資料２

Fab Academyと日本における IoTクリエイター養成講座

ファブラボ鎌倉
慶応義塾大学SFC研究所訪問研究員
加藤 未央

目次

- 自己紹介
- Fab Academyとは
- Fab Academyの学習内容
- Fab Academyを支えている基盤技術
- 2015年 - 日本からのFab Academy参加
- 日本で行っているIoTクリエイター養成講座

自己紹介

2012 : 6年間勤めた計測機器メーカーを退社

2013 : Fab Academyにオランダ、アムステルダムから参加

2014

- ファブラボ鎌倉で働き始める

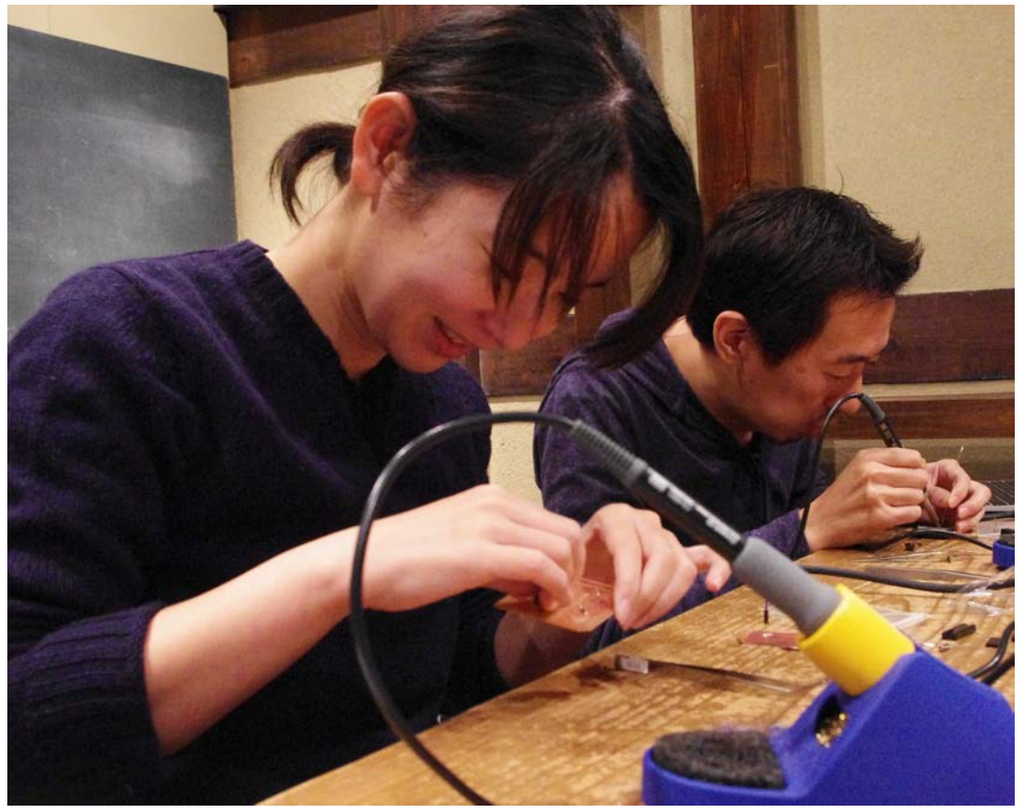
- 慶応義塾大学SFC研究所訪問研究員になる

2015 : ファブラボ鎌倉からFab Academyにローカルチューターとして参加

Fab Academyとは

Global

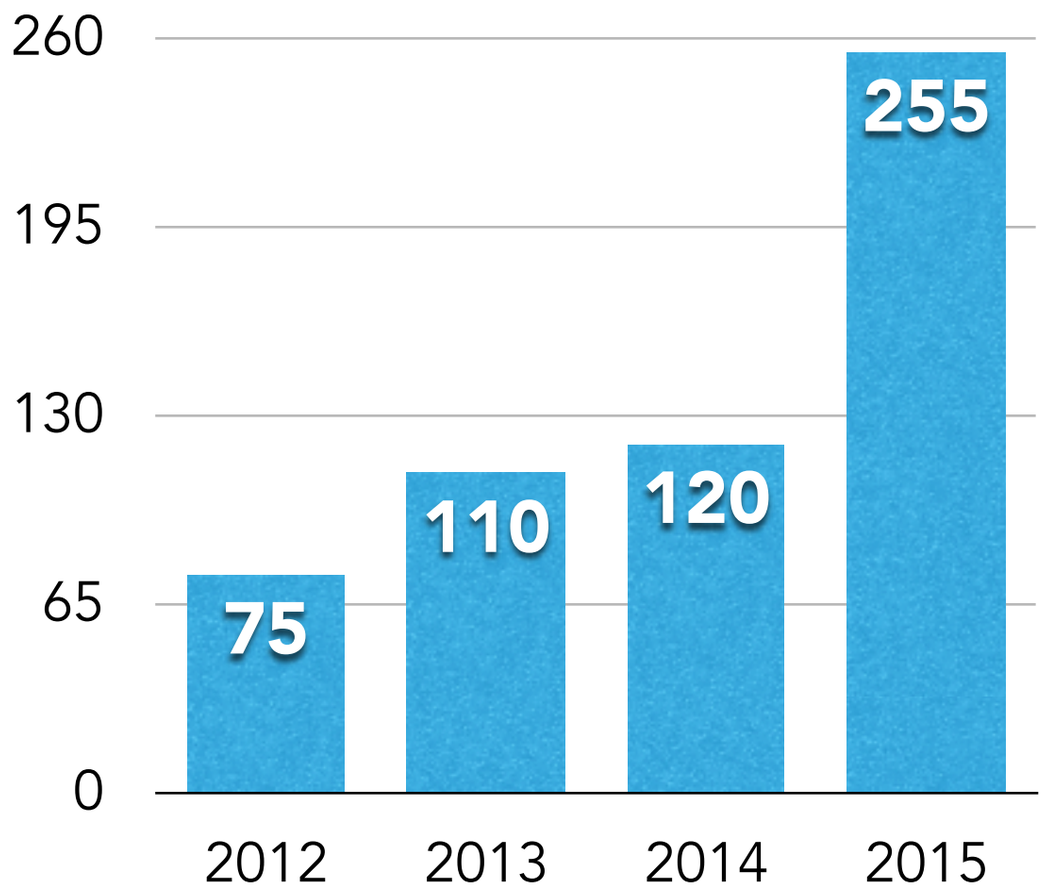
Local



- 世界52箇所のファブラボから255人の生徒が参加 (2015)
- 多様なバックグラウンド(エンジニア、デザイナー、マネートレーダーなど)
- How to make almost anything (アイデアを形にするあらゆる手法を学ぶ)
- MIT Center for Bits and Atomsのニール・ガーシェンフェルド教授が始める

概要

- 参加人数 : 255人 (2015年)
- 参加ラボ : 52箇所 (2015年)
- リージョン
  - North America
  - South America
  - Europe / Middle East
  - Africa
  - Asia
- 受講料 : $5,000
- スカラーシッププログラム
- 授業開始 EST9:00 (日本 PM11:00)



Fab Academyの学習内容

1. **Principles and Practices, Project Management**
2. **Computer-Aided Design**
3. **Computer Controlled Cutting**
4. **Electronics Production**
5. **3D Scanning and Printing**
6. **Electronics Design**
7. **Embedded Programming**
8. **Computer-Controlled Machining**
9. **Molding and Casting**
10. **Input Devices**
11. **Output Devices**
12. **Composites**
13. **Networking and Communications**
14. **Mechanical Design, Machine Design**
15. **Interface and Application Programming**
16. **Invention, Intellectual Property, and Income**
17. **Project Development**
18. **Project Presentation**

- グローバルレクチャー 1.5h (毎週水曜日)
- グローバルレビュー 1.5h (毎週水曜日)
- ローカルレビュー
- ビデオレコーディング (Vimeo)
- 作品製作記録 (Mercurial)
- ハンズオン講習 (12h以上)
- メーリングリスト
- 18weeks

