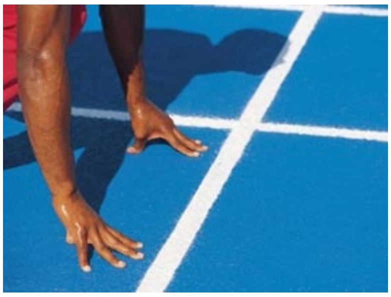
maXbox Starter 38



Start with a 3D Printing Lab

1.1 We need an API

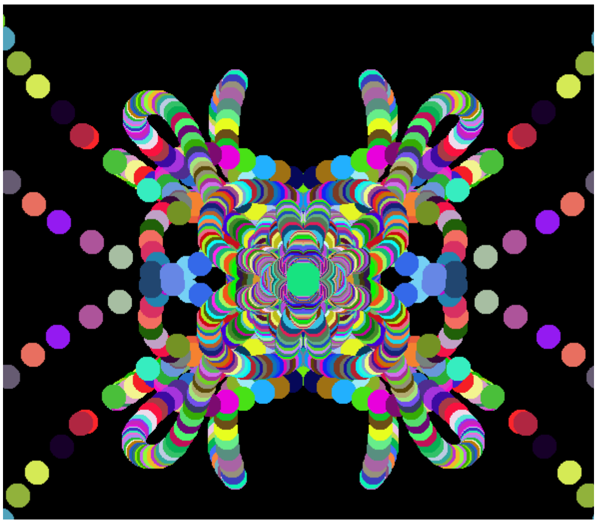
3D printing or additive physical manufacturing is a process of making three dimensional solid objects from a digital file like a CAD Stereo lithography (SLA). The topic is huge and the applications include research (so in our school), design visualization, prototyping/CAD/CAM, metal and box casting, architecture, education, geographical topology, healthcare and also entertainment, research and amusement.

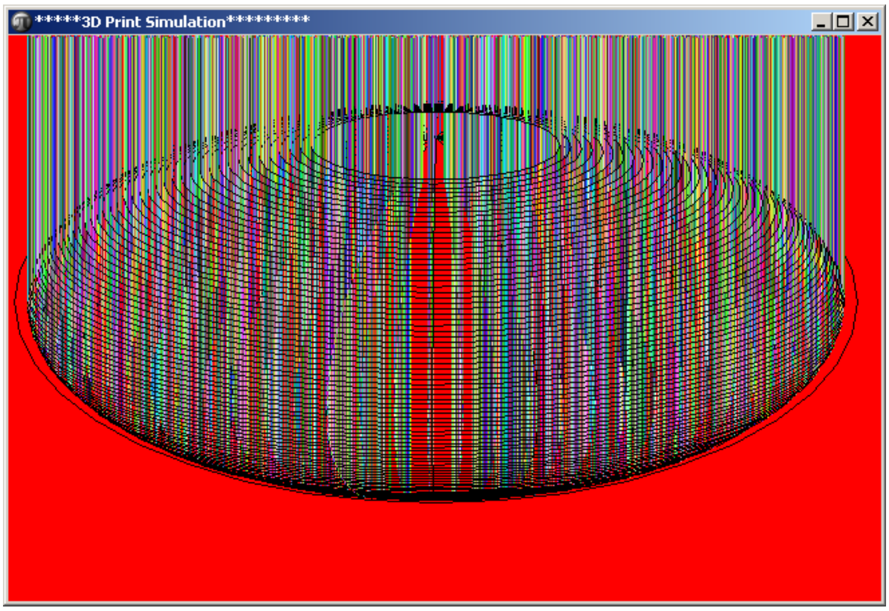
Such an amusement was a flyer I have seen a while ago with the following content: The Future is now! And what they told us is more fun than future, that you can get your portrait or family portraits by 3d printing. The company say they can scan virtually anyone, including babies, children, monsters, dogs, cats or rats.
The figures are made using an eco-friendly material called high-performance composite. A 3D printer creates the objects from a white powder (not resin) which is laid out layer by layer and solidified by multiple jets across the ground.

