1+

<https://github.com/Formlabs/OpenFL>

Modifying Formlabs Print Files (.FLPs)

Customizing prints



What does a print look like?

2D slices
(.flp file)

```
[<TimeRemaining(1891 s) at 0x10b668470>,  
  <XYMoveClockRate(60000 Hz) at 0x10b668890>,  
  <ZCurrent(80) at 0x10b6688f0>,  
  <TiltCurrent(80) at 0x10b668950>,  
  <TiltFeedRate(472 usteps/s) at  
0x10b6689b0>,  
  <TiltMove(2362 usteps) at 0x10b668a10>,  
  <ZFeedRate(132 usteps/s) at 0x10b668a70>,  
  <ZMove(-665 usteps) at 0x10b668ad0>,  
  <ZFeedRate(4000 usteps/s) at 0x10b668b30>,  
  <WaitForMovesToComplete() at 0x10b668b90>,  
  ...  
  <LaserPowerLevel(39099) at 0x106e97b90>,  
  <XYMove(3 points) at 0x10b667c18>,  
  <LaserPowerLevel(0) at 0x106e97bf0>,  
  <XYMove(1 points) at 0x10b667c80>,  
  <LaserPowerLevel(39099) at 0x106e97c50>,  
  <XYMove(2 points) at 0x10b667ce8>,  
  <LaserPowerLevel(0) at 0x106e97cb0>,  
  ...]  
...]
```

Laser moves:

Power then (x, y, dt) sequence.

```
>>> print laserSequence  
[<LaserPowerLevel(39099) at 0x106e97c50>,  
 <XYMove(3 points) at 0x10b667c18>,  
 <LaserPowerLevel(0) at 0x106e97bf0>]
```

```
>>> print laserSequence[1].points  
((38352, 32099, 69),  
 (38322, 32069, 3),  
 (38262, 32166, 8))
```

Motor Moves

Set current and speed, start moves, wait.

```
[ZCurrent(moving=True),  
TiltCurrent(moving=True),  
TiltFeedRate(usteps_per_s=472),  
ZFeedRate(usteps_per_s=132),  
TiltMove(usteps=2362),  
ZMove(usteps=-665),  
WaitForMovesToComplete(),  
TiltCurrent(moving=False),  
ZCurrent(moving=False)]
```

Embed things in prints.

```
from OpenFL import FLP
from OpenFL import Printer
from examples.insert_material_swaps import insert_pause_before

p = Printer.Printer() # Connect to the printer
layer_i = 8
flp = p.read_block_flp(layer_i)
flp = insert_pause_before(flp, zJog_mm=150.0 - 0.2*i)
# Overexpose the next layer w/ 6 more copies of the laser move:
flp += [laser for laser in flp
        if isinstance(laser, FLP.LaserCommand)] * 6
p.write_block_flp(layer_i, x) # Send it back to the printer
p.start_printing(0, 16) # Print!
```