Red = 筐体

Green = 電子回路

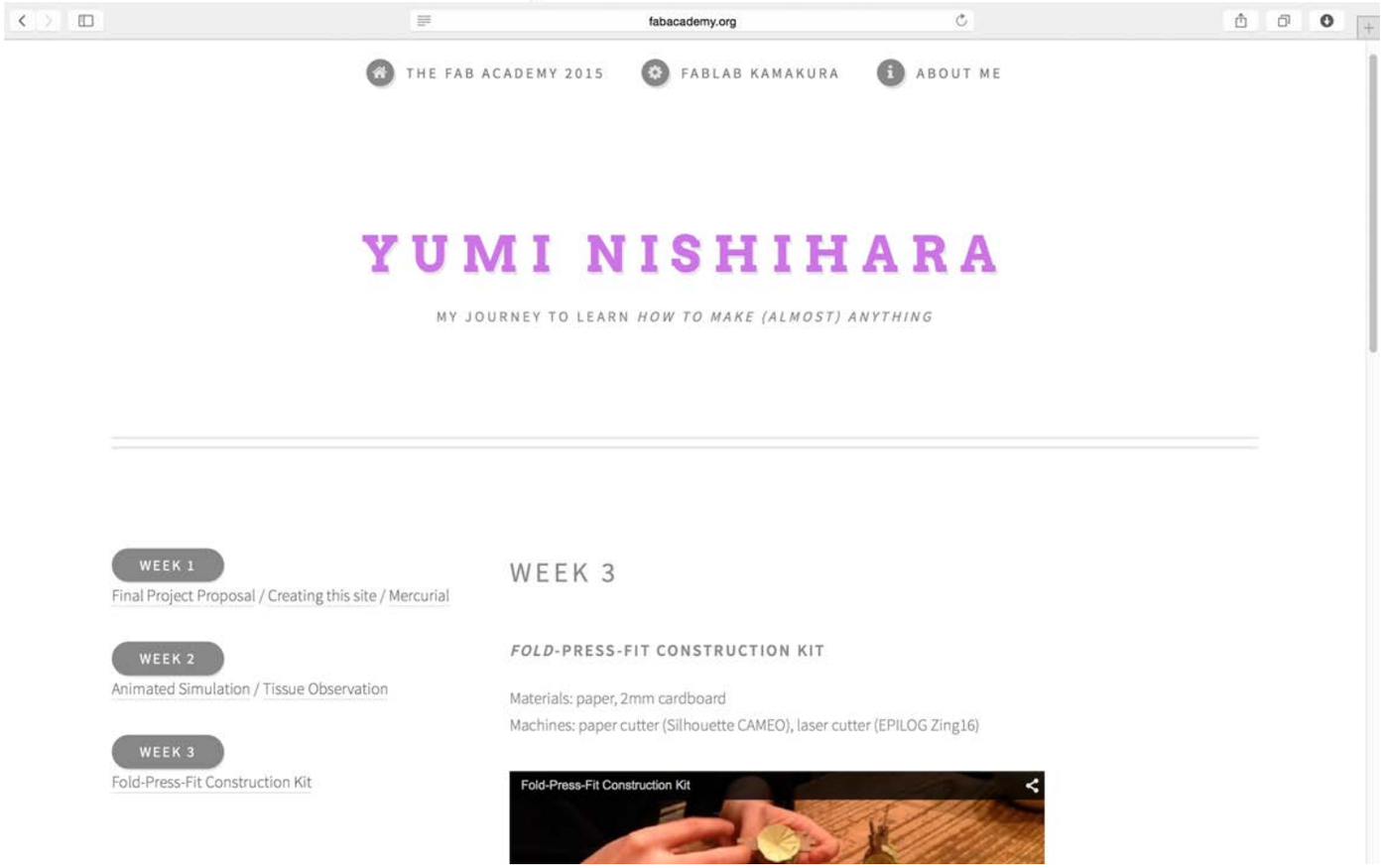
Blue = ソフトウェア

Brown = 機械

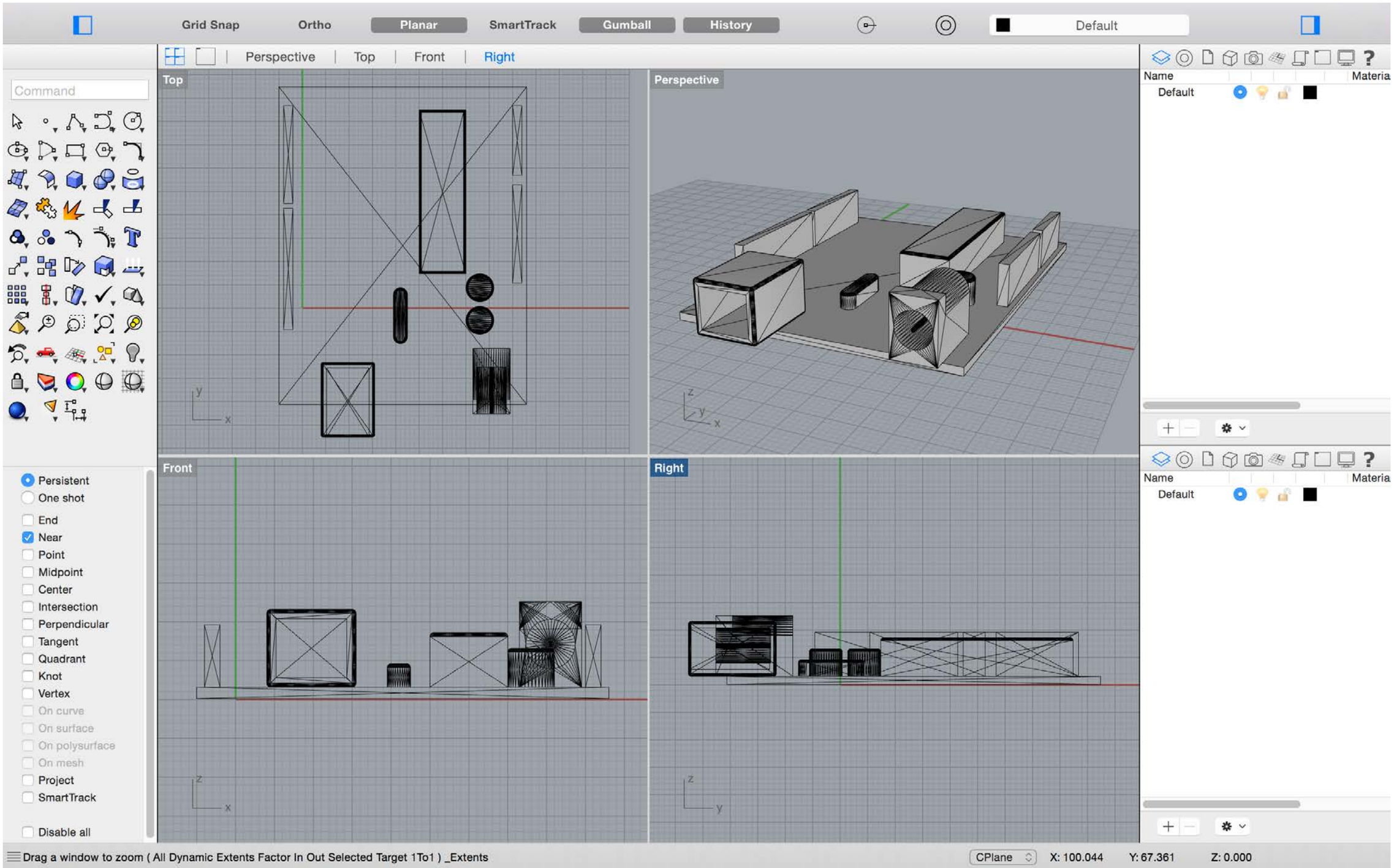
Project Management


Random page

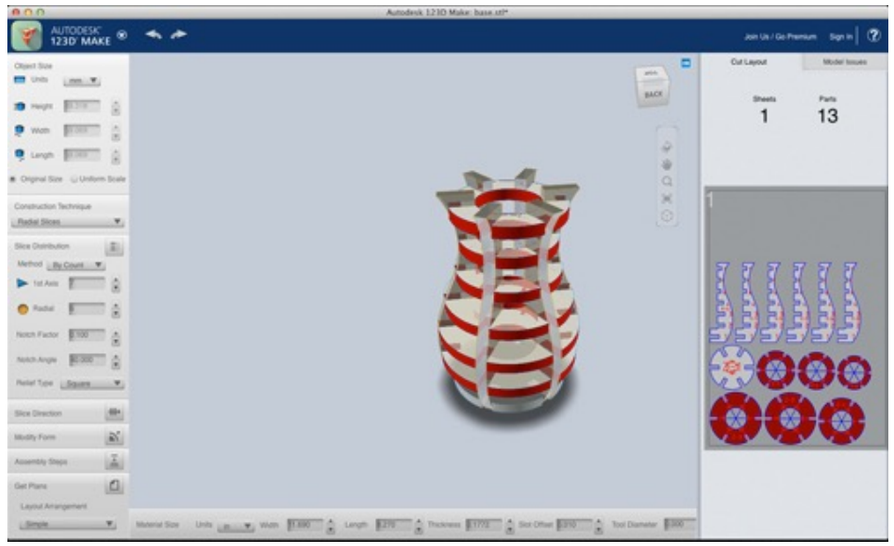
Adolfo Benítez Herrera
Adom Kwabena
Ahmed Al Balooshi
Alejandro Escario Méndez
Alessandra Ferreira
Alessandro Papaleo
Alex Brufsky
Alexander Nikolas Walzer
Ali Awadh
Ali Rajaie
Allen Maker
Aman Garg
Ana Karyna Gómez
Anders Haldin
Andrew Gregson
Andrew Moore
Anna Aflalo
Antonio Alliva
Antonio Burrai
Apeksha Bochare
Arely Amaut
Aris Masvros
Aristarco Cortes
Asako Okazaki

