

Multi Schrittmotor steuern

Spezialisierung / Professionalisierung

Su Gao

MA Multimedia Design

Burg Giebichenstein Kunsthochschule Halle

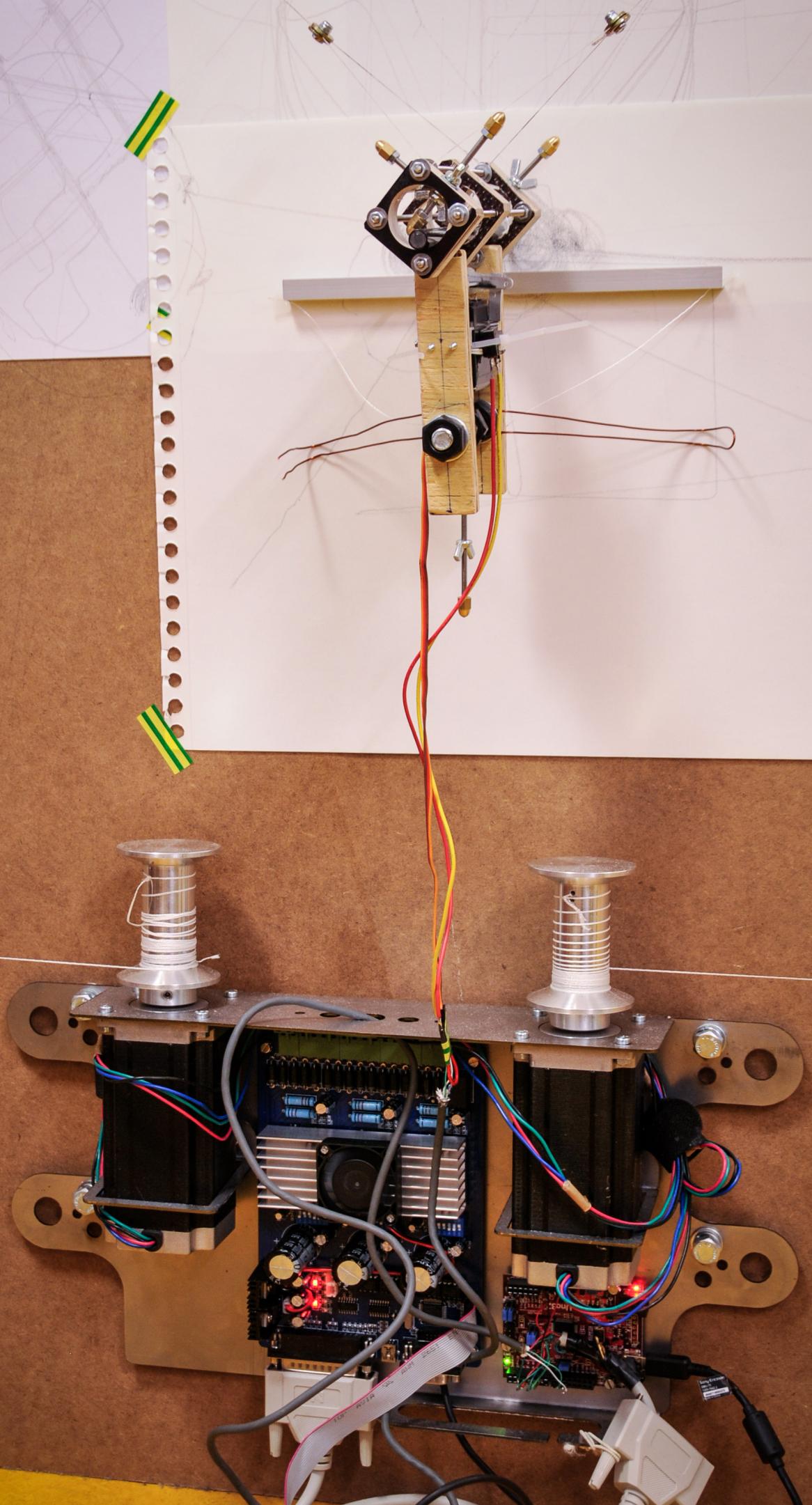
WS 2013/2014

天下万物生于有，有生于无。道生一，
一生二，二生三，三生万物。

——老子

(Alle in der Welt sind aus sein, Sein ist aus nichts. Eins ist von Dao geboren, zwei ist von eins geboren, drei ist von zwei geboren, alle der Welt sind von drei geboren.)

-Lao Zi



Inhaltverzeichnis

Warum Multi Schrittmotoren?	1
-Die Kunst von Programm -- singel Threading	1
-Multi Schrittmotoren bewegen	1
Elegant Zeichnen.....	2
Von svg zu G-Code.....	3
Penholder.....	4
Motor, Motor Treiber & Steuerboard	5
Software.....	7
Versuch des Drawbots.....	9
Ausblick	19

Warum Multi Schrittmotoren?

Die Kunst von Programm -- singel Threading

Trotzdem gibt es heutzutage in neuem Rechner mehrere „Core“ in CPU. Aber die Struktur der Software bleibt wie paar Zehnten Jahre bevor. Und zwar in einem bestimmten Zeitschritt würde eine und nur eine Zeile Code ausgeführt werden. Durch die Bemühung aller Menschen die in IT Bereich arbeiten ermöglicht Rechner Multi Auftrag machen zu können. Das Prinzip davon ist, dass der ganze Auftrag in sehr kleiner Schnitt zuerst geschnitten wird, dann bearbeitet der Rechner an einem Schnitt in einer sehr sehr kleinen Zeit Schnitt bis es fertig ist, dann an einem Schnitt, der zur anderen Auftrag gehört. Dadurch fühlt man, dass der Rechner parallele viele Auftrag machen dürfte.

Multi Schrittmotoren bewegen

Schrittmotor ist der besten Freund für alle Mechatroniker. Es kann durch drei Zeile Programmierung Code eine vordefiniert Winkel des Motors drehen(Pseudo Code: stepPin High, wart kurz, stepPin LOW). Das bietet man die Möglichkeit die Bewegung des Motors genau durch Programmierung Sprache beschreiben.

Weil es schon CAD (Computer aided Design) Software im Computer gibt, würde es schöner die CAD Datei mit Hilfe von der Computer steuerte Schrittmotoren direct als endgültige Produkte exportieren. In der vergangene paar Decaden hat man natürlich viele solche Systemen gebaut. Es gibt sogar

eine schöne Name davon - CAM (Computer aided Manufacture). Aber wie die Programme überhaupt gleichzeitig unterschiedliche Schrittmotoren steuern und wie die Anwendbarkeit des Methode in der Kunstbereich ist würde ein sehr interesante Thema damit zu beschäftigen.

Elegant Zeichnen

Wir habe uns an Drucker gewöhnt. Die unterschiede sind nur Marke, Schneller oder langsamer, größer oder kleiner, Monoton oder Farbig. Wenn wir „Druck“ Taste auf Bildschirm drucken, folgt es nur noch lauter oder leichter Geräusch von irgendwo der „Kiste“, dann spuckt es die Papier. Ich finde wir fehlen etwas noch. Wir lassen ein Künstler auf der Straße unsere Porträt zeichnen. Wir kaufen nicht nur die Ende Produkte, in diese fälle eine Zeichnung, sondern auch das Erlebnis. Bis heute kann ich noch erinnern wie die Fühle während der Porträtzeichnung viele Jahre bevor ist. Deswegen möchte ich mit Hilfe des Rechners und Schrittmotoren das fehlende Erlebnis wieder anbieten. Lässt Drucken genauso künstlerisch wie Porträtzeichnen. Natürlich öffne ich zuerst google um andere Ideen zu suchen. Dann entdecke ich viele verschiedenen Lösungen dafür. Das Project „Der Kritzler“, eine vertikale Plotter habe ich aufgefallen. Ich würde auch so was zu gestalten, bauen und nach meinen Wünsche verbessern.

