



waag society

institute for art, science and technology

Waag Society - CC-BY-SA 3.0



BioHack Academy Spectrometer Design

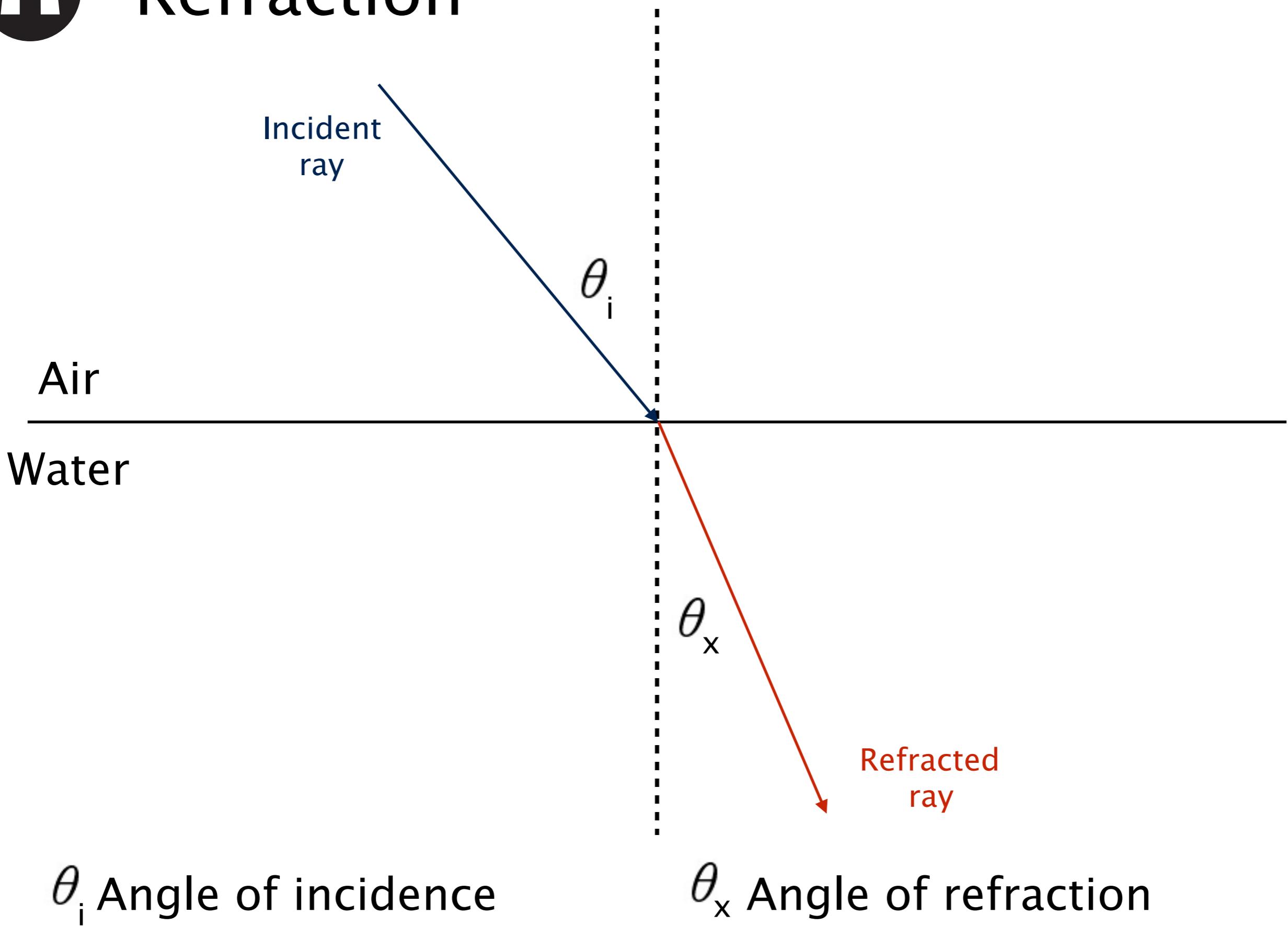


Tibicos – kefir





Refraction



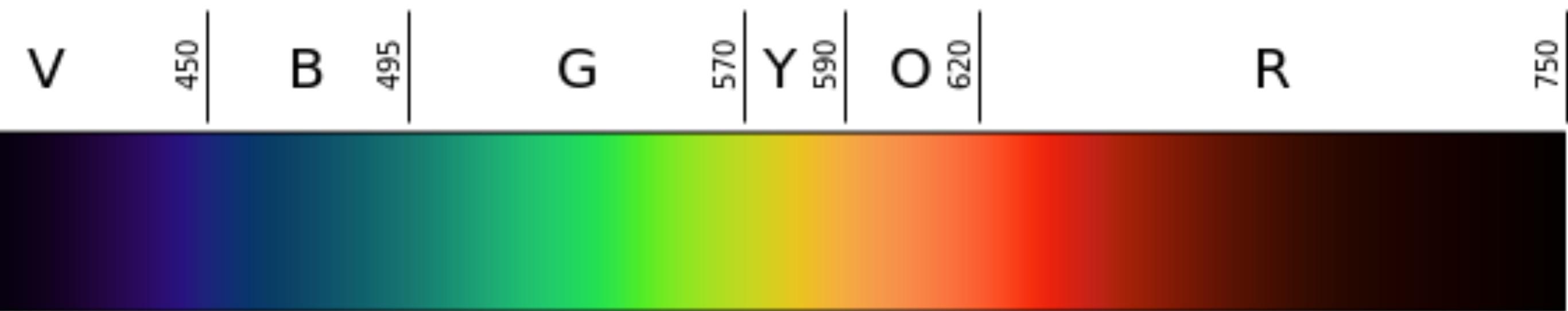