It's hard to imagine but take a look at this slicing simulation and you get the idea:

https://github.com/maxkleiner/maXbox3/blob/masterbox2/examples2/356_3D_printer.txt

Then load the script **356_3D_printer.txt** in the box an run it. It produces pics like this:





For each layer, the (virtual in the simulation script) laser beam traces a cross-section of the part pattern on the surface of the liquid resin. Exposure to the ultraviolet laser light cures and solidifies the pattern traced on the resin and joins it to the layer below.
So the main technology in which photopolymerization is used to produce a solid part from a liquid is SLA, mentioned in the first line of this report.

By the way this technique was invented in 1986 by Charles Hull, who also at the time founded the company, called 3D Systems.

The fact that the topic is so large we start with a real API of the Doodle3d Interface. We also focus on a less often discussed feature of open source printers and models and I want to highlight, or should I say stress, some of the practical problems that can arise.
As you may know these codes are widely used in the web of things or embedded computing industry; Beside the fact that many firmware is about new ideas, prototyping, concepts and new technologies that are in the lab, we start with end-users, programmers and products for:

- Electronics and Gadgets
- 3D Printing Production
- Modelling and Entertainment Pictures
- Spare Parts in Industry

1.2 3D Interface as “Printerface”

Doodle3D supports almost all open source printers (because their firmware is usually based on **Marlin**, more below), this includes material printers like the Ultimaker, PrintrBot, Delta RostockMax. They also support the Makerbot printers and printers that use the same firmware you find in a hacker space, fab-lab or maker zone.

Doodle3D is a product line for 3D printing. The Doodle3D WiFi-Box makes almost all 3D printers wireless controllable through a simple REST API (second part). This means you can control them using Processing with Arduino, open Frameworks, JavaScript, Pascal, Lazarus, Arduino, maXbox, Cinder etc. Basically any language that can send and receive HTTP requests (AJAX).

Lets start with G-Code.

In fundamental terms, G-code is a language in which people tell computerized machine tools what to make and how to make it. The "how" is defined by instructions on where to move to, how fast to move, and along what path to move.
The G-Code we focus on is based on what is supported by Marlin and this software (firmware) runs on the 3D printer itself. Almost all open source 3D printers use a version of Marlin. G-Code is a commonly use language to control CNC machines. It is light-weight, simple to parse as a format, designed to be able to run on modest hardware, such as Arduino loaded with this firmware (in this case). It is a format that is prepared in advance and created from the digital design files (CAD files).

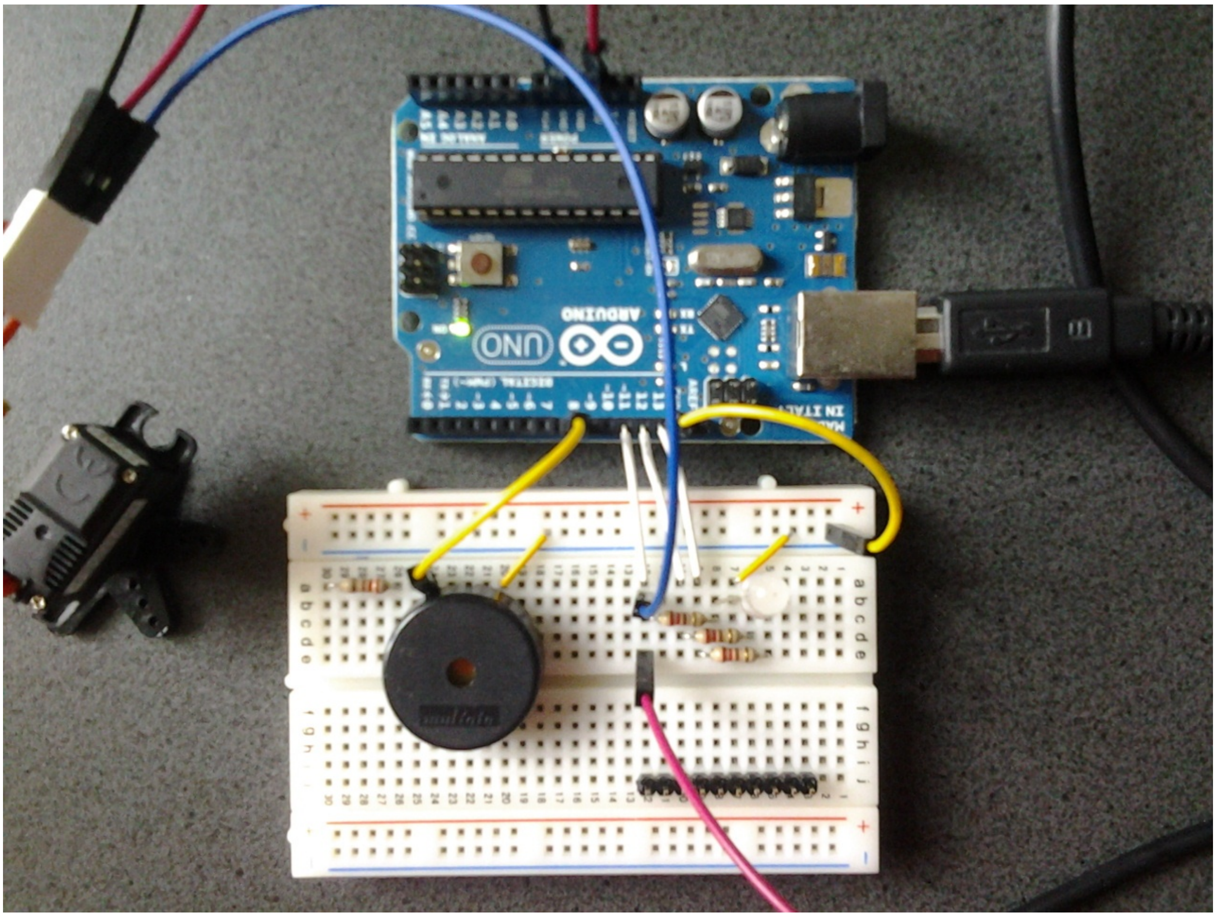
```
G1 X0.0 Y0.0 Z0.0
G1 X10.0 Y10.0 Z0.0 F1000
```