Von svg zu G-Code

Ich spreche meine Idee mit Prof. Kraft. Sie habe die gleiche Lust darauf.

Eigentlich Sie haben schon ein Prototype nach „Der Kritzler“ gebaut. Sie schlagen mir vor, dass ich vielleicht das Roboter per „G-Code“ steuern lassen sondern nicht wie der Kritzler benutzt - svg. Es ist viel zu langsam.

In Wikipedia habe ich die Definition für svg und G-Code gefunden:

Scalable Vector Graphics (SVG) is an XML-based vector image format for two-dimensional graphics that has support for interactivity and animation. The SVG specification is an open standard developed by the World Wide Web Consortium (W3C) since 1999¹.

G-code is the common name for the most widely used numerical control (NC) programming language, which has many implementations. Used mainly in automation, it is part of computer-aided engineering. G-code is sometimes called G programming language. The first implementation of high level numerical control programming language was developed at the MIT Servomechanisms Laboratory in the late 1950s².

Nur nach der Definition kann man wissen, dass G-Code länger Geschichte hat und besser für Automation gerichtet. In weitere Recherche finde ich wo

¹ http://en.wikipedia.org/wiki/Scalable_Vector_Graphics

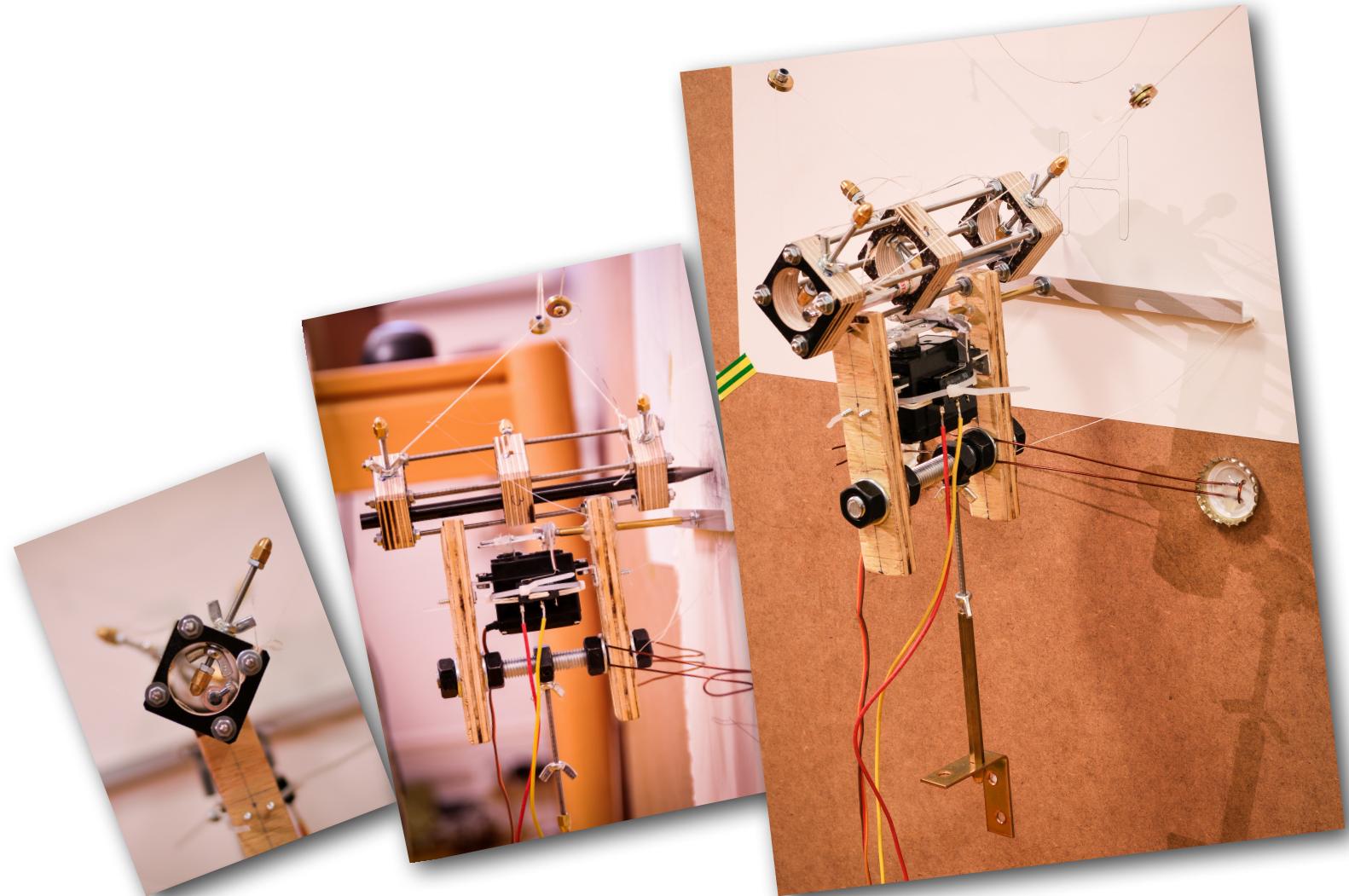
² <http://en.wikipedia.org/wiki/G-code>

die Vorteile (auch die Nachteil von svg methode) steht. Wenn eine Bilde in G-Code Kompiliert, beschreibt diese Sprache dieses Bild mit Linien und Arc. Bei svg Methode kann nur Linen Befehle gespeichert und transportiert werden. Das macht schon riesige Unterschied, weil ein Bilde normalerweise aus mehr Arc als Linien besteht. Das bedeutet durch G-Code Methode kann man mit viel weniger Code die gleiche Bild zeichnen. Das Ziel - Schneller zu zeichnen hat theoretische Möglichkeiten.

Penholder

Die Stichwörter für das Penholder sollte: Stift festhalten, präzis, einfach zu einstellen und stabile (Weil es ganzen Zeit bewegt werden muss).

In der praktische gestaltung habe ich Holz als haupt Material benutzt, weil es stabile und auch einfach zu bearbeiten ist. Da das Penholder nur durch zwei Seile gehängt wird, sollte es sich relative größer Gewicht hat, dadurch die Seile richtig ausdehnen und die eigenes Figur gut halten kann. Dafür setze ich eine 11cm lang M10 Gewindestang mit fünf M10 Mutter in der unten Teile von der Penholder. Um den Pen von der Papier hochheben oder legen zu können, setze ich ein Servo. Ich lasse es nicht direct mit Papier kontaktieren sondern durch eine Mechanismus es ermöglichen. Die Vorteile ist, dass die ganzen Penholder nicht während „Pen up“ von einer Punkt nach andere Punkt bewegt wird und dann „Pen down“ sich ein Stückchen drehen zu passieren. Diese Servo ist nicht ein „Position Servo“ sondern ein Servo, die durch PWM Signal drehen Geschwindigkeit und Richtung einstellen lassen zu dürfen. Damit



kann ich die genaue Geschwindigkeit von Stift Heben und legen zu einstellen.

Damit die Stift nicht „Bump“ wenn es erst gelegt wird. Ich fine tune das Design immer weiter während der Entwicklung und auch Wahnsinn viele Spaß damit gemacht.