Snell's Law

$$\sin(\theta_i) = n_{material} \times \sin(\theta_x)$$

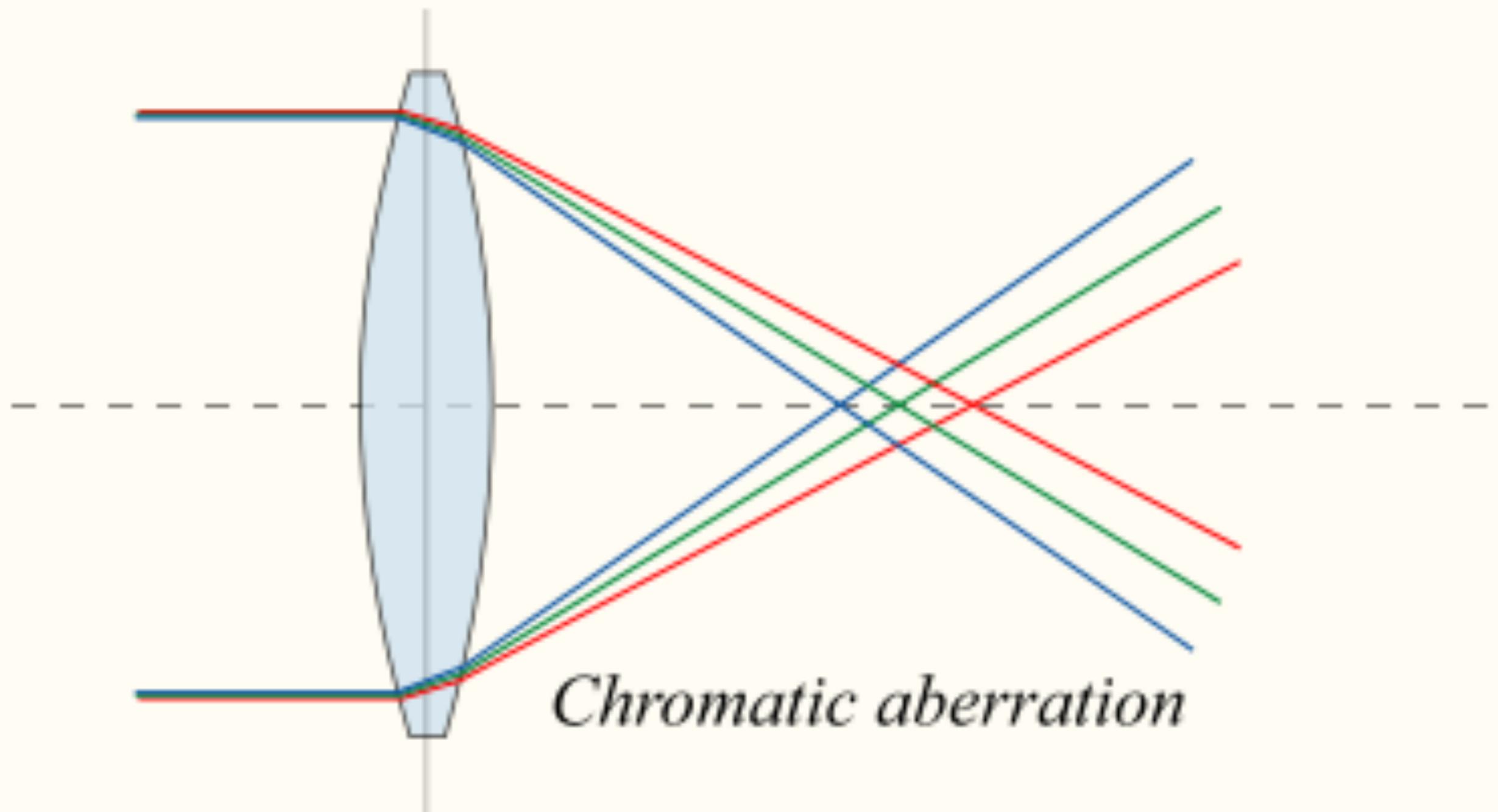