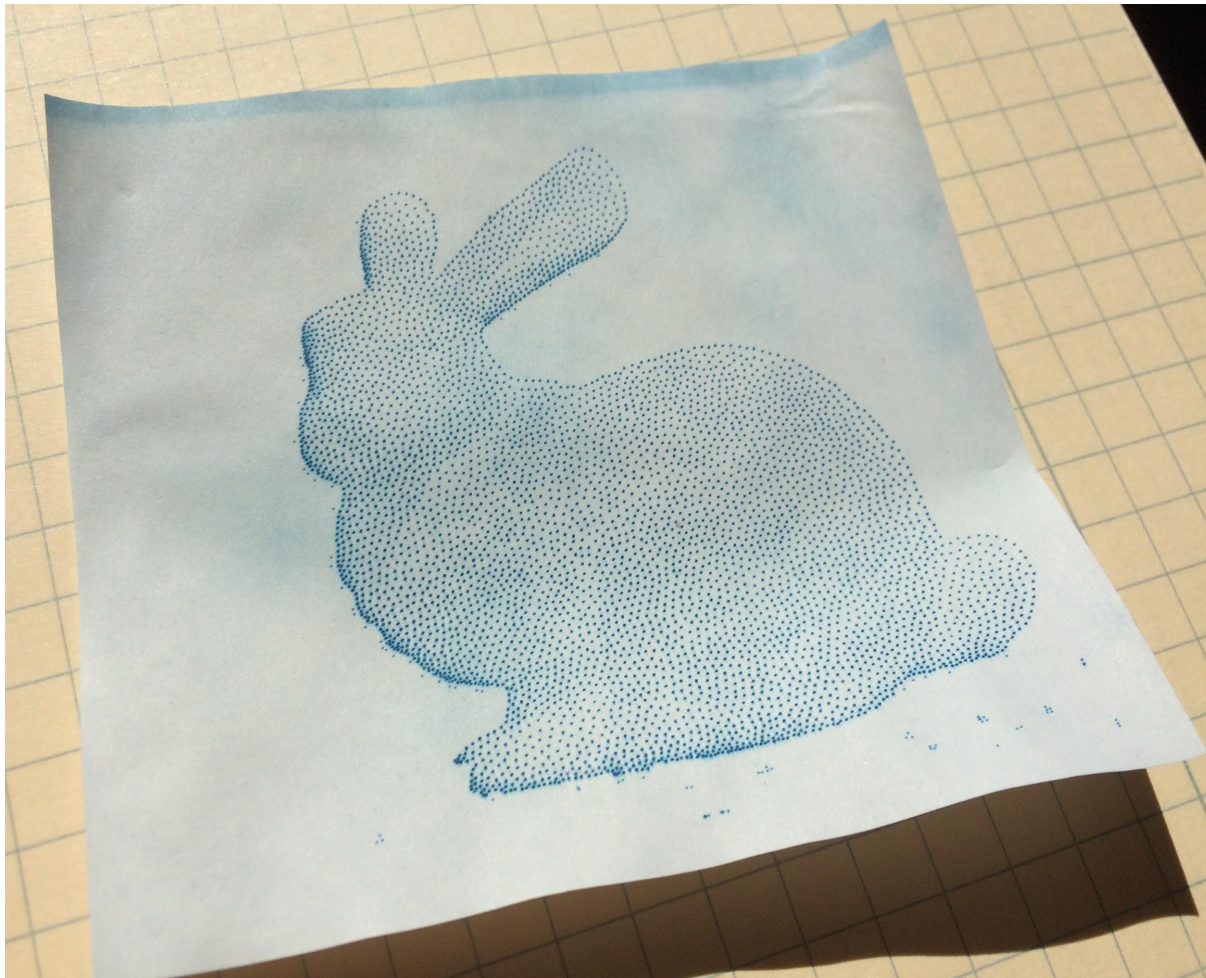
Things... like
carbon fiber

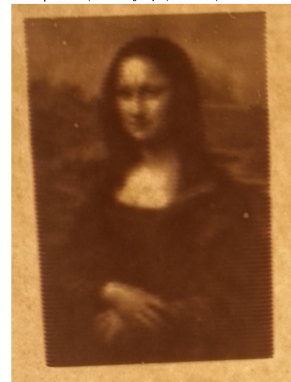
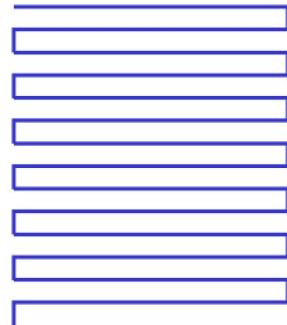
Creating custom designs



UV-reactive stippling

Stipple patterns are generated in Python then printed onto UV-sensitive paper, turning the 3D printer back into a 2D printer.