Motor, Motor Treiber & Steuerboard

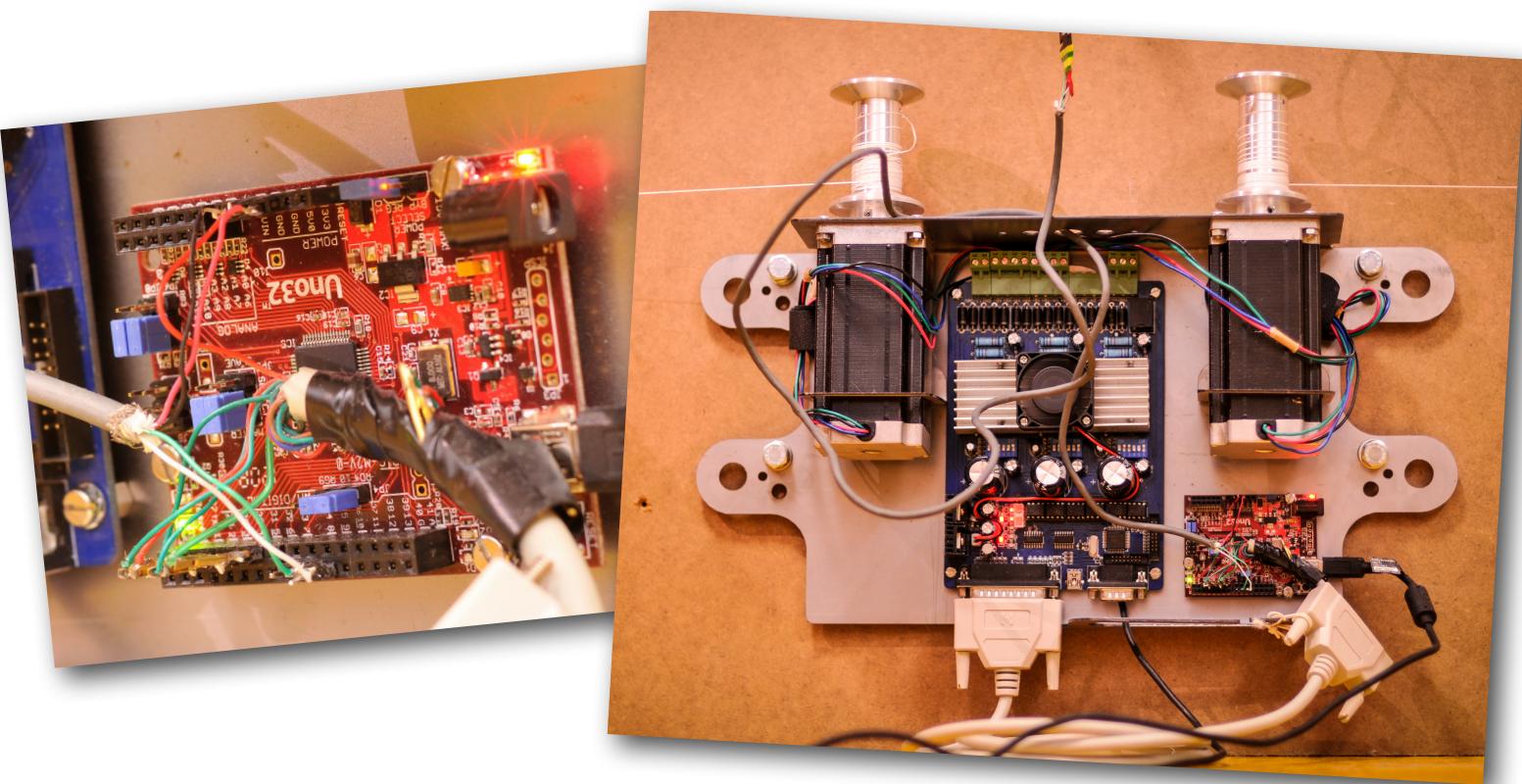
Ich habe Glück. Prof. Kraft bieten mir für das Projekt drei NEMA 23 Schrittmotoren (200 schritt pro Umdrehung) und auch ein semiprofi Schrittmotor Treiber dafür. Die erst Eindruck davon ist alle sind riesig. Um diese Neue Situation anzupassen, baue ich die Motoren nicht wie „Der Kritzler“ sonder zusammen mit dem Motor Treiber und Mikrokontroller Board zusammen in ein gelasert Gehäuse. Vielgleichen mit die Idee von

Kritzler spart ich vier fach Kabeln verlangt nur die Seile, die viel mehr billiger ist. Es lässt das Design noch robuster.

Auf das Arduino Board ist ein 8 bit AVR RISC MCU, die relative nicht so fähig für floating point rechnen ist. In unsere Anwendung gibt es aber echt viele floating point operation und auch viele trigonometrische Umrechnungen.

Deswegen habe ich ChipKIT UNO32 board ausgewählt, weil es ein 32bit board ist. Die Operationsgeschwindigkeit ist auch fünf fach schneller als ein standard Arduino (80MHz). In der später Versuch überzeugt es mich seine Geschwindigkeit. Nennenswert diese MikroKontrollers operations Voltage ist auf 3.3V, Aber das LPT Port, das das Motor Treiber Board benutzt, ist 5V.

Nach der Versuch kennen wir die 3.3V Signal von dem Mikrokontroller kann auch ohne Problem das Board steuern.

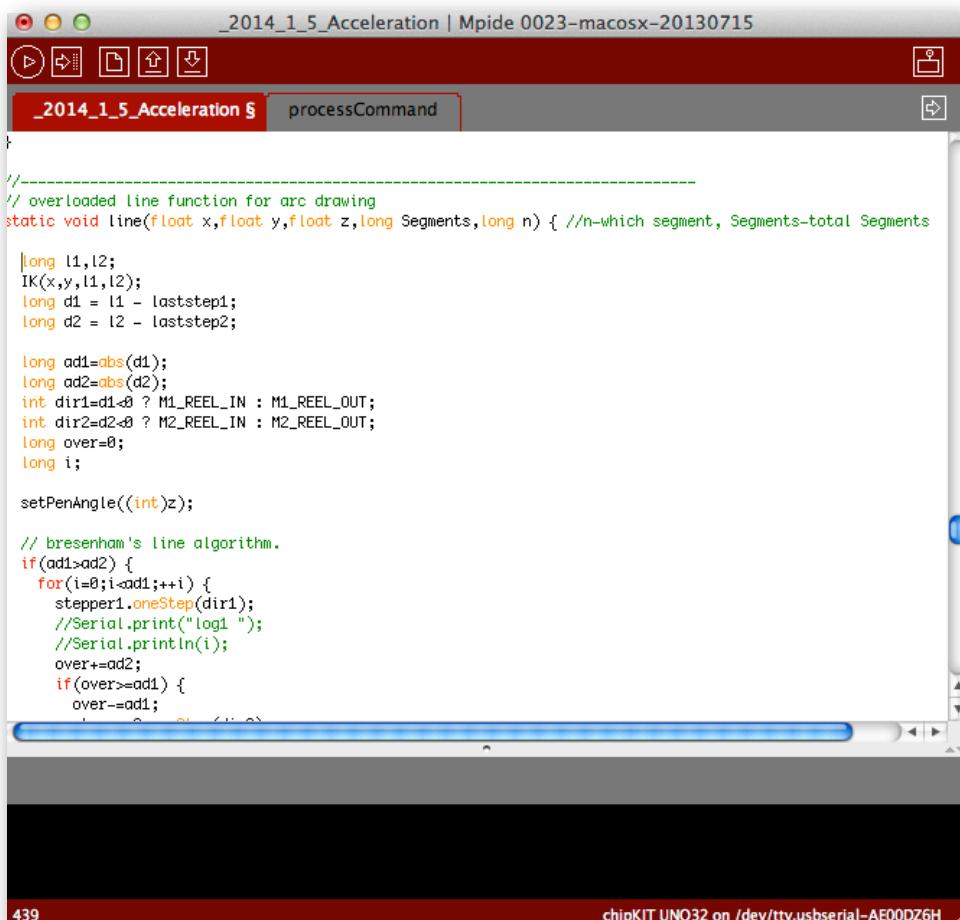


Software

Außer das Projekt Hektor ist fast alle andere opensource. Ich konzentriere mich auf „der Kritzler“ und von Dan Royer gemacht Projekt „Makelangelo“.