Visible light spectrum



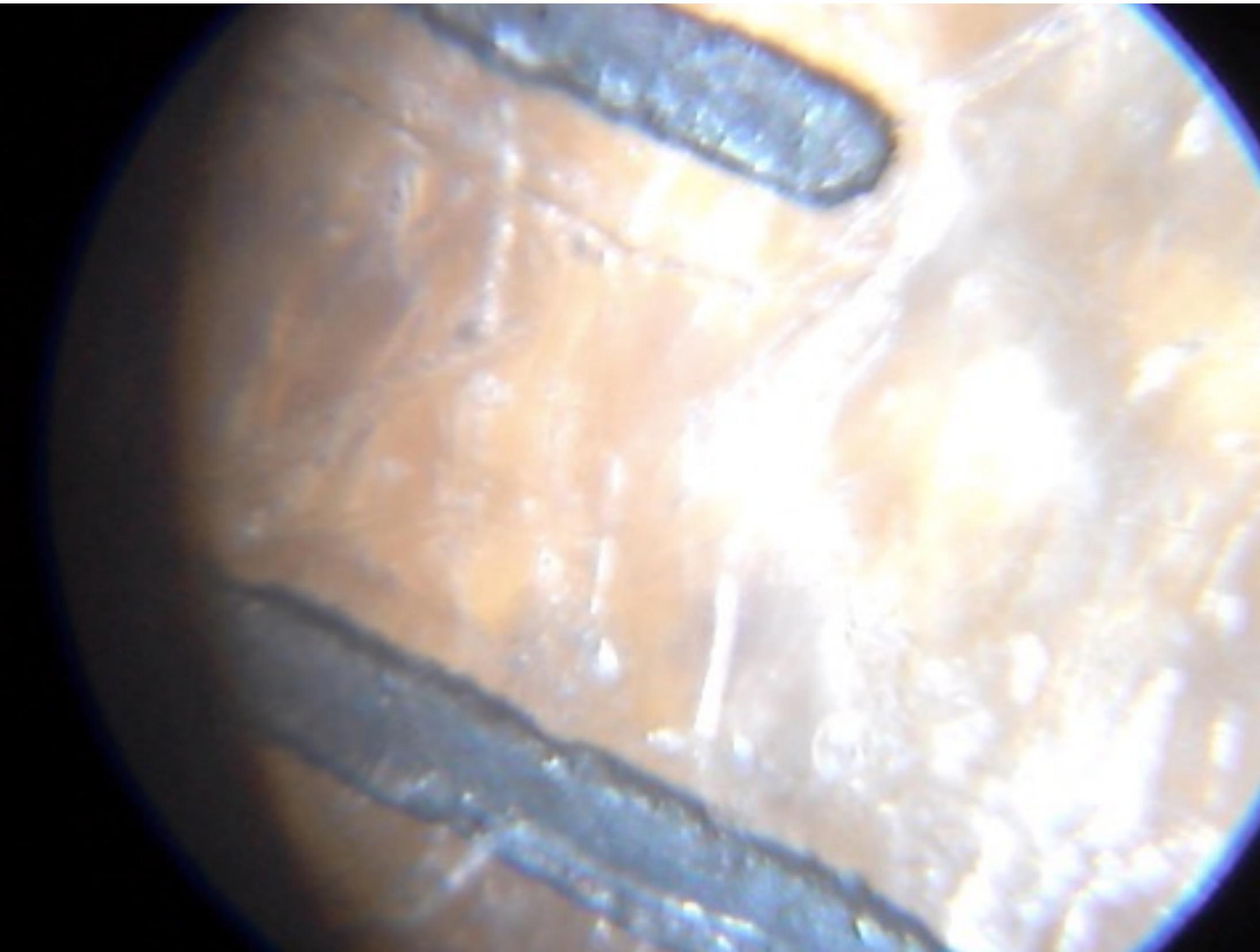


Chromatic aberration