These instructions can then be sent to a machine that will interpret these lines and execute them one by one. The G Code instructions frequently have an X, Y and Z coordinate, these are the points in 3D space that the print-head will move in. We developed with Turtle in maXbox/Arduino these steps to move the print-head:

```
penDown:= true;
 setpos(180, 430);
    cPort.WriteStr(chr(180));
    //# marlinServo body
    //turn(aturn)
    sleep(1000)
    left(180)
    cPort.WriteStr(chr(180));
    sleep(1000)
    moveforward(100)
    right(90)
    cPort.WriteStr(chr(120));
```

The point is that through this piece of Turtle G Code, the head will be moved from its current position to X, Y position 180.0 which in this case is an angle. The Z position is missing. This will happen at the last speed setting (feed-rate, F) that was sent, depending on the printers specification in FDM or FFF technology:

```
Printing technology: FFF (Fused Filament Fabrication)
Printing volume: 170x170x160m
Layer resolution: 100µ (0.1mm)
X-Y-Z precision: 20µ (0.02mm)
Filament diameter: 1.75mm
Heated bed: No
Bed autoleveling: Yes
Printing material: PLA (any material with not a heated bed)
Time to first print-out: <1m
Injector diameter: 0.4mm
Filament capacity: 500gr
Dimensions: 275x275x360
Approx. weight: 7Kg
Chassis: Unibody aluminum
```



Note that the X and Y will expand if you expand the size() and it is possible that you make the printer move outside of its border! This can do harm or damage to your 3D printer!!

The G-Code firmware source is available from Source Forge as part of the RepRap Arduino firmware package. Apart from printing your objects, the Doodle3D WiFi Box will also be able to wirelessly 3D print more complex 3D models by streaming the GCODE to your printer. You can also monitor your printer's speed, temperature and progress.
But wait a minute, what is RepRap?

RepRap is a general-purpose self-replicating manufacturing machine. RepRap takes the form of a desktop 3D printer capable of printing plastic objects. Since many parts of RepRap are made from plastic and RepRap prints those parts, RepRap self-replicates by making a kit of itself - a kit that anyone can assemble and build given time and materials.
But which RepRap to choose?