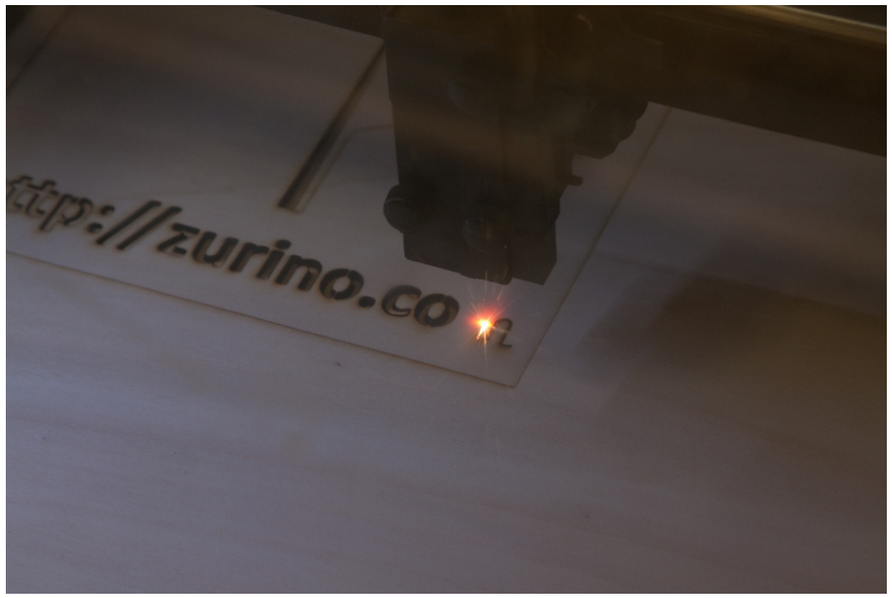


Computer Aided Design

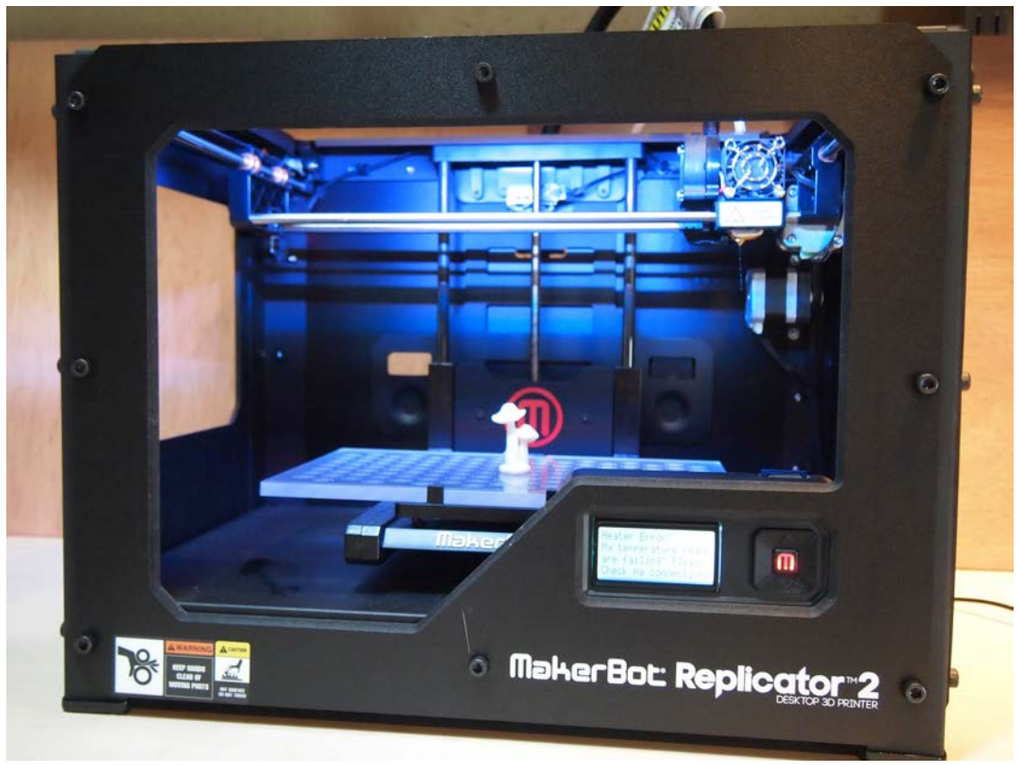


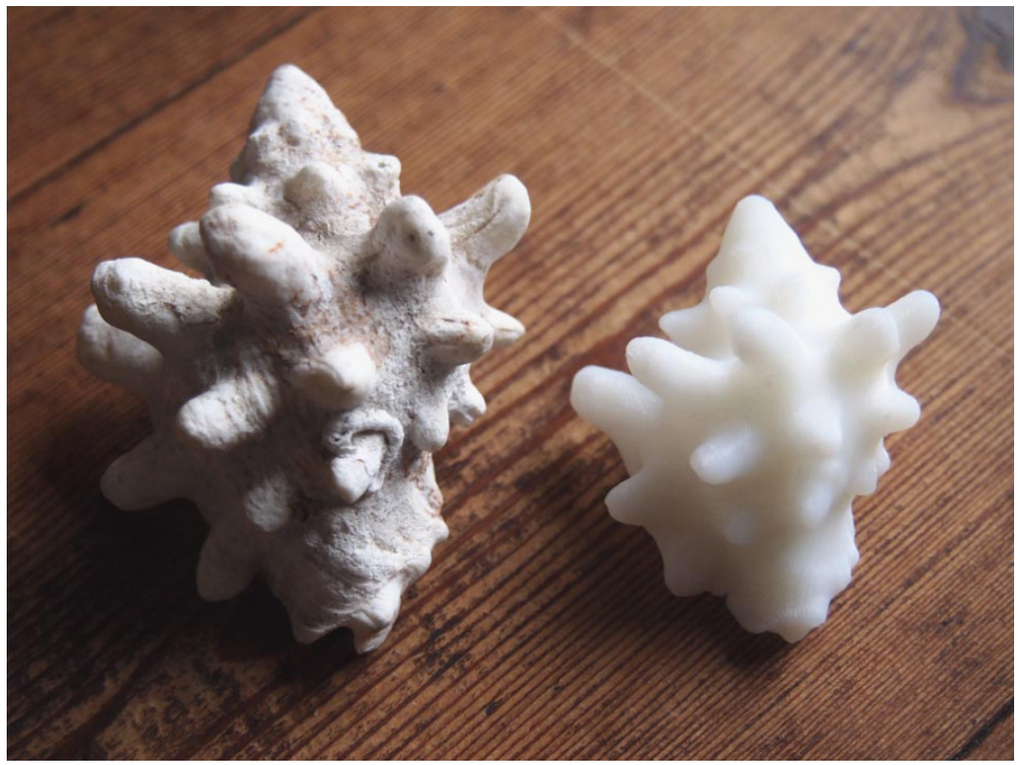
Computer Controlled Cutting





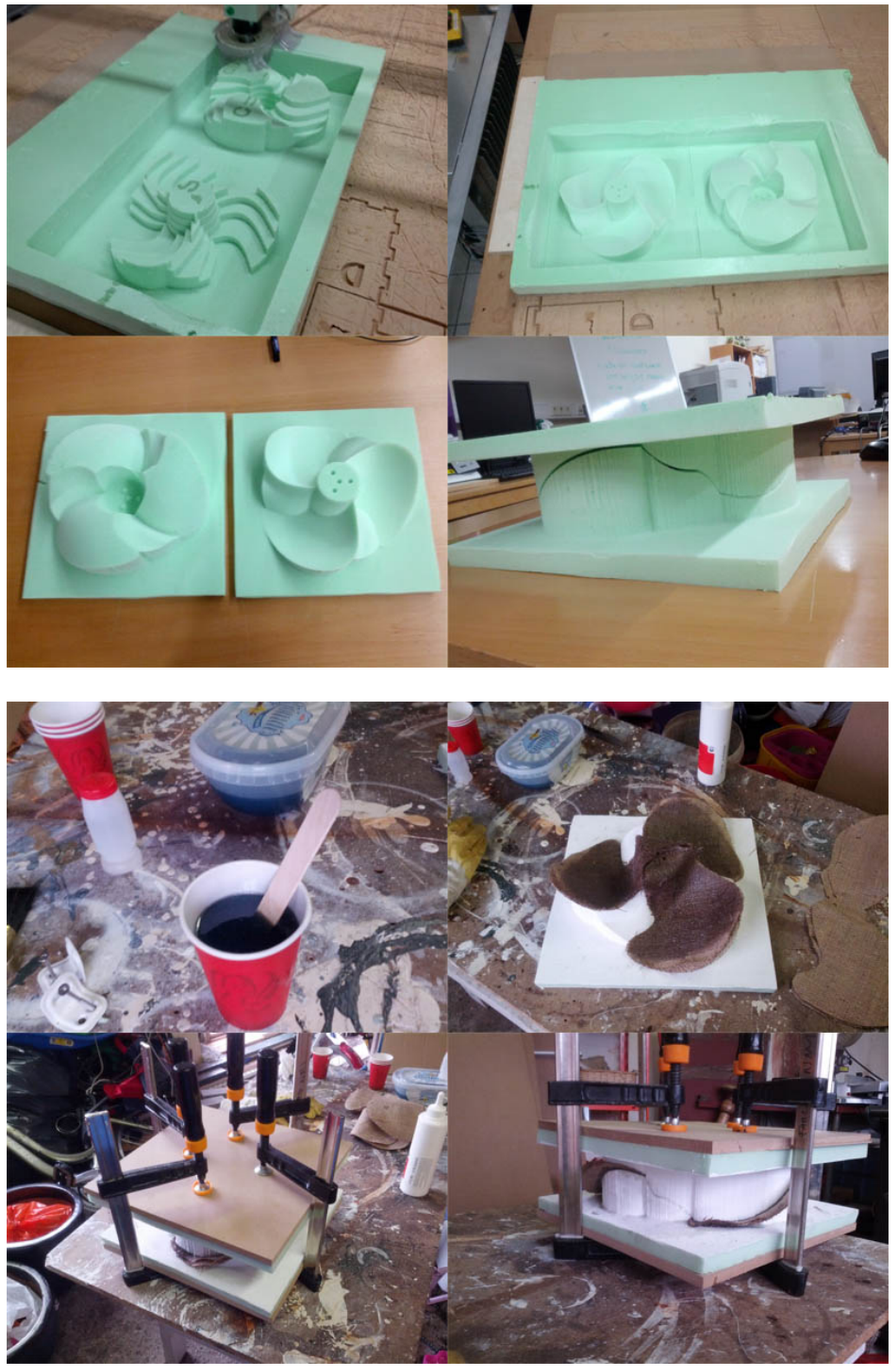
3D Scanning and Printing

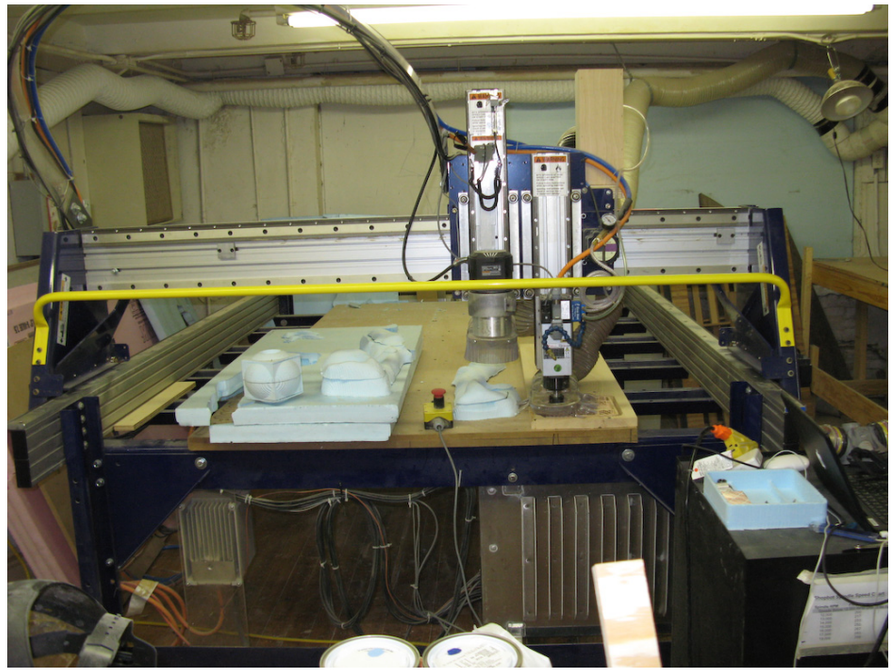


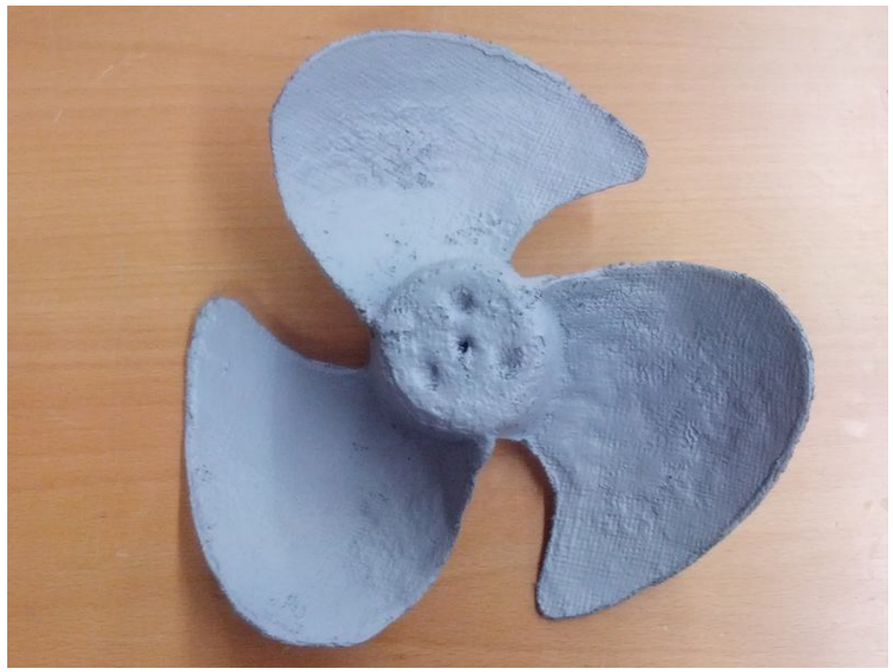


Computer Controlled Machining

Molding and Casting
Composites

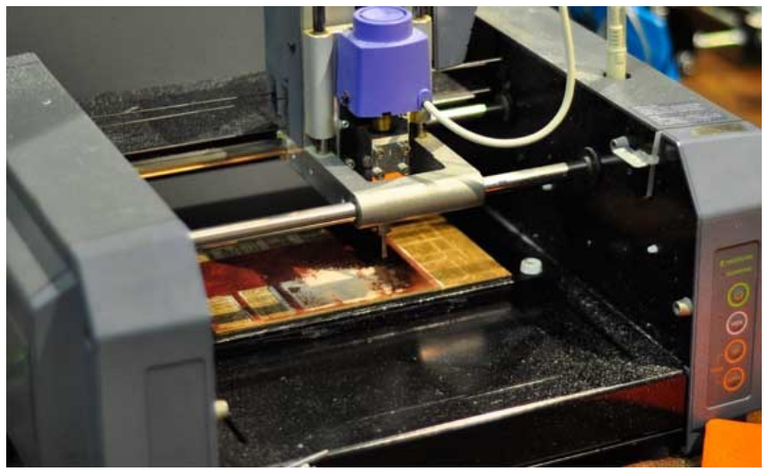




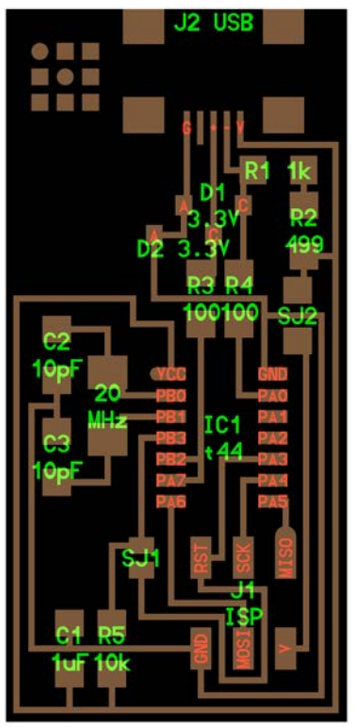


Electronics Production

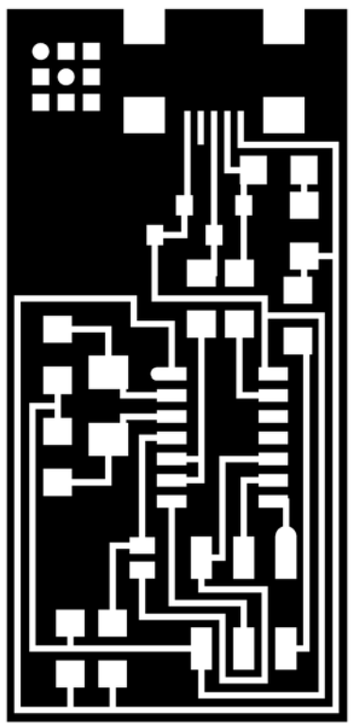
Electronics Design



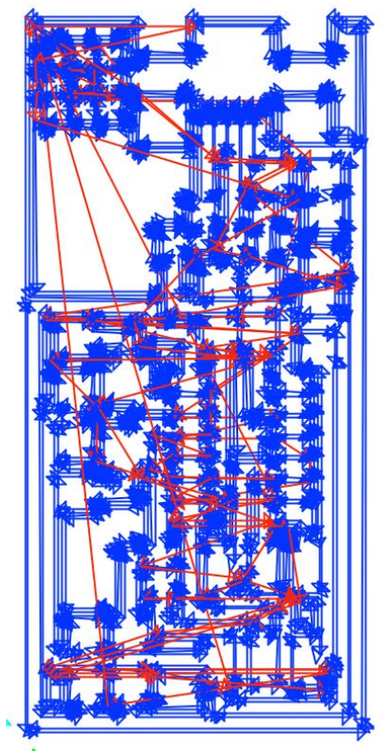
Eagle file



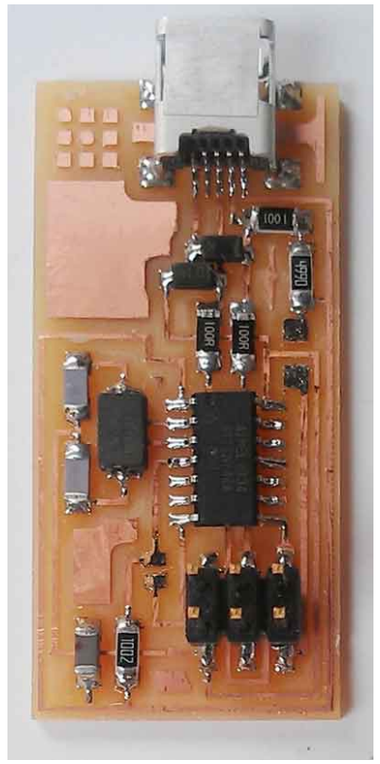
png file



path data



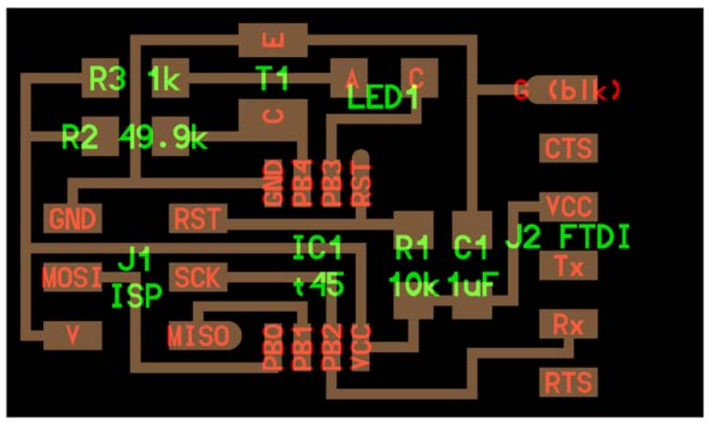
pcb



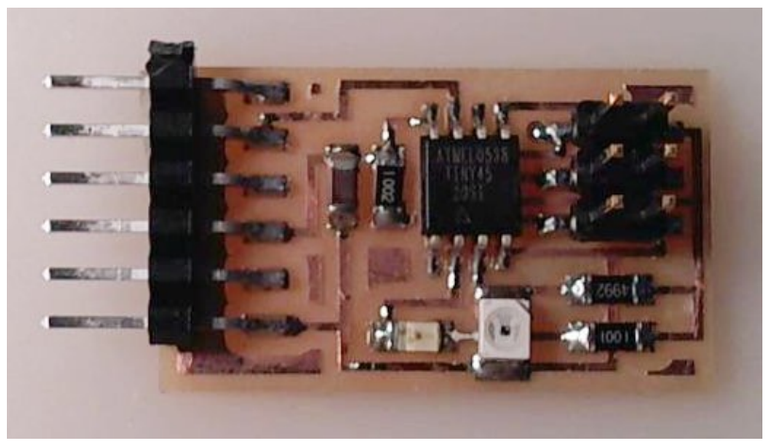
Input devices

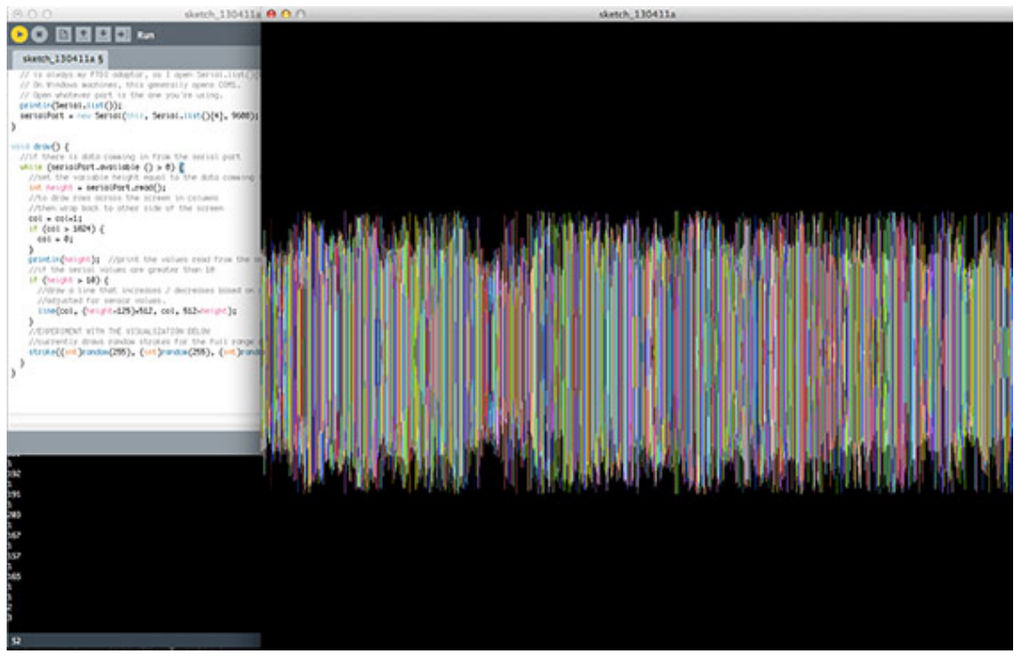
Output Devices