Playing with the microscope



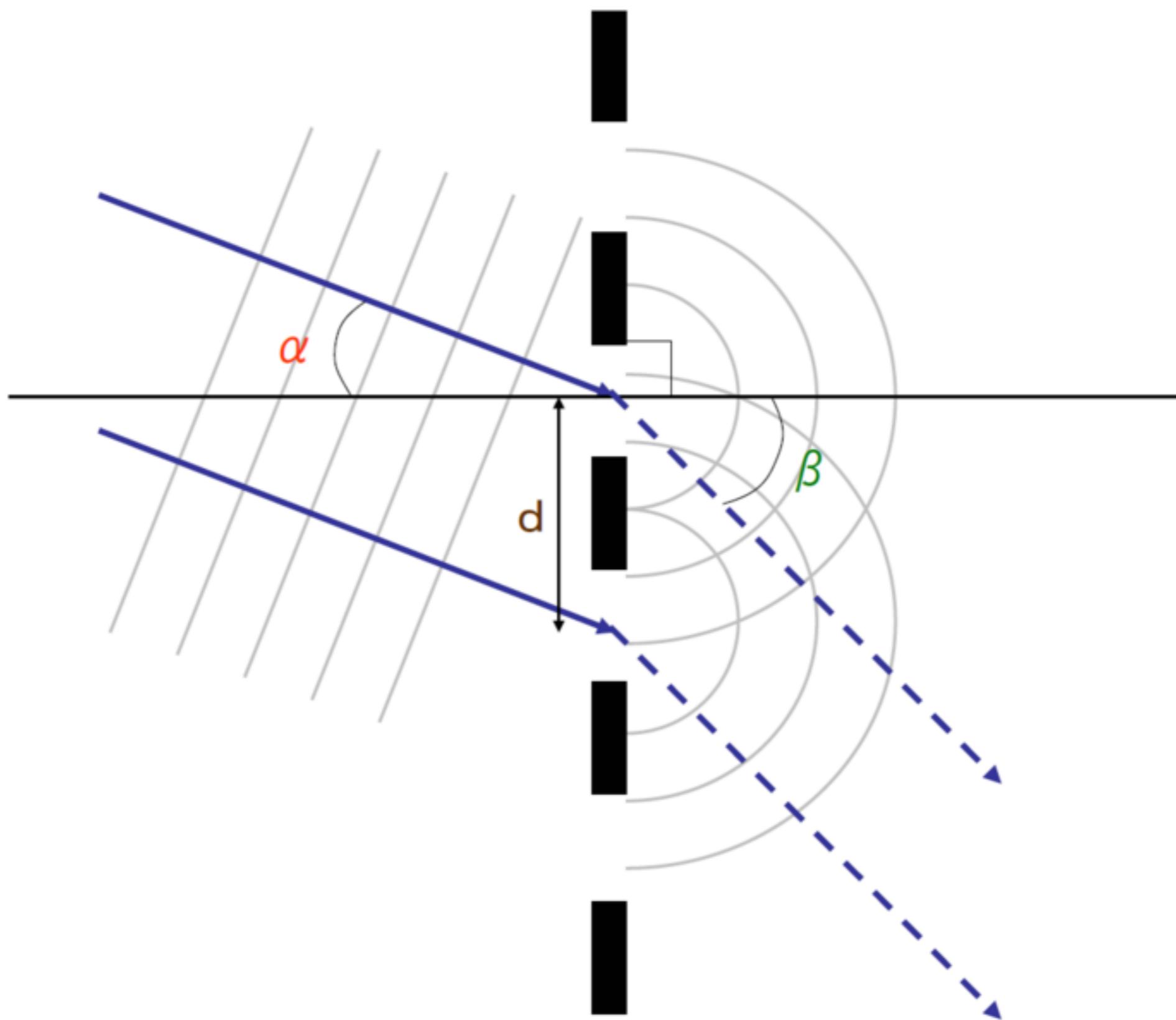


Grating diffraction



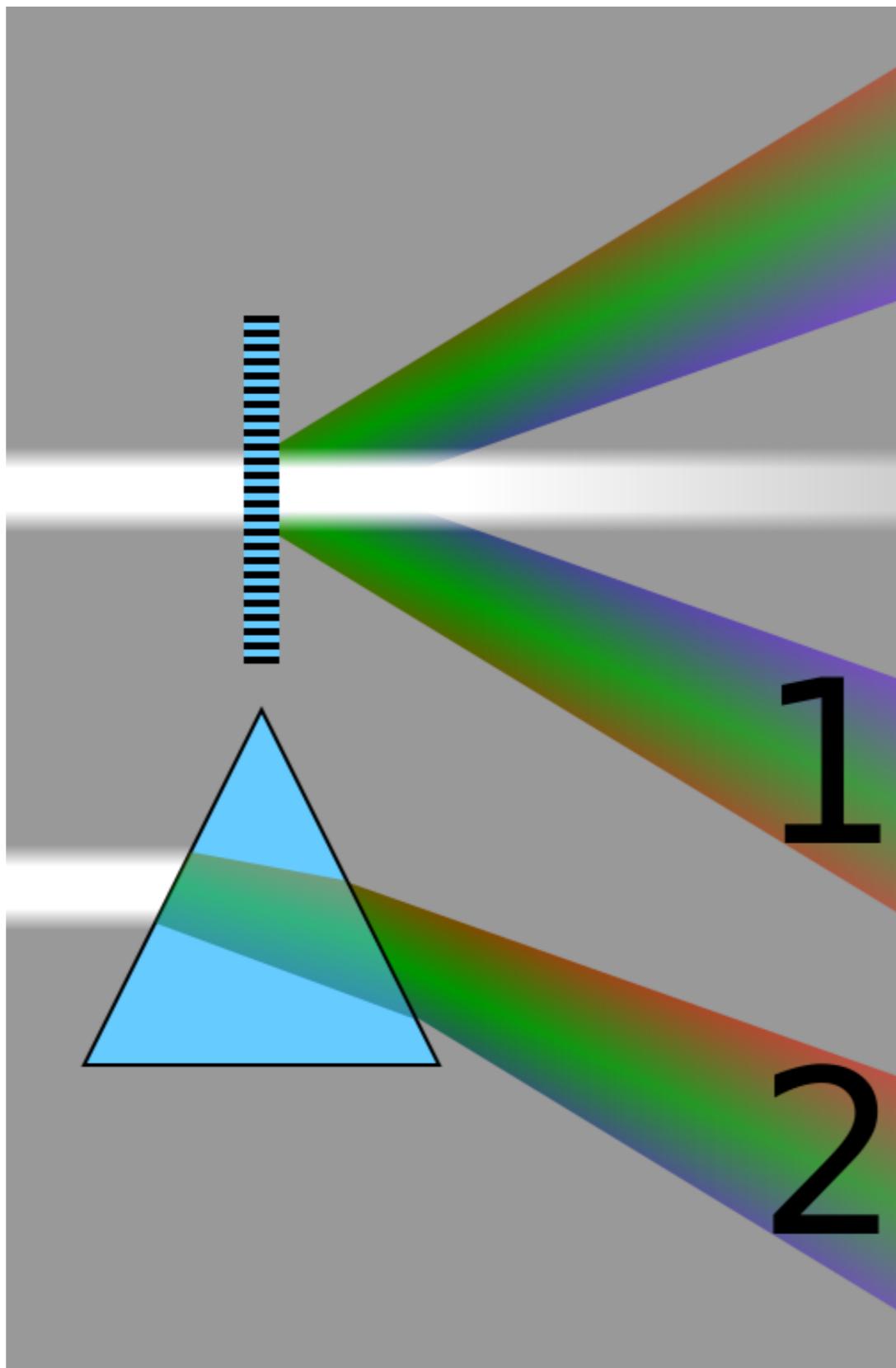