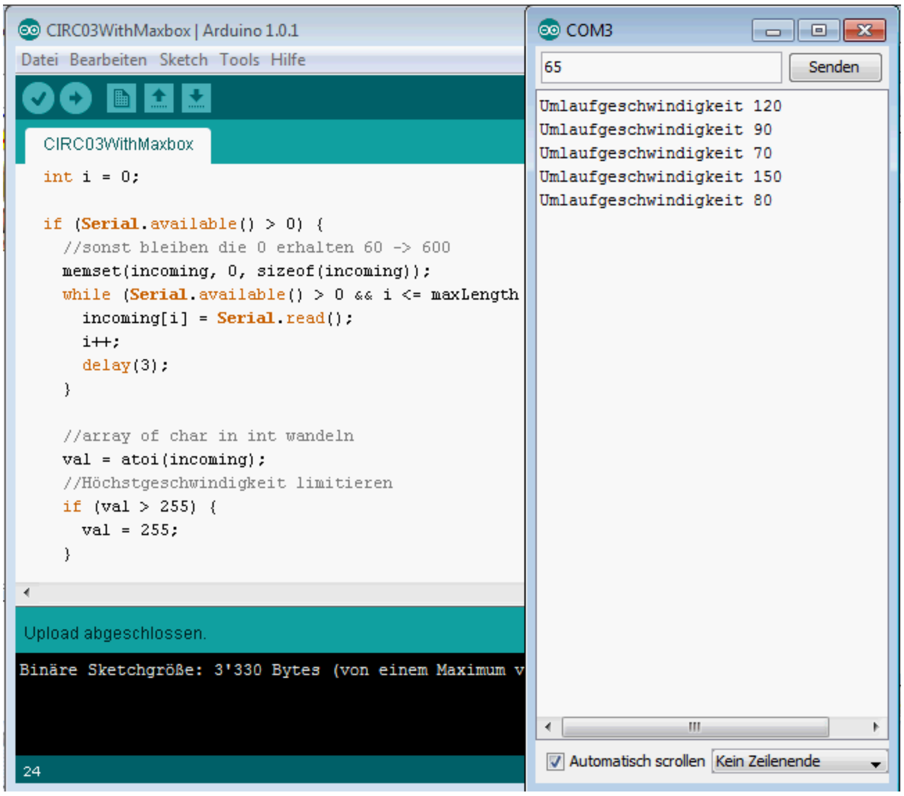
1.3 3D as an Illusion or back to 2D

Most of our strength to put something in 3D is paved with good intentions but lacked of services and simplification.

For example 3D Glasses like the coming ones invoke headaches or discomfort.
Motivated by an increased interest of physical computing and embedded in automatic management of large systems they want us to buy 3D things but forget that 2D is most of the cases more efficient and easy to use.
Obviously a seemingly 3-D world exists only on a 2-D screen!
For all we know, the three dimensional world we see around us is really an illusion — one that's actually in 2D:

http://www.theverge.com/2014/8/26/6071511/a-physics-experiment-might-soon-tell-us-if-were-living-in-a-2d

The scientists are using a device called Holometer or holographic interferometer. It is the most sensitive device ever created to measure the quantum jitter of space itself.


CIRC03WithMaxbox | Arduino 1.0.1
Datei Bearbeiten Sketch Tools Hilfe
CIRC03WithMaxbox
COM3
Senden
24

```
int i = 0;

if (Serial.available() > 0) {
  //sonst bleiben die 0 erhalten 60 -> 600
  memset(incoming, 0, sizeof(incoming));
  while (Serial.available() > 0 && i <= maxLength
    incoming[i] = Serial.read();
    i++;
    delay(3);
  }

  //array of char in int wandeln
  val = atoi(incoming);
  //Höchstgeschwindigkeit limitieren
  if (val > 255) {
    val = 255;
  }
```