Ich greife alle was gut sind. Schreibe Die um, um die Software zu mein Hardware zu passen. Am Ende probiert ich selbst entwickelte Motor Beschleunigung Funktion. Es funktioniert gut aber die arc zeichnen Qualität ist nicht so perfekt. Es braucht noch zu verbessern.

Ich habe auch meine eigene Software für Drawbot entwickelt. Grundsätzlich ist es ein „G-Code Sender“, um schon kompilierte G-Code file nach Drawbot zu senden. Gleichzeitig kann es auch das drawbot als eine typische Schreibmaschine steuern. Man tippt irgend welche Wort ein, und Drawbot schreibt es mit bestimmte Font auf dem Papier auf.



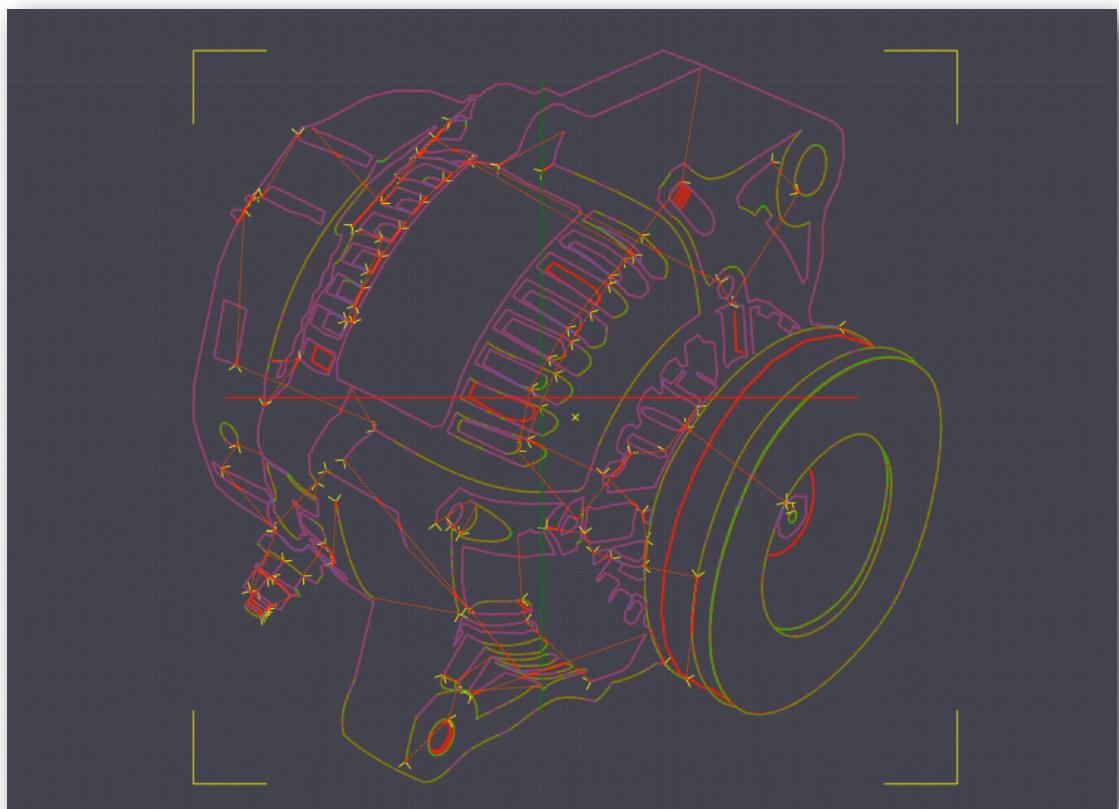
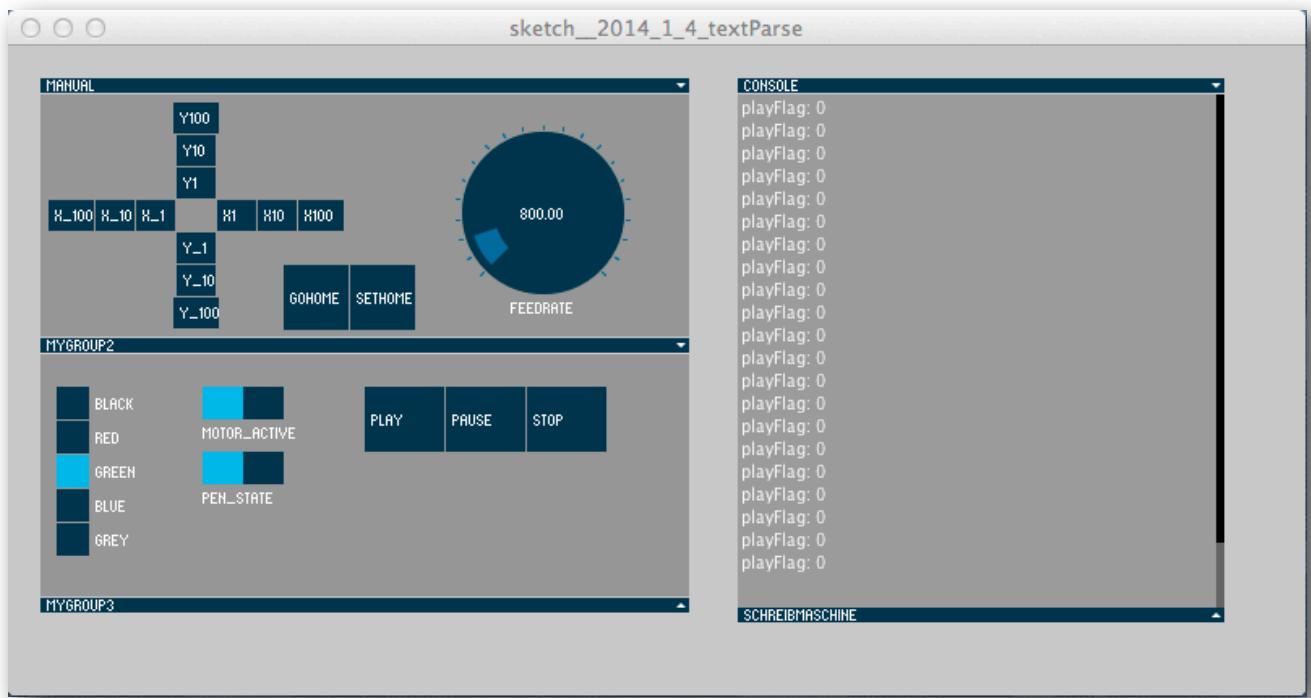
```
_2014_1_5_Acceleration | Mpide 0023-macosx-20130715
processCommand

// overlaoded line function for arc drawing
static void line(float x, float y, float z, long Segments, long n) { //n=which segment, Segments=total Segments
    long l1,l2;
    IK(x,y,l1,l2);
    long d1 = l1 - laststep1;
    long d2 = l2 - laststep2;

    long ad1=abs(d1);
    long ad2=abs(d2);
    int dir1=d1<0 ? M1_REEL_IN : M1_REEL_OUT;
    int dir2=d2<0 ? M2_REEL_IN : M2_REEL_OUT;
    long over=0;
    long i;

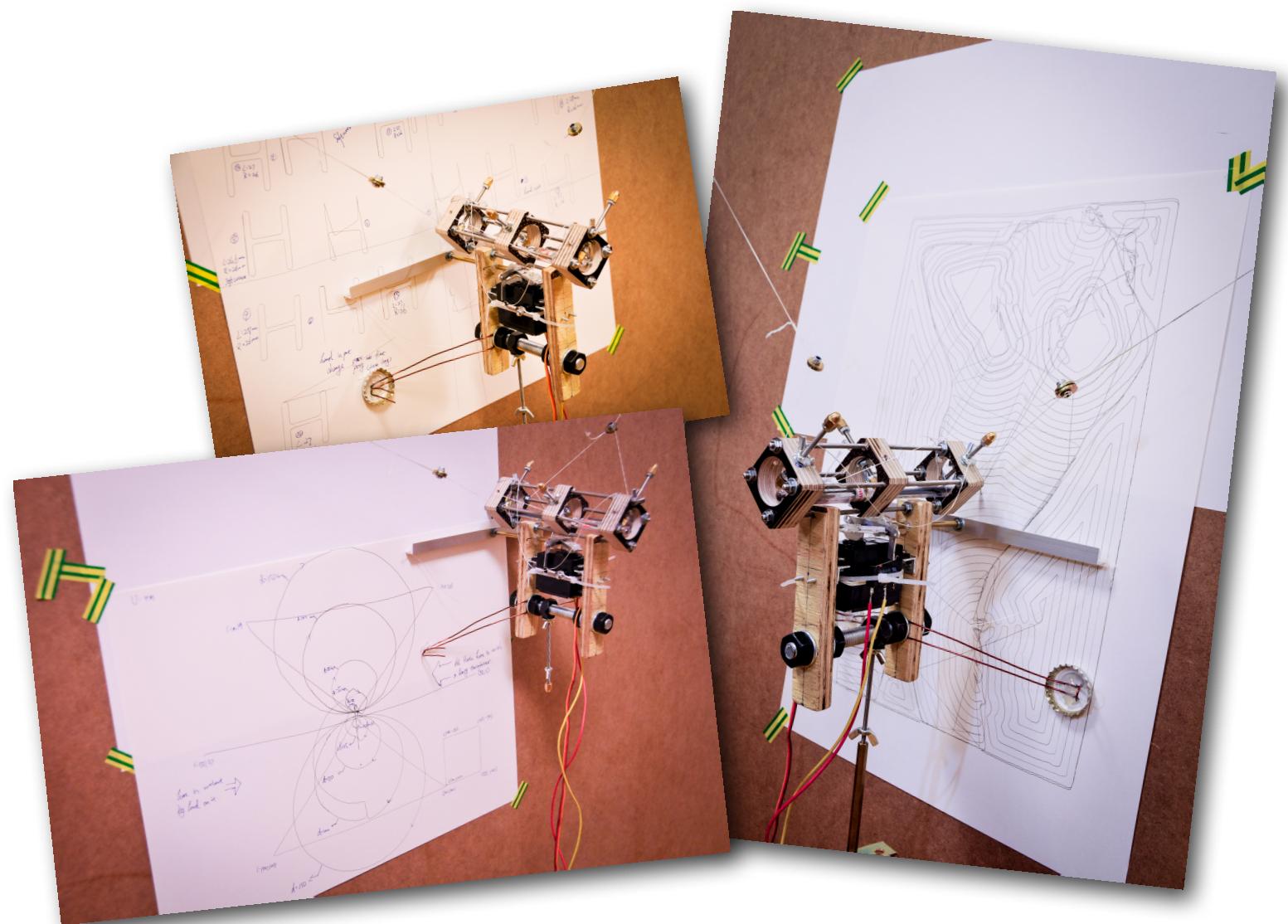
    setPenAngle((int)z);

    // bresenham's line algorithm.
    if(ad1>ad2) {
        for(i=0;i<ad1;+i) {
            stepper1.oneStep(dir1);
            //Serial.print("log1 ");
            //Serial.println(i);
            over+=ad2;
            if(over>=ad1) {
                over-=ad1;
            }
        }
    }
}
```