Diffraction grating principle





Just like a prism



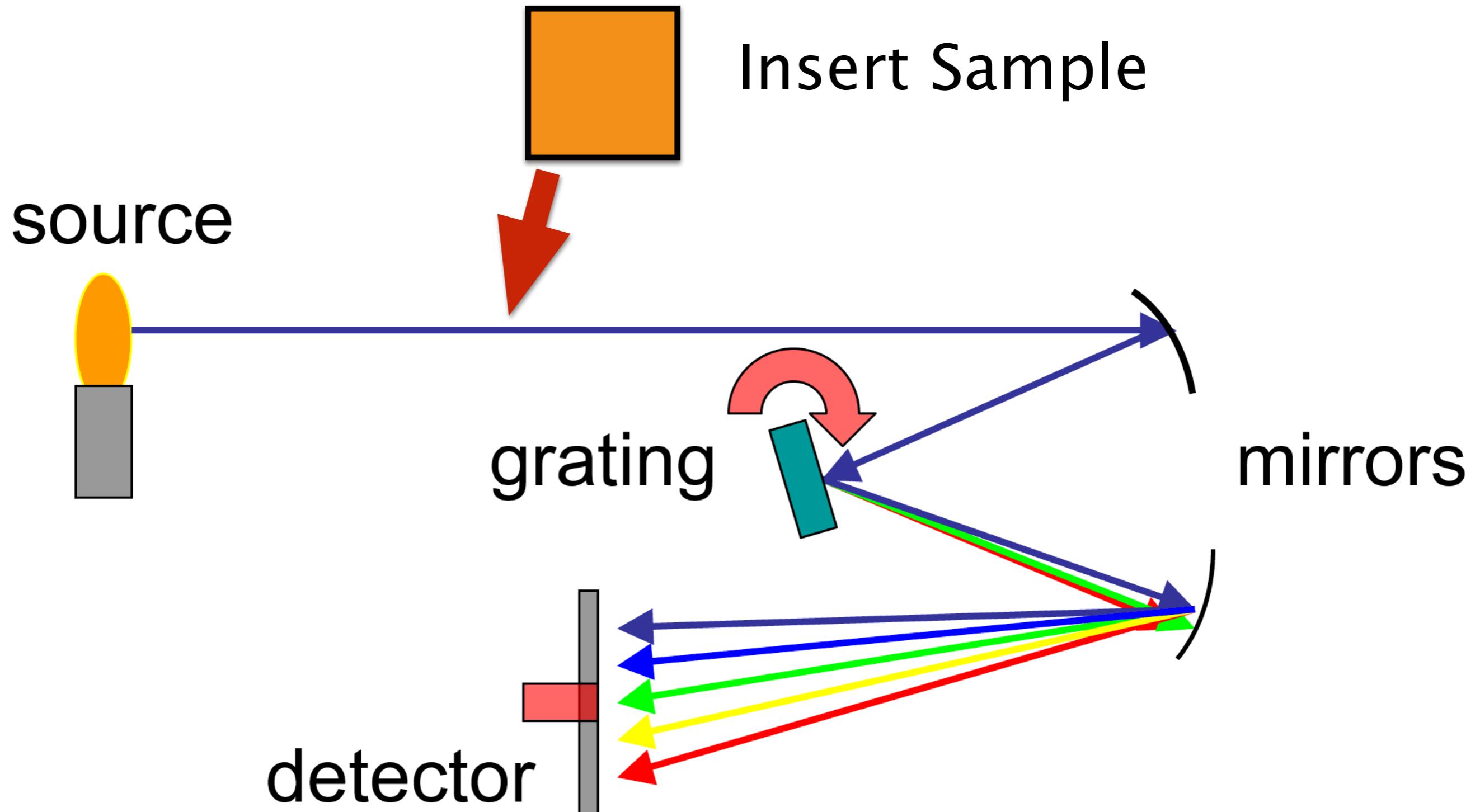