board



components





Embedded Programming

Networking and Communications
Interface and Application Programming

c

```c
//
//
// hello.reflect.45.c
//
// light reflection synchronous detection hello-world
//    9600 baud FTDI interface
//
// Neil Gershenfeld
// 10/25/12
//
// (c) Massachusetts Institute of Technology 2012
// Permission granted for experimental and personal use;
// license for commercial sale available from MIT.
//

#include <avr/io.h>
#include <util/delay.h>

#define output(directions,pin) (directions |= pin) // set port direction for output
#define set(port,pin) (port |= pin) // set port pin
#define clear(port,pin) (port &= (~pin)) // clear port pin
#define pin_test(pins,pin) (pins & pin) // test for port pin
#define bit_test(byte,bit) (byte & (1 << bit)) // test for bit set
#define bit_delay_time 102 // bit delay for 9600 with overhead
#define bit_delay() _delay_us(bit_delay_time) // RS232 bit delay
#define half_bit_delay() _delay_us(bit_delay_time/2) // RS232 half bit delay
#define char_delay() _delay_ms(10) // char delay

#define serial_port PORTB
#define serial_direction DDRB
#define serial_pin_out (1 << PB2)

#define led_port PORTB
#define led_direction DDRB
#define led_pin (1 << PB3)

#define nloop 100 // number of loops to accumulate

void put_char(volatile unsigned char *port, unsigned char pin, char txchar) {
   //
   // send character in txchar on port pin
   //    assumes line driver (inverts bits)
   //
   // start bit
   //
   clear(*port,pin);
   bit_delay();
   //
   // unrolled loop to write data bits
   //
   if bit_test(txchar,0)
     set(*port,pin);
   else
     clear(*port,pin);
   bit_delay();
   if bit_test(txchar,1)
     set(*port,pin);
```