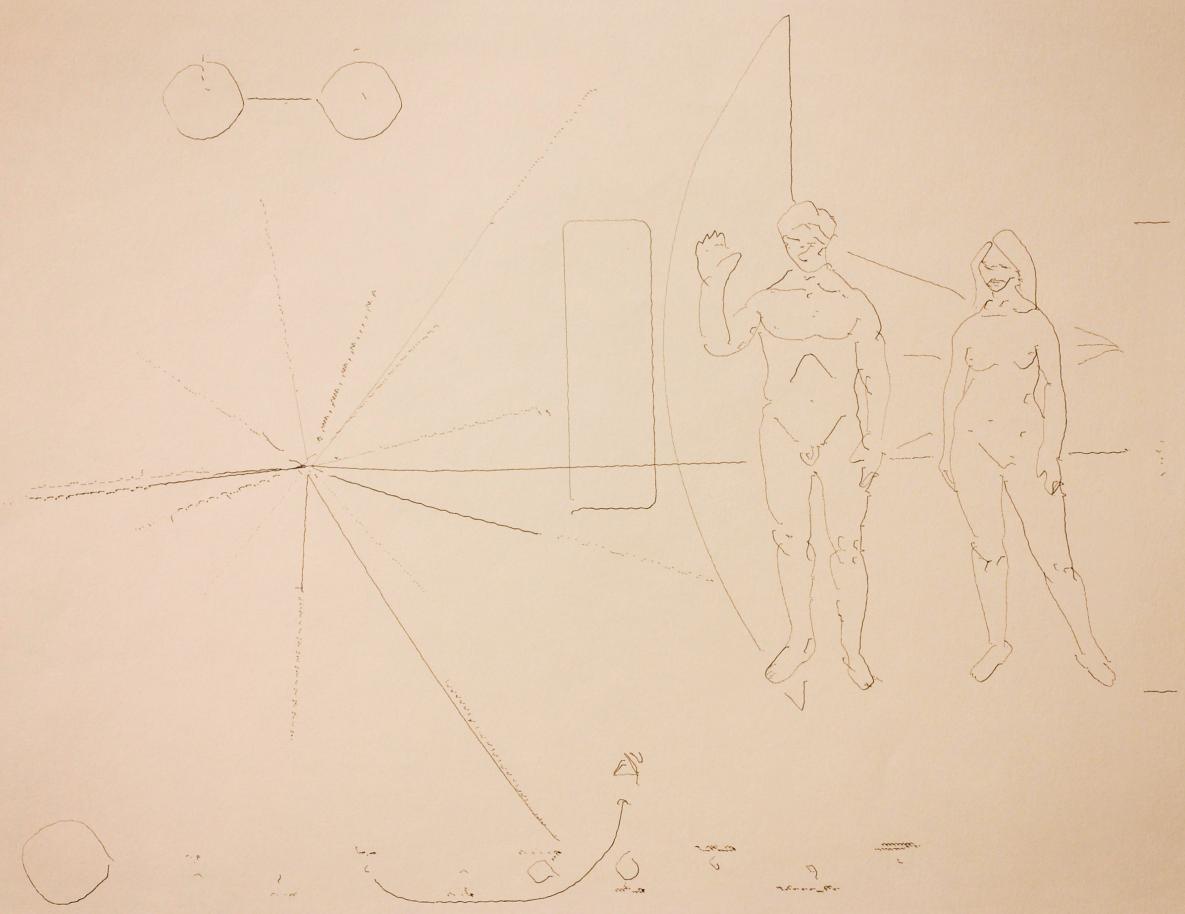


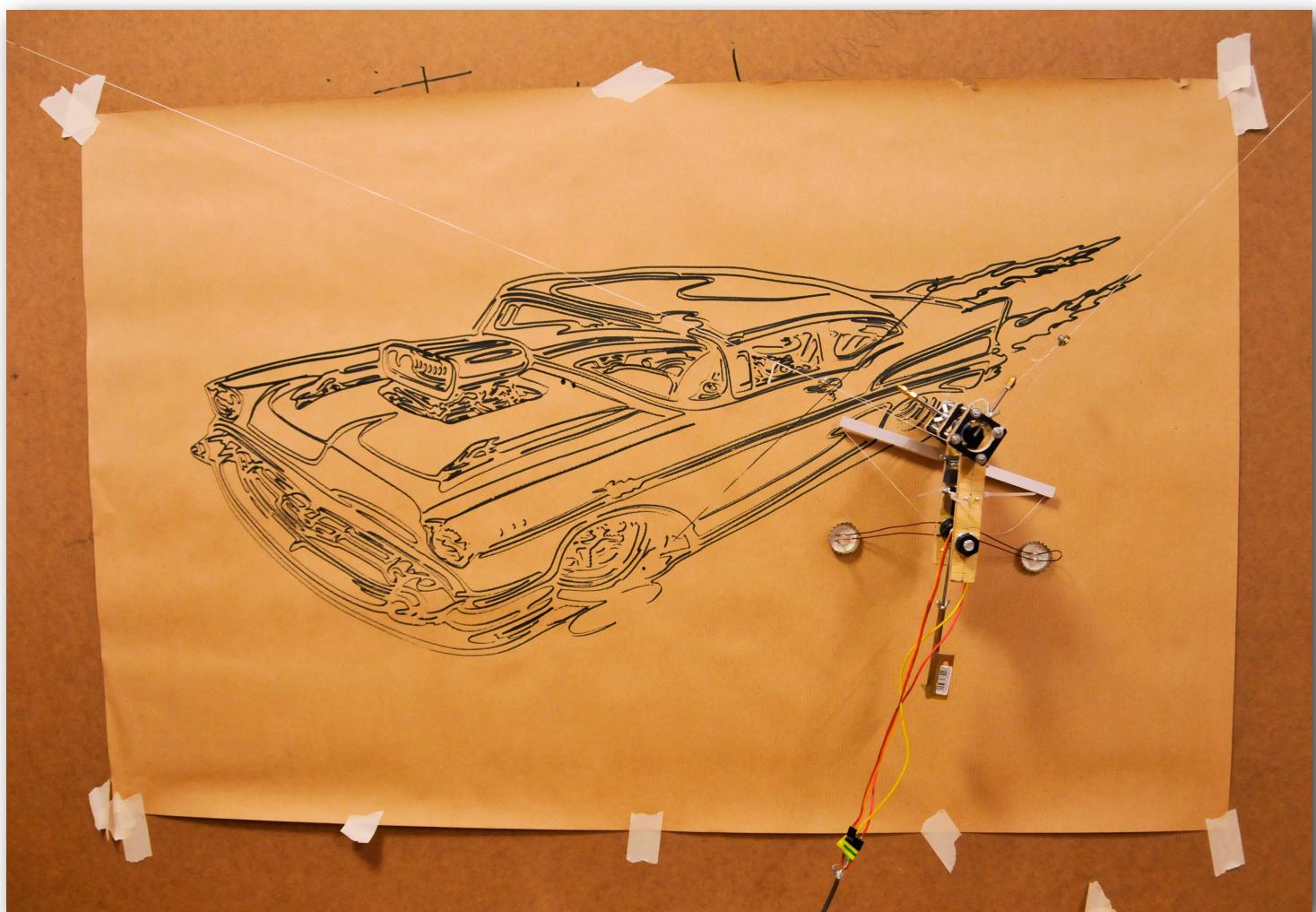
Versuch des Drawbots

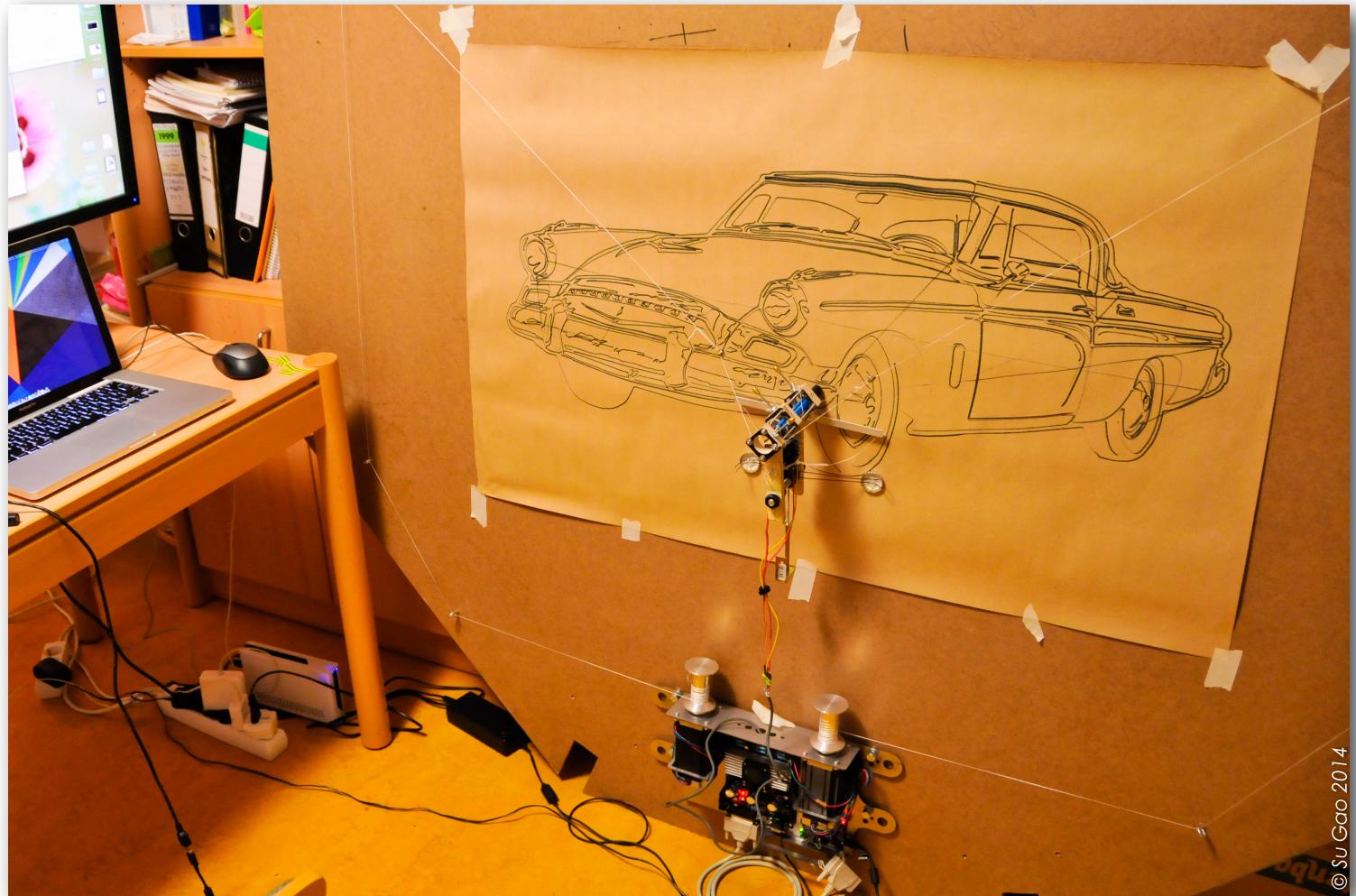
Anfang probiere ich das Drawbot auf A3 Format Papier zu zeichnen. Dann wechsel ich sehr schnell auf größere Format: 70 x 110 cm. Der Stift wechselt ich auch von Kuli zu Edding Marker.