Industry standard



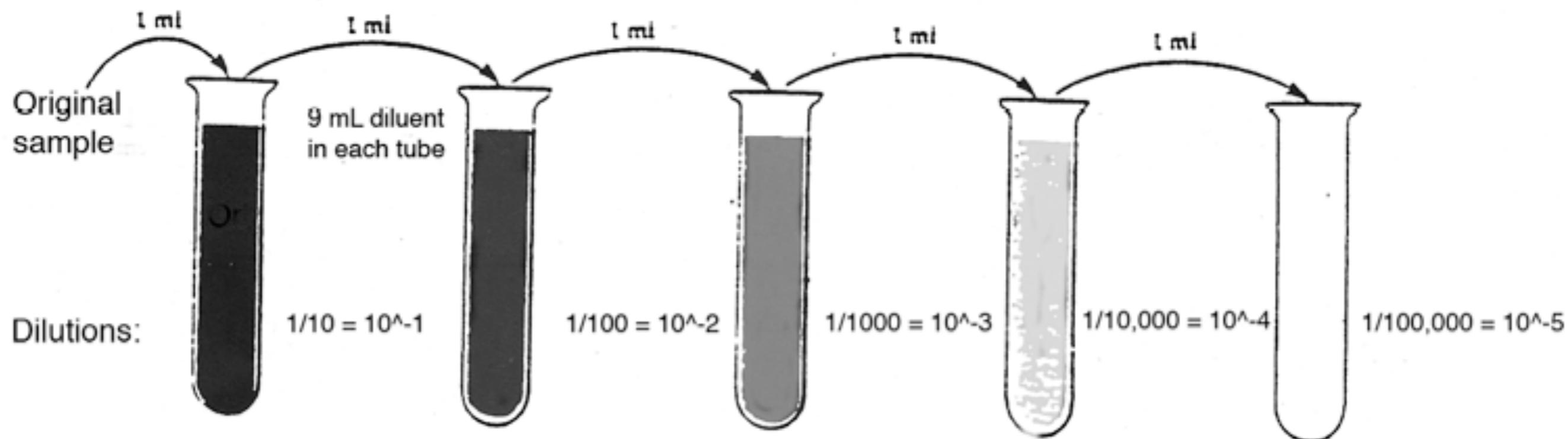
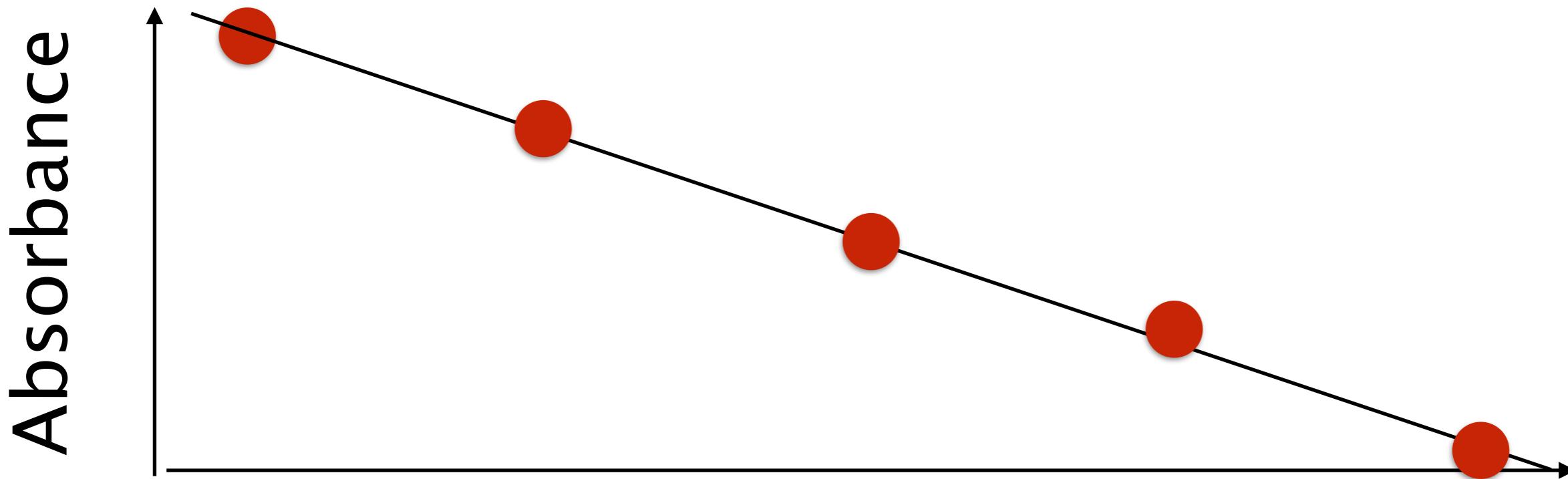


Inside