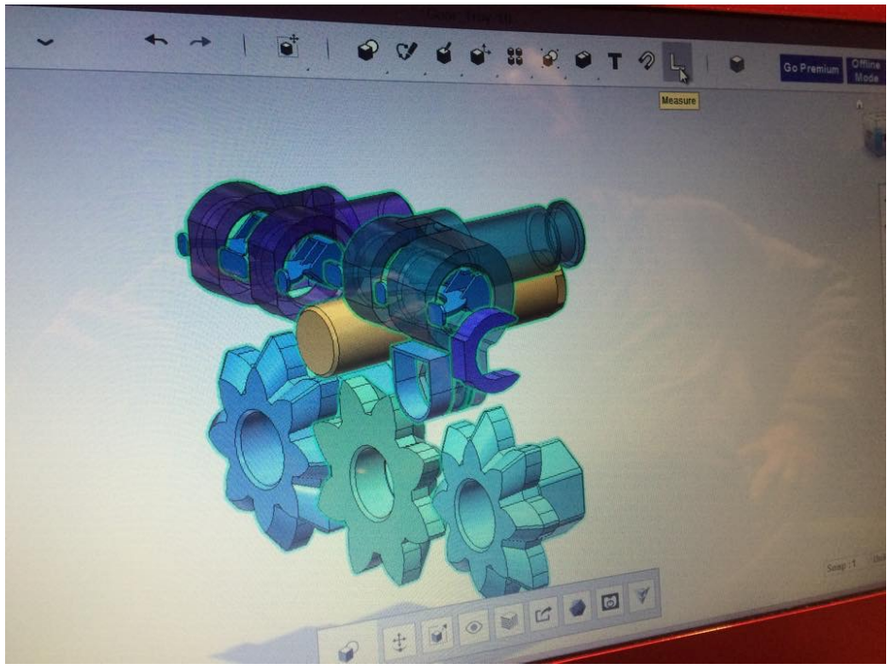
NORMAL hello.reflect.45.c

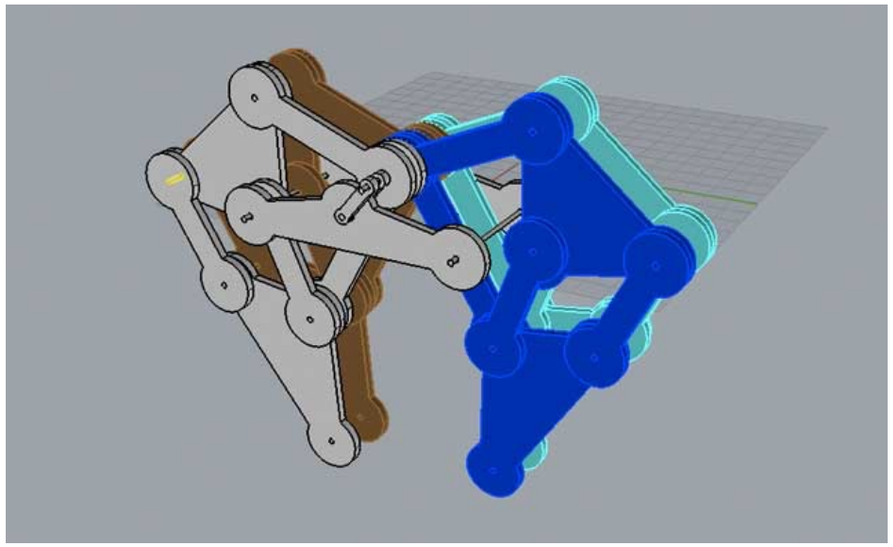
python

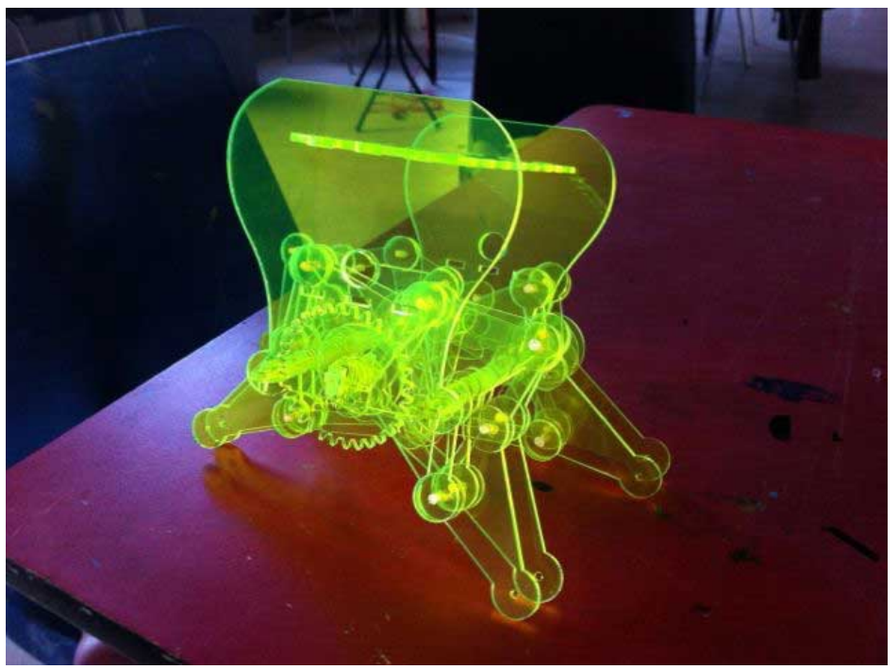
```python
#
# hello.reflect.45.py
#
# receive and display synchronous light reflection
# hello.light.45.py serial_port
#
# Neil Gershenfeld
# CBA MIT 10/25/12
#
# (c) Massachusetts Institute of Technology 2012
# Permission granted for experimental and personal use;
# license for commercial sale available from MIT
#

from Tkinter import *
import serial

WINDOW = 600 # window size
eps = 0.9 # filter time constant
filter = 0.0 # filtered value
nloop = 100.0 # number of loops accumulated
amp = 25.0 # difference amplitude

def idle(parent,canvas):
   global filter, eps
   #
   # idle routine
   #
   byte2 = 0
   byte3 = 0
   byte4 = 0
   ser.flush()
   while 1:
     #
     # find framing
     #
     byte1 = byte2
     byte2 = byte3
     byte3 = byte4
     byte4 = ord(ser.read())
     if ((byte1 == 1) & (byte2 == 2) & (byte3 == 3) & (byte4 == 4)):
       break
   on_low = ord(ser.read())
   on_high = ord(ser.read())
   on_value = (256*on_high + on_low)/nloop
   x = int(.25*WINDOW + (.9-.25)*WINDOW*on_value/1024.0)
   canvas.itemconfigure("text_on",text="on %.1f"%on_value)
   canvas.coords('rect1_on',.25*WINDOW,.05*WINDOW,x,.2*WINDOW)
   canvas.coords('rect2_on',x,.05*WINDOW,.9*WINDOW,.2*WINDOW)
   off_low = ord(ser.read())
   off_high = ord(ser.read())
   off_value = (256*off_high + off_low)/nloop
   x = int(.25*WINDOW + (.9-.25)*WINDOW*off_value/1024.0)
   canvas.itemconfigure("text_off",text="off %.1f"%off_value)
   canvas.coords('rect1_off',.25*WINDOW,.25*WINDOW,x,.4*WINDOW)
   canvas.coords('rect2_off',x,.25*WINDOW,.9*WINDOW,.4*WINDOW)
   filter = (1-eps)*filter + eps*amp*(on_value-off_value)
```