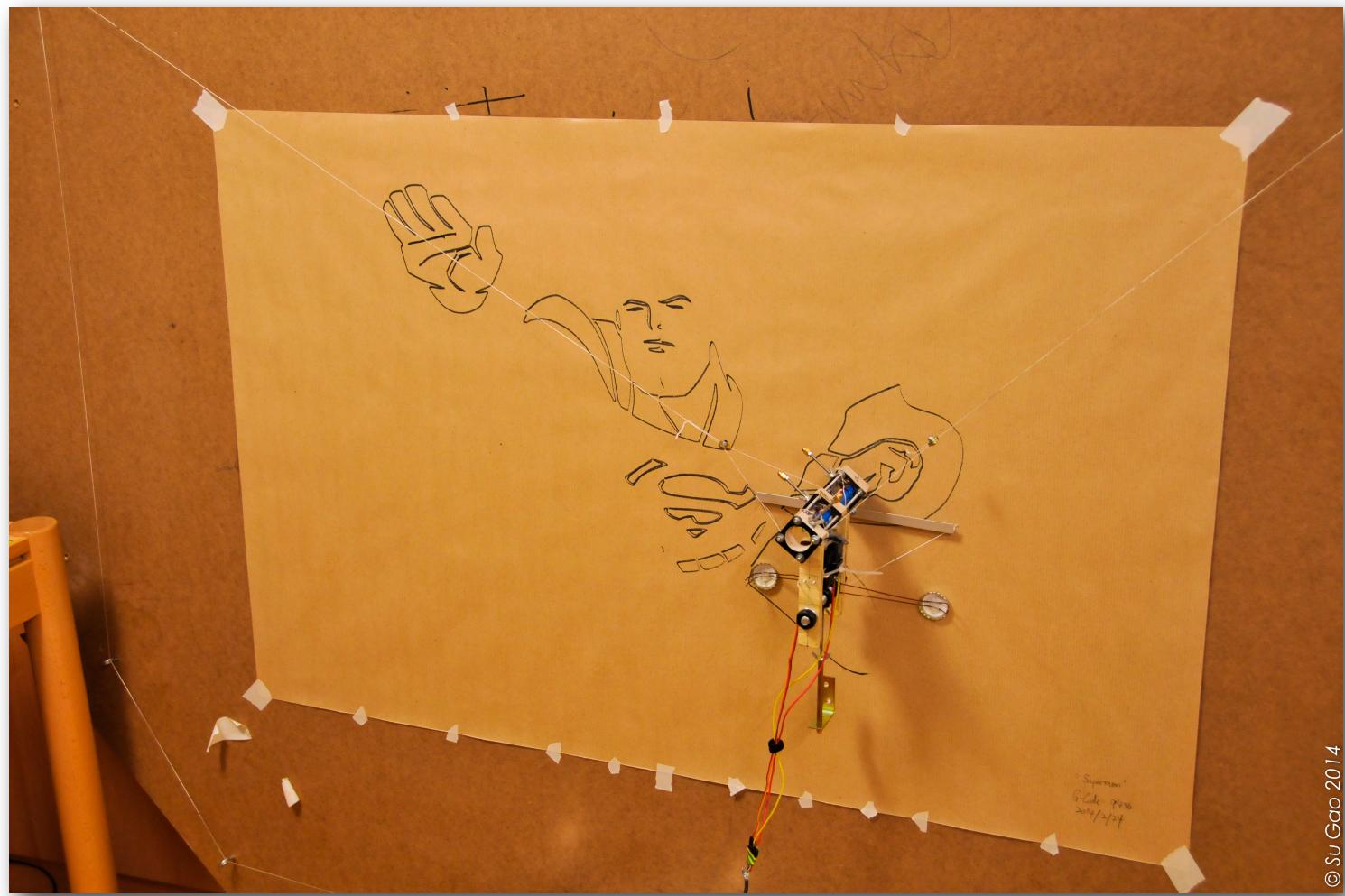




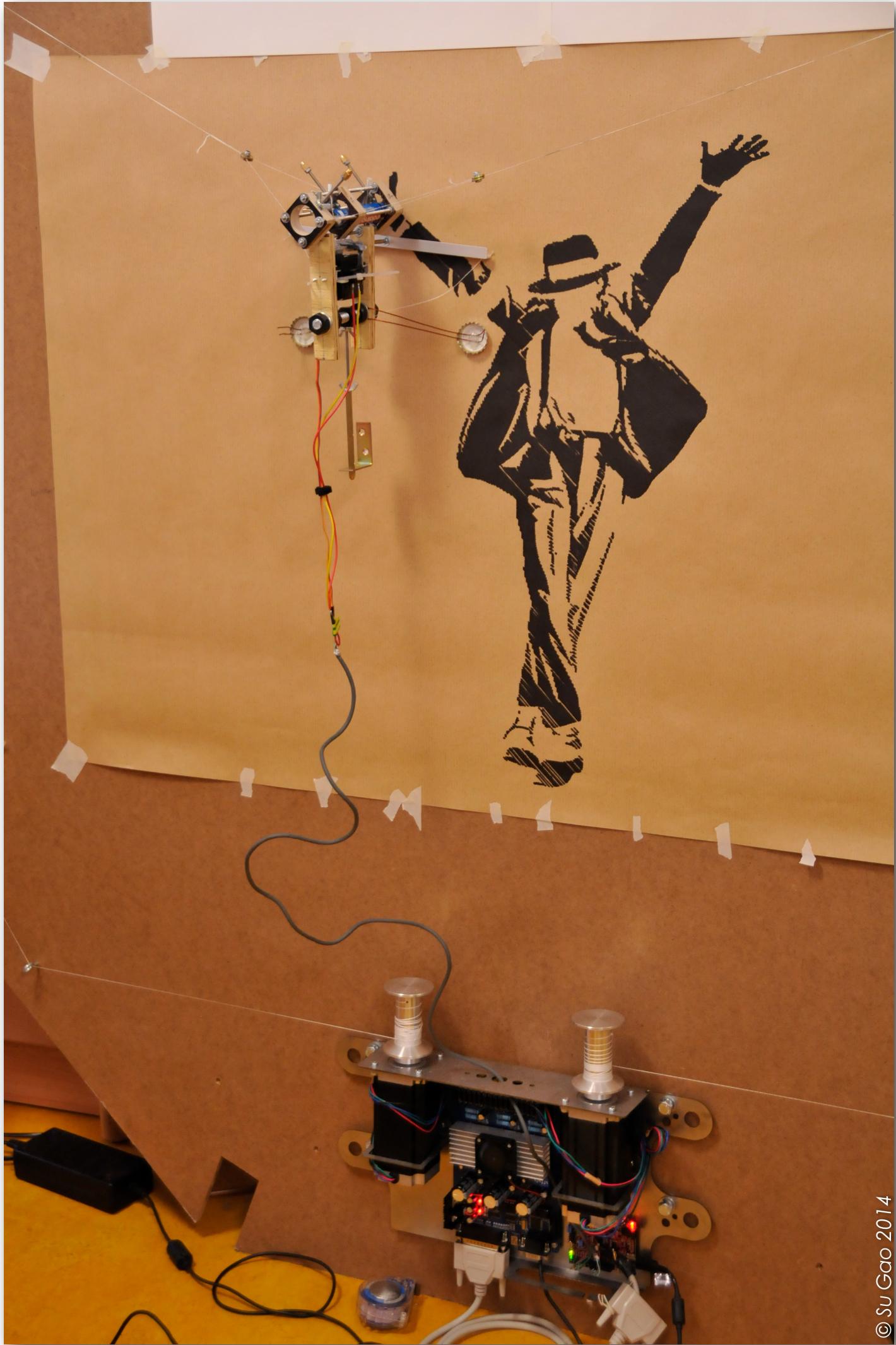




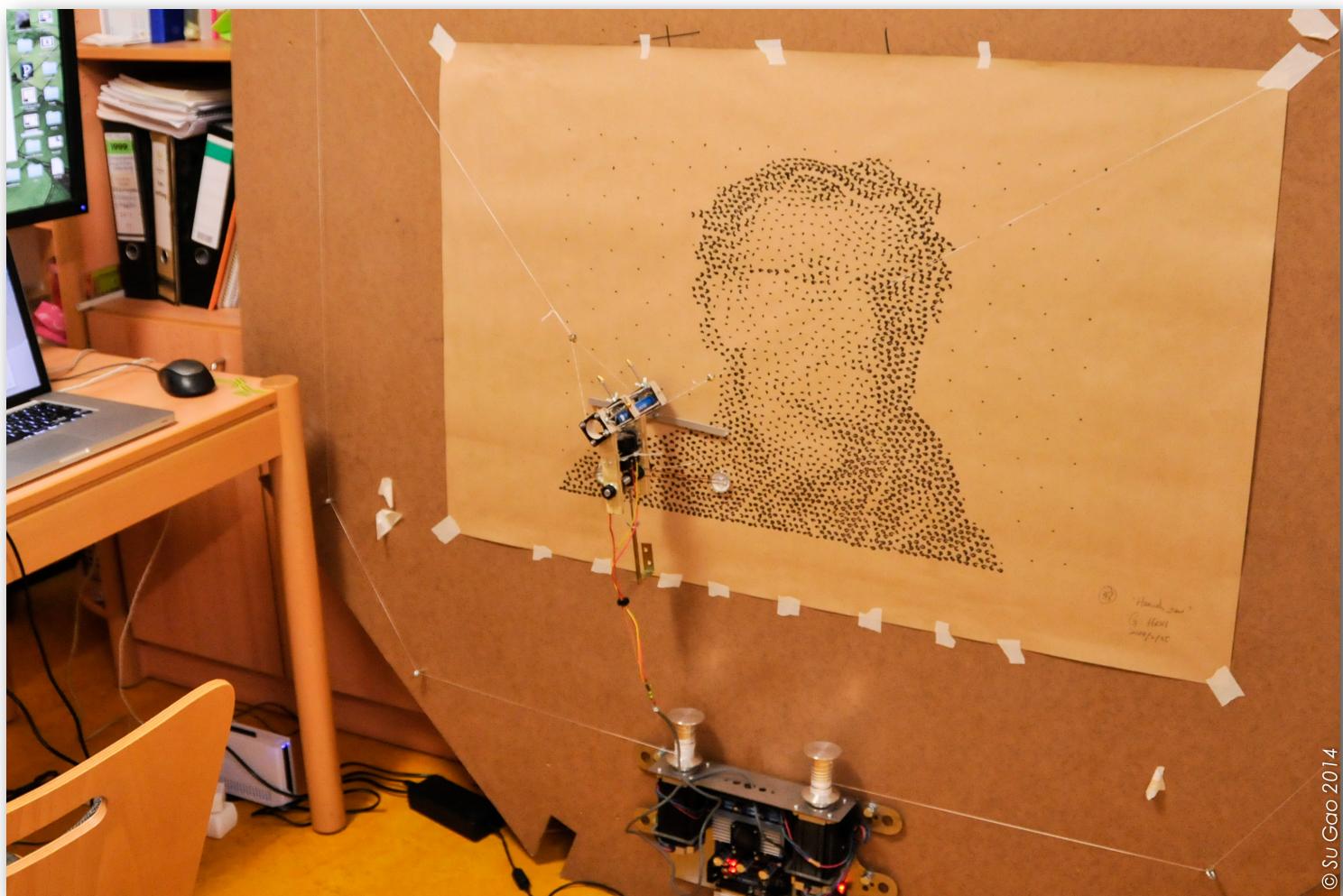
© Su Gao 2014



© Su Gao 2014



© Su Gao 2014



© SJ Gao 2014



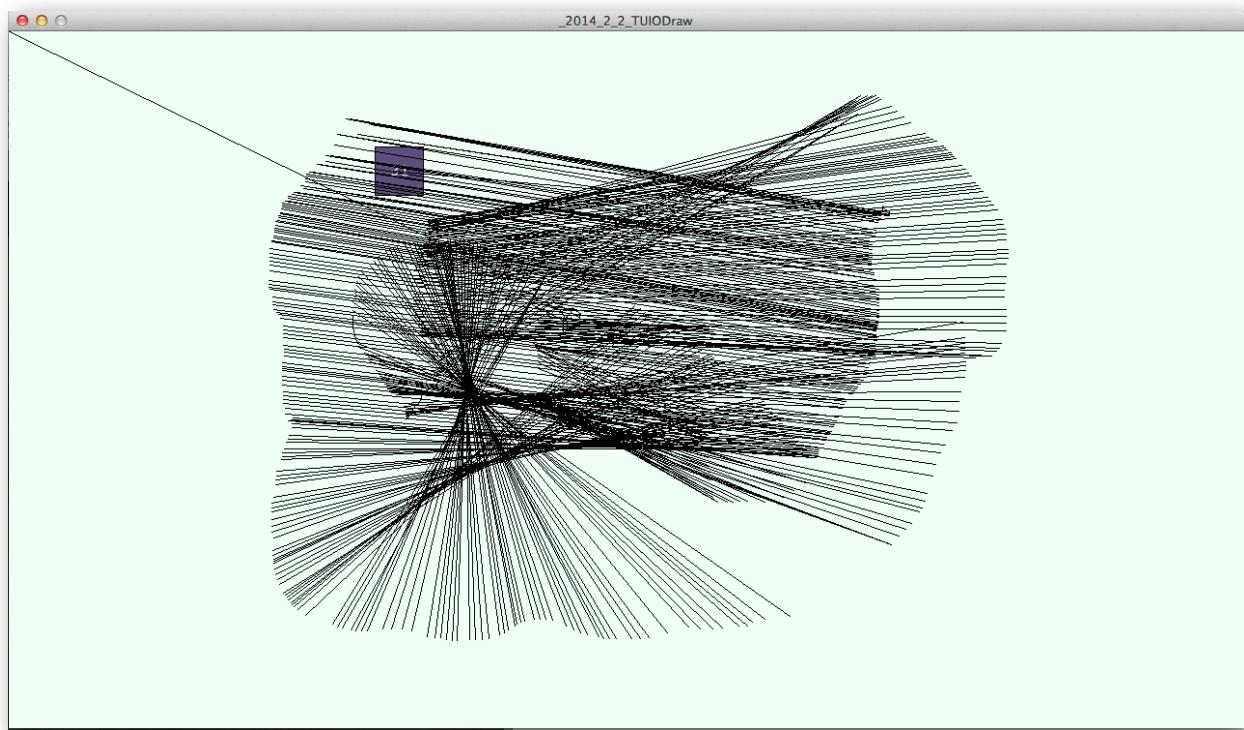
© Su Gao 2014



© Su Gao 2014

Ausblick

Ich habe viele Zeit mit dem Drawbot verbracht. Es gibt noch viele Möglichkeit würde ich gern probieren. Wie zum Beispiel, habe ich schon ein klein code in Processing geschrieben, durch das das Spur von dem Stift in meiner bewegenden Hand als G-Code umwandeln, speichern und auch „live“ mit dem Drawbot gezeichnet werden. Weil die Spur-Tracing-Software (reactIVision) Multi „Fiducial“ (ein besonderes Symbol) kennen kann. Darum zu programmieren würde man noch unschätzbar Möglichkeiten entdecken.



Darüber hinaus habe ich auch geplant, dass ich mehrere „bitmap scanner“ entwickeln. Zur Zeit habe ich nur die schräg Linien scanner entwickelt, mit dem man ein bitmap (wie zum Beispiel .jpg, .png, .bmp u.s.w) mit schräg

Linien die schwarze Teile von dem Bild bedecken kann (wie das Zebra und Michel Jackson). Nach dem Prinzip kann man auch „arc bitmap scanner“ oder „Runde bitmap scanner“ sogar irgend welche Form als scanner Form das bitmap scanner und als G-Code exportieren.

Die Erfahrung vom diesem Projekt ist sehr schön. Die Technik, die ich davon lernte, sollte große Rolle in meinem vorkommenden Projekt spielen. Ich bedanke mich an meimem Betreuer Professor Manfred Kraft. Sie haben viele Beratung und Materialien Unterstützung mir gebietet. Und auch vielen Danke für die Hilfe und Beratung an Tom Hanke und Martin Schwandt. Sie boten mir ein sehr cooles Praktikum, von dem ich unglaublich viel Erfahrung gelernt haben.