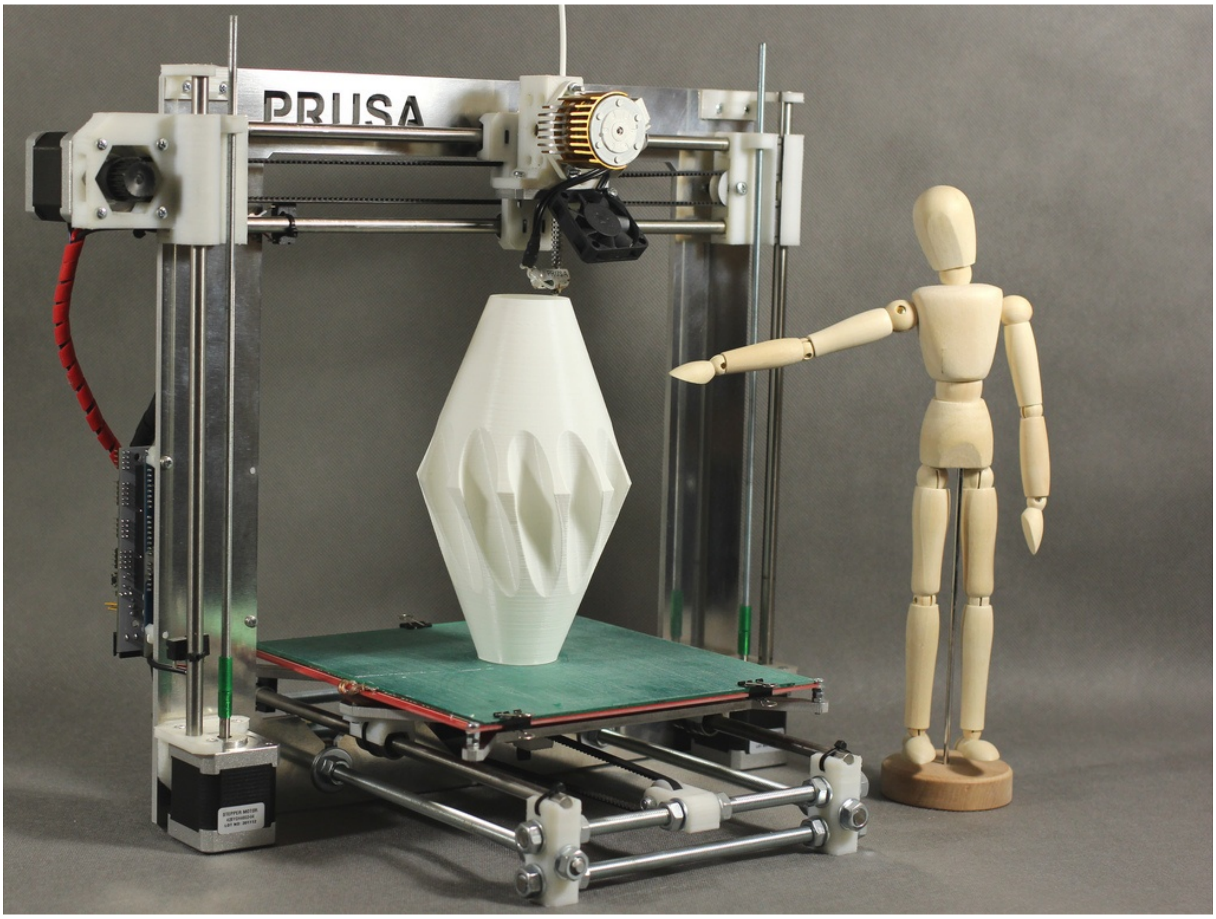
Upload abgeschlossen.

```
Binäre Sketchgröße: 3'330 Bytes (von einem Maximum v
```

65

```
Umlaufgeschwindigkeit 120
Umlaufgeschwindigkeit 90
Umlaufgeschwindigkeit 70
Umlaufgeschwindigkeit 150
Umlaufgeschwindigkeit 80
```

☑ Automatisch scrollen   Kein Zeilenende



Ref: reprap.org/wiki/File:Prusai3-metalframe.jpg

So in my opinion the Prusa I3 (picture) or Mendelmax is well appreciated and documented in the community. For about a thousand dollars people have been able to buy the RepRap kit and put together their own personal 3D printer, complete with any customizations they were capable of making. What really speeds the development is the open source idea and made some prototypes to share. Using 3D printers for these purposes is called rapid prototyping.

But what’s the matter with the REST API? Simply that each thing, device or gadget will have an REST API to communicate, could also be a code to implement.

You can think of Doodle3D as the first application written on top of the API, it makes 3D printing a simple sketch child's play. There are projects, which uses the API to enable sending a print to your printer. Using the WiFi-Box you could control a printer using a Kinect, influence a print with sound input in a sonar way, maybe create a cup designer or adding an embeddable print button to an existing design application.
Our API enables you to go to a URL, and retrieve a json or XML object with information. Like retrieving status info using the following URL snippet:

```
{IP address of WiFi-Box}/d3dapi/info/status
```