NORMAL hello.reflect.45.py

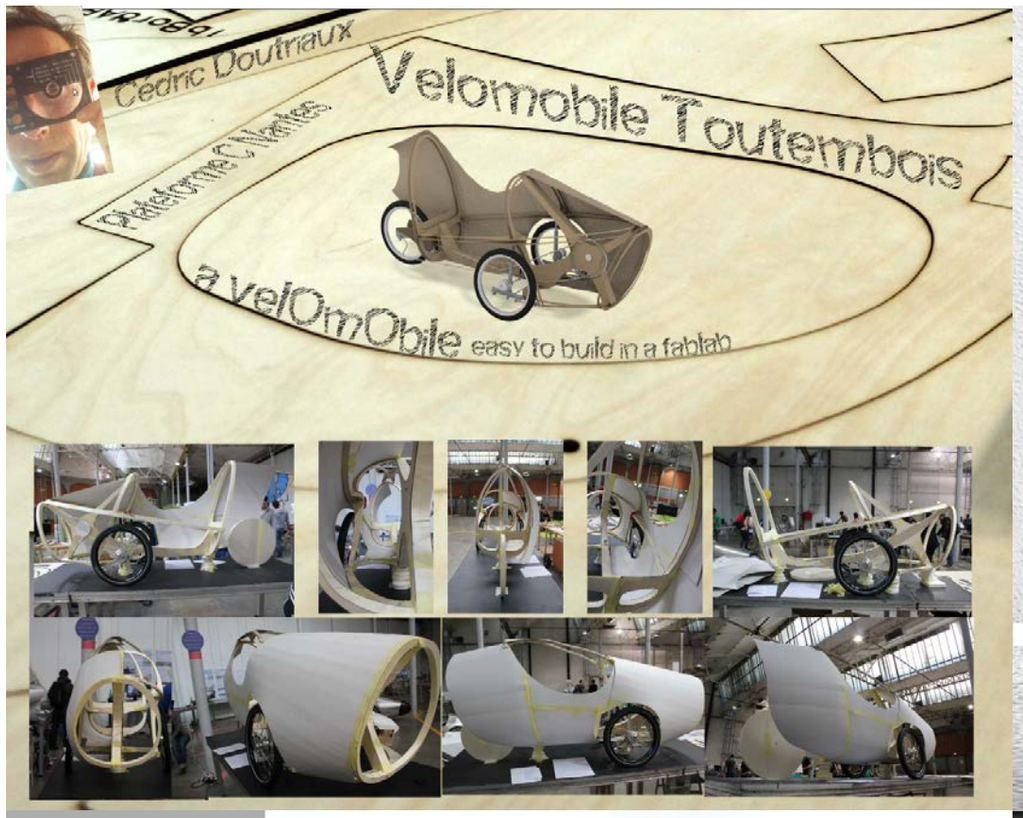
Mechanical Design and Machine Design

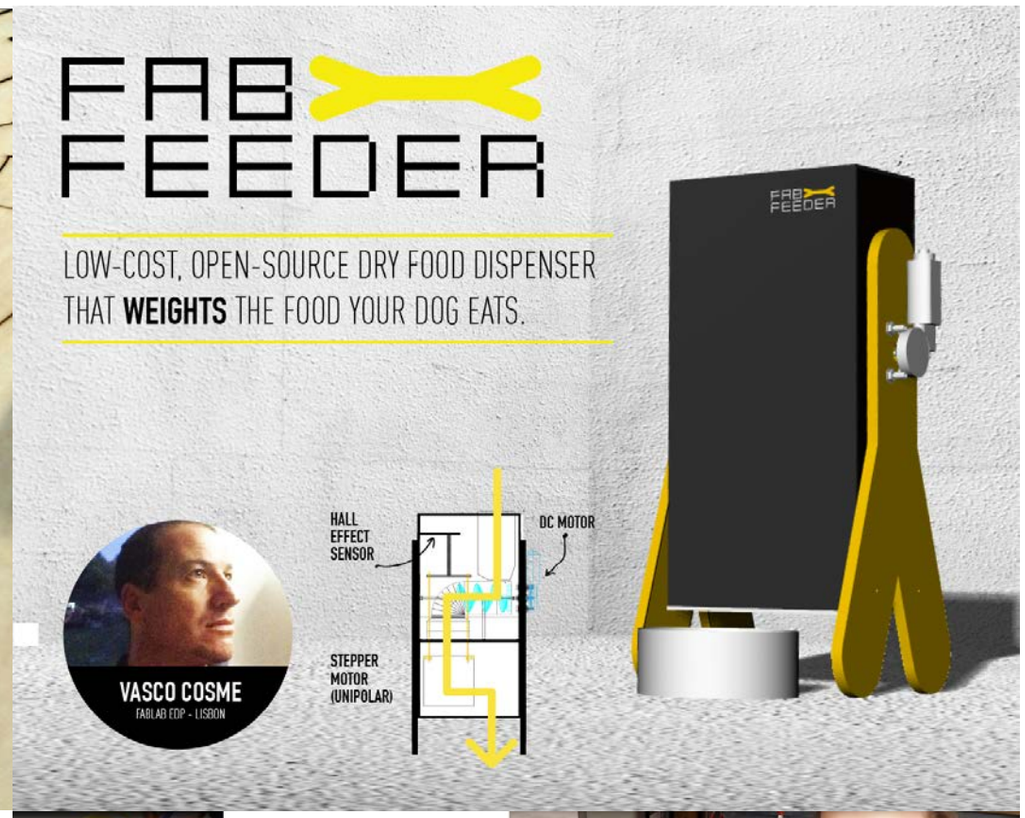


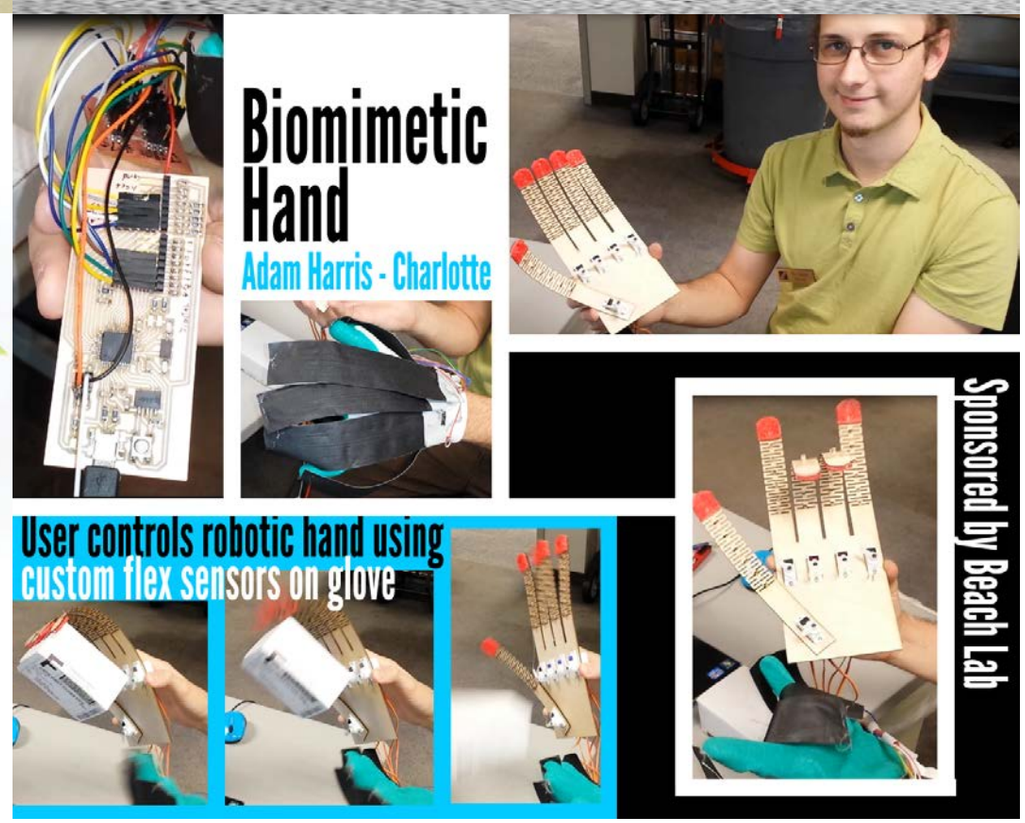




Final Project

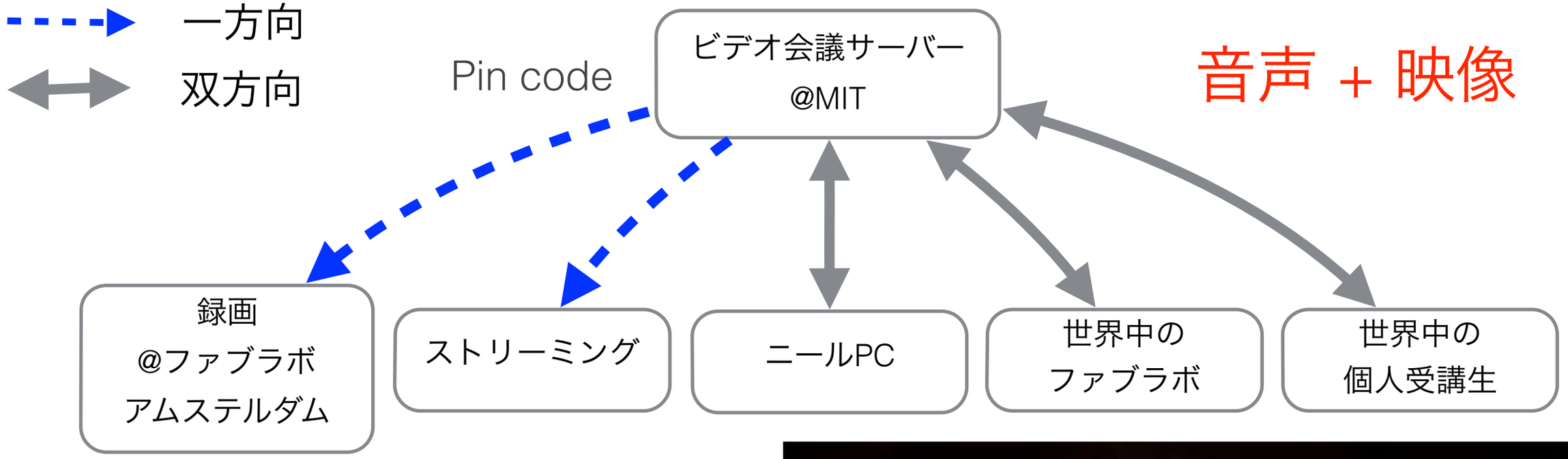






Fab Academyを
支える基盤技術

MCU - Multi Point Control Unit



ハードウェア

- Polycom
- Lifesize
- etc

ソフトウェア

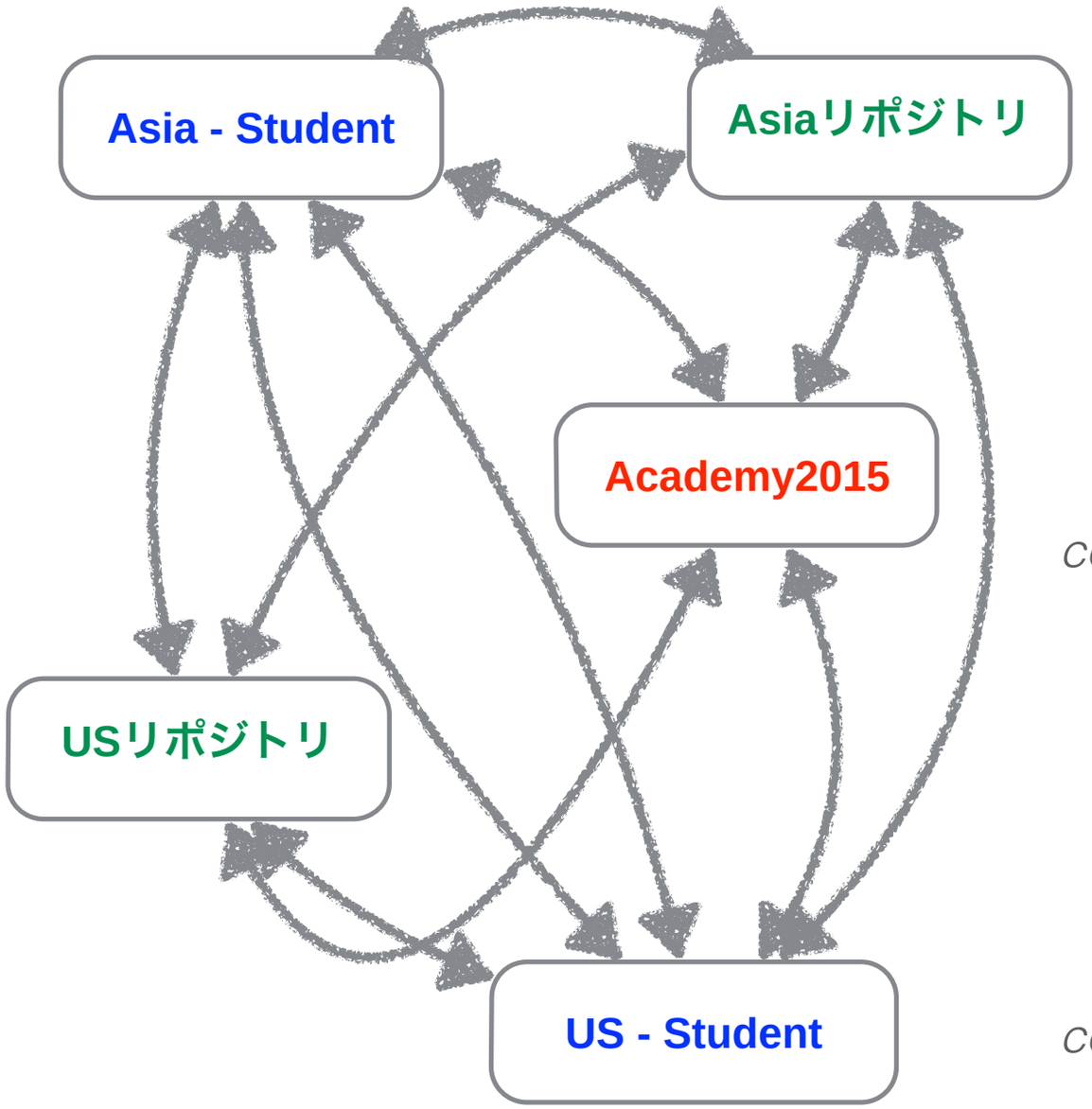
- Jabber Video
- Clear Sea
- Ekiga
- etc