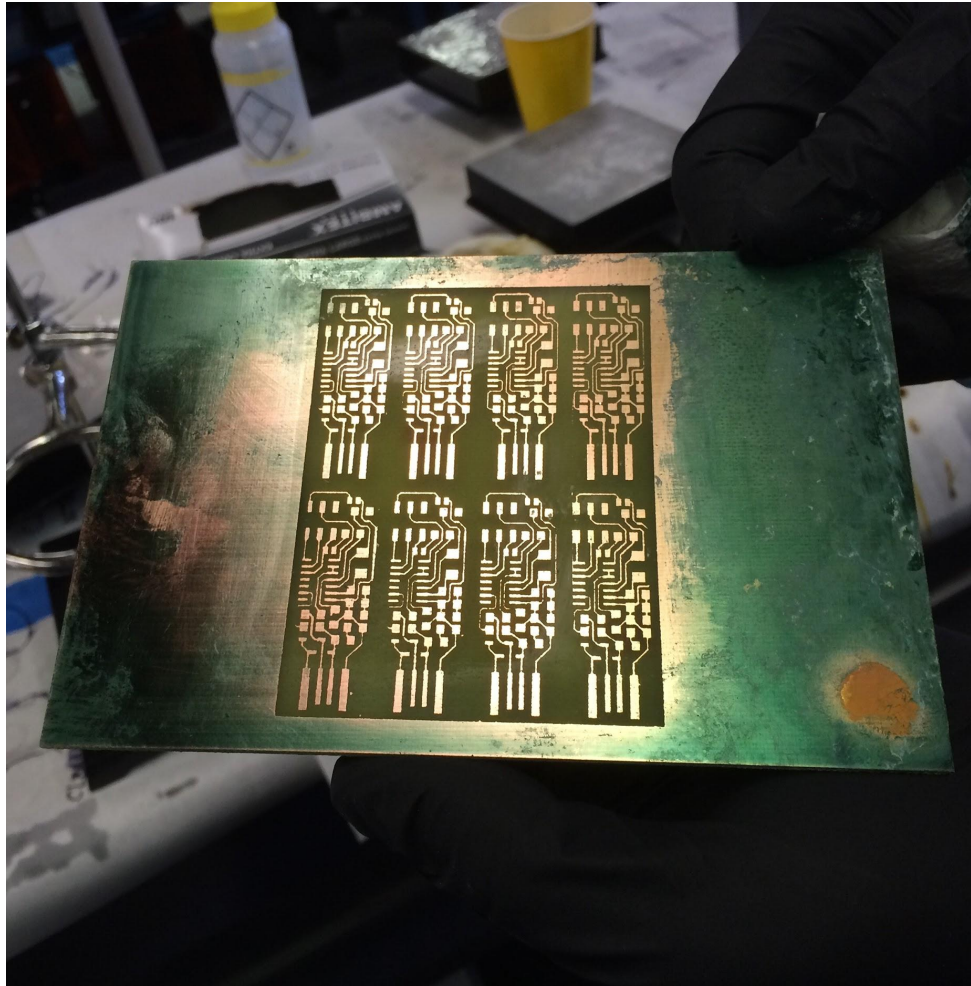


Single-line lithopane

The laser is scanned across the build platform with continuously changing power

The resin has a log response, so pixel brightness can map directly to laser power.

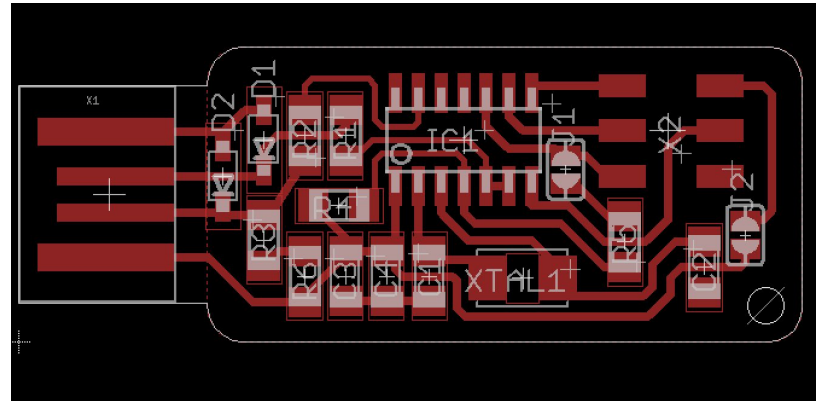
Project by Dima Megretski



PCB etching

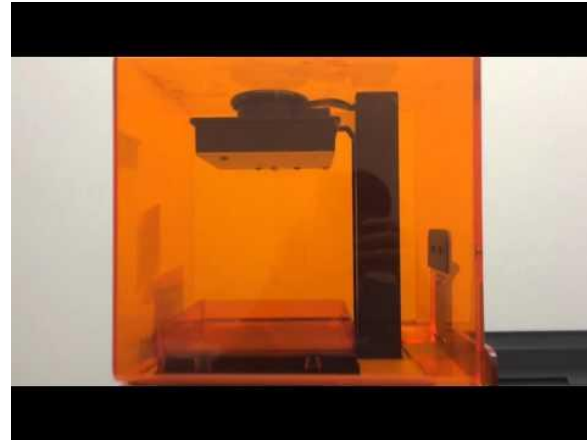
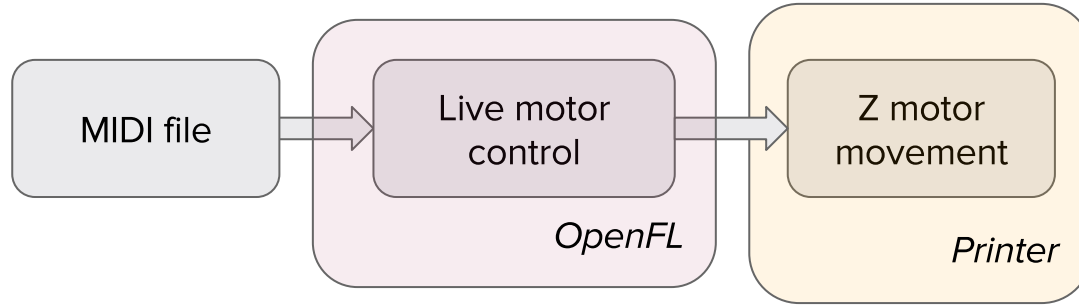
The laser is used to expose a pre-sensitized PCB, which is then etched to make a panel of boards.

fabiskey by Andy Bardagjy



Fun!

Printers as musical instruments



Thanks!

<https://github.com/Formlabs/OpenFL>

formlabs 