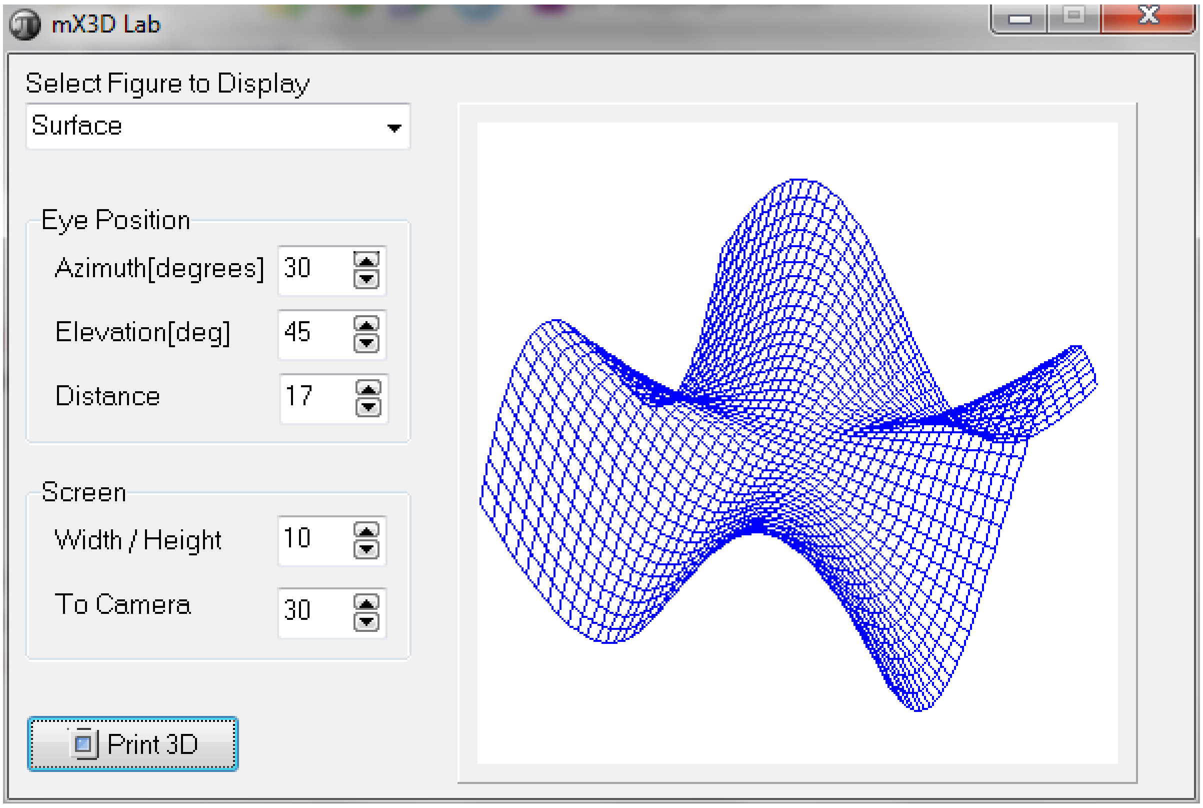
The call could be that simple:

```
import httprocessing.*; TurtleSim
     String boxIP = "10.0.0.188";
     PostRequest post = new
     PostRequest("http://"+boxIP+"/d3dapi/printer/print");
```

```
void setup() {
     post.addData("gcode", "G28 X0.0 Y0.0 Z0.0");
     //Send printer to the x0 and y0 position
     post.addData("start", "true");
     post.send();
     size(100, 100);
```

This is so simple because Doodle3D provides a service called host module we use.
The parts that connects users wirelessly to their printer (or more specificly the controller) are:

- The web (websocket / REST) API,
- The logic to connect to networks and talk to the Controller.