ルール

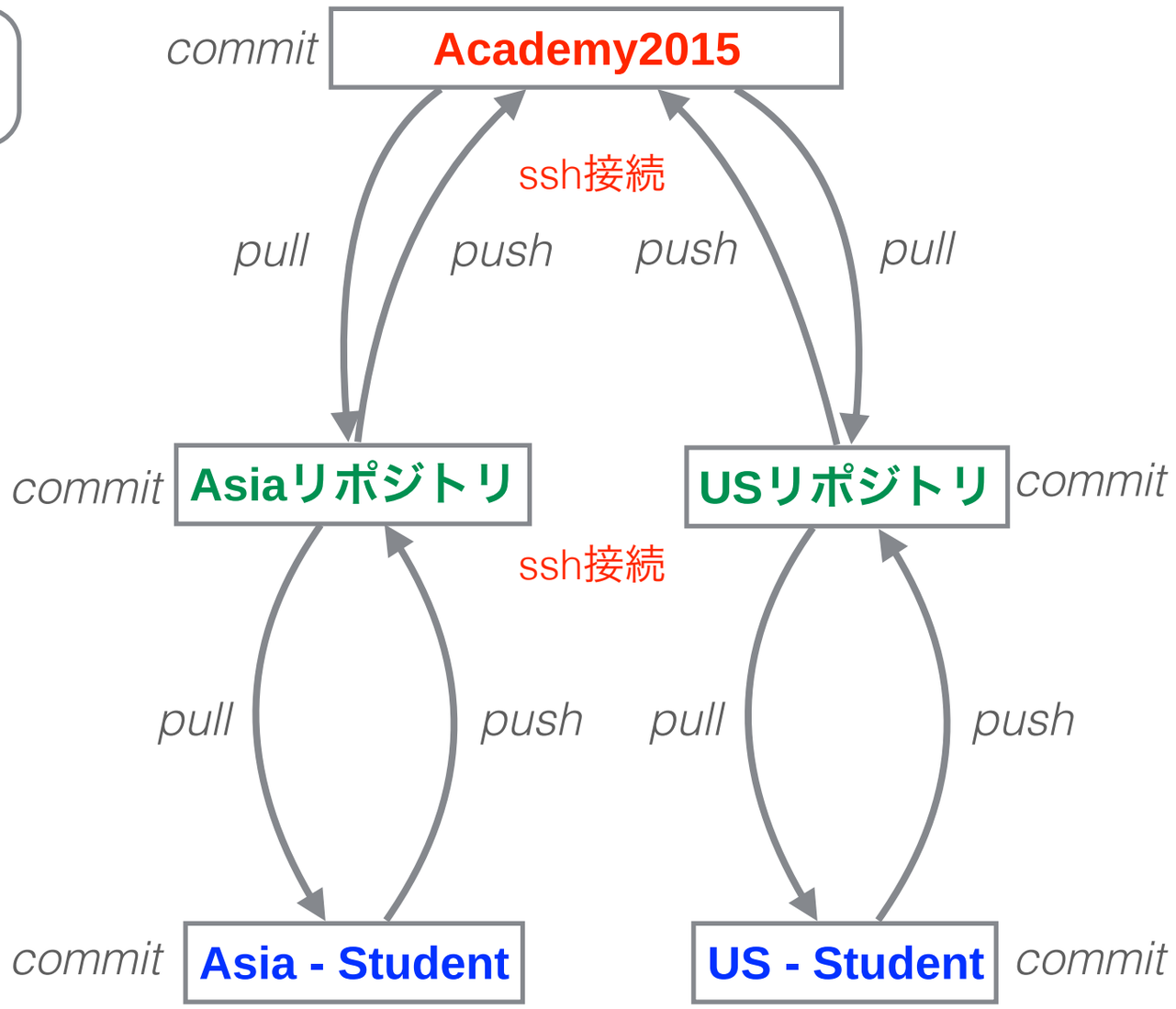
- 基本ミュート
- 回線1Mbps以上

Mercurial - 分散型バージョン管理システム

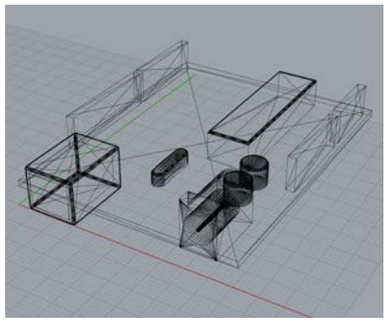
概念図

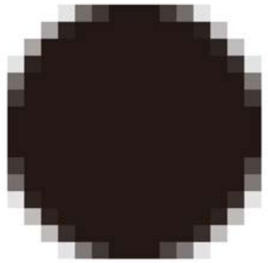


フロー



Fab Modules - 統合型CAMツール





STL (3D)

SVG (Vector)

PNG

ETC

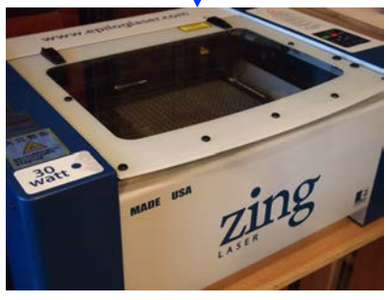
Fab Modules [Local or HTML5]

G-CODE

EPI

RML

DXF

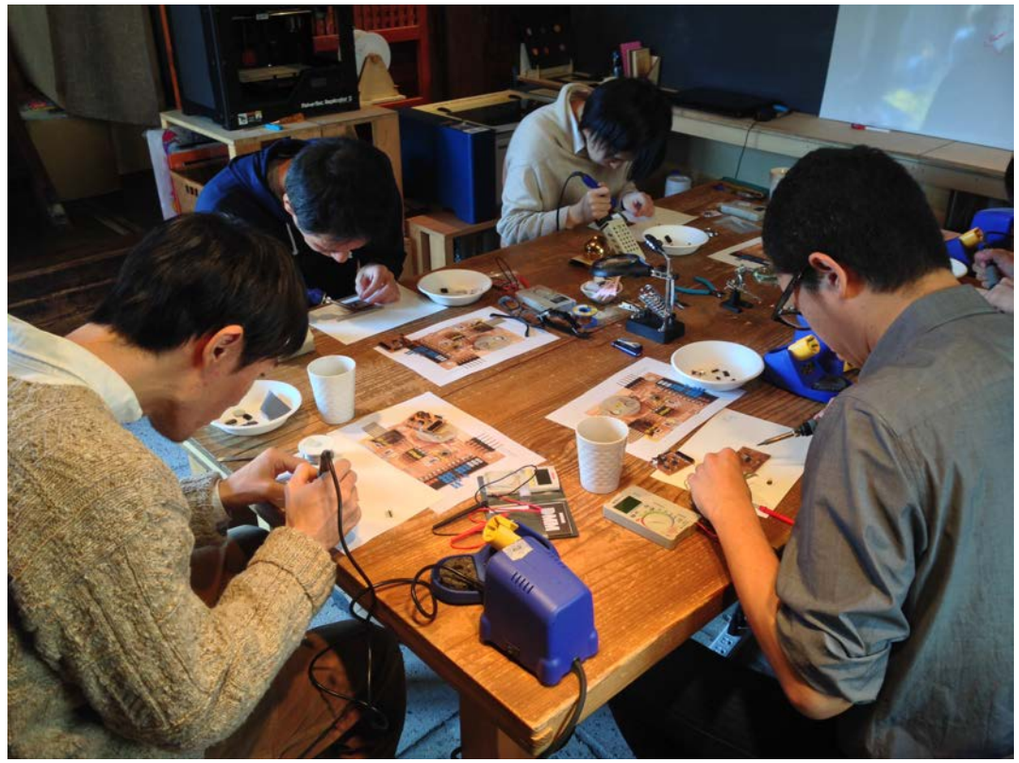


2015年 - 日本からのFab Academy参加

- 参加人数 : 4名
- バックグラウンド
  - 美術家 : ファブラボ北加賀屋
  - エンジニア : ファブラボ浜松
  - デザイナー : ファブラボ鎌倉
  - 大学研究員 : ファブラボ鎌倉

IoT Creator養成講座(鎌倉)

- 開催場所 : ファブラボ鎌倉
- 期間 : 10日間(3ヶ月)
- 人数 : 4名
- バックグラウンド :
  - 家電メーカー エンジニア
  - 通信会社 研究者
  - 印刷会社勤務
  - 経営コンサルタント
- 記録



IoT Creator養成講座(山口)

- 開催場所 : 山口情報芸術センター
- 期間 : 6日間
- 人数 : 6名
- バックグラウンド :
  - 大学生:専攻(建築)
  - 大学生:専攻(分子生物学)
  - 大学院生(教育学部)
  - ラジオDJ(哲学科卒業)
  - 木工作家(オートマタ)
  - WEBデザイナー
- 記録
  - チームA
  - チームB
  - チームC