Select Figure to Display: Surface

Eye Position

Azimuth[degrees] 30

Elevation[deg] 45

Distance 17

Screen

Width / Height 10

To Camera 30

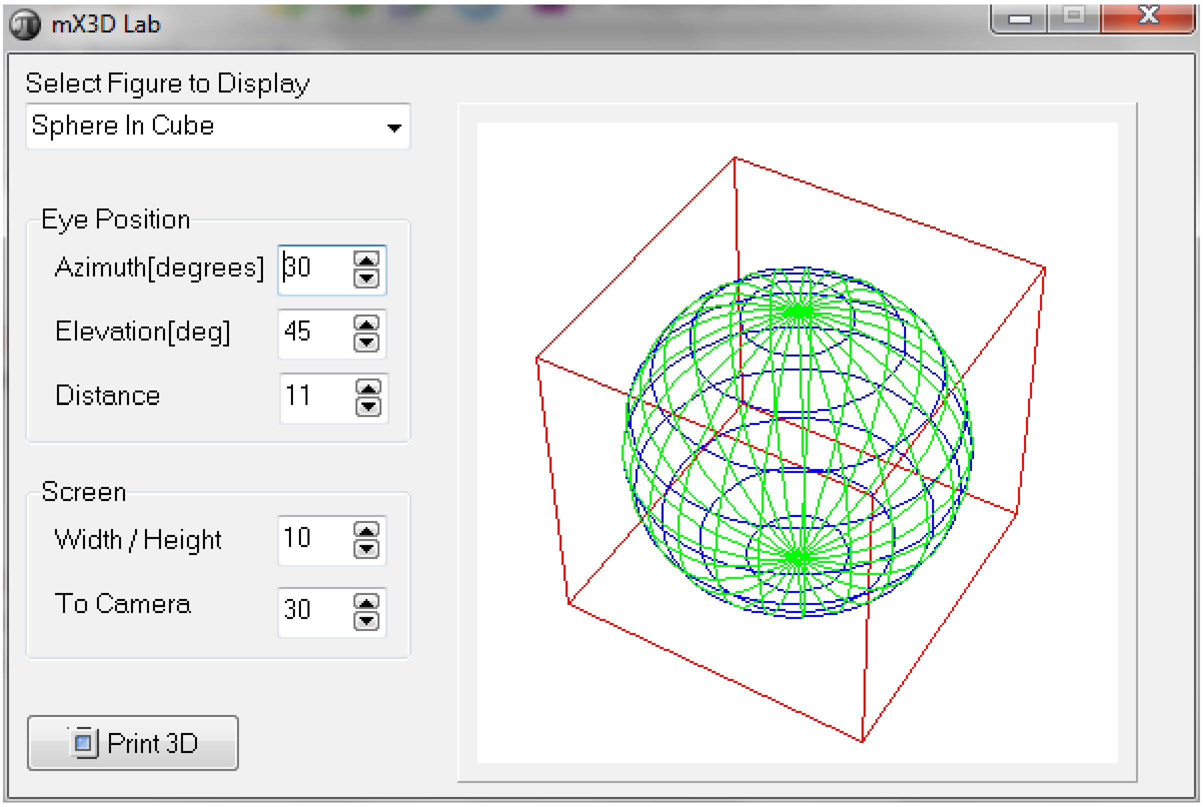
Use Case of a 3D Print GUI

```
//Forecast: REST API and 3D GUI Case Study
```

☞ So in a next number (Tutor 40 REST Coding) we will go deeper in that API Code and we will made some calls. Concerning G-Code is still under research. For example a Web-Interface means that most of the proposals are too large, had an unusable interface or are unsupported for various reasons.
Node.js for instance, cannot be used because it cannot currently be built for MIPS processors (issues with big endian instructions). Python is simply too big, even python-mini leaves only about 200 KB on the device even without any dependencies.

To keep the firmware code simple and consistent it will only handle G-code. So the G-code will be generated using javascript or another scripting language.

"Variables in the end are a box. You put something in there, and it stays there, unchanged, until you do something else with what's in that box. When you refer to the box by its name, the contents of that box are used to do stuff: math, make decisions, communicate with something or someone, and so on. There are several different types of box, for holding different kinds of data: boolean, bytes, integers , long integers, characters, floating-point numbers, and so on. You decide what type of box you want when you declare the variable."



Select Figure to Display
Sphere In Cube

Eye Position
Azimuth[degrees] 30
Elevation[deg] 45
Distance 11

Screen
Width / Height 10
To Camera 30

```
//maXbox: ..menu/Output/3D Lab
```

Feedback @
max@kleiner.com

Literature: Kleiner et al., Patterns konkret, 2003, Software & Support

Links of maXbox, 3D Coding, Printing and Web of Things:

http://www.slideshare.net/maxkleiner1/a-3d-printing-solutionekon

http://www.softwareschule.ch/download/webofthings2013.pdf

http://3dprinting.com/what-is-3d-printing/
http://www.doodle3d.com/help/api-documentation
http://replicat.org/mcodes
http://www.doodle3d.com/help/g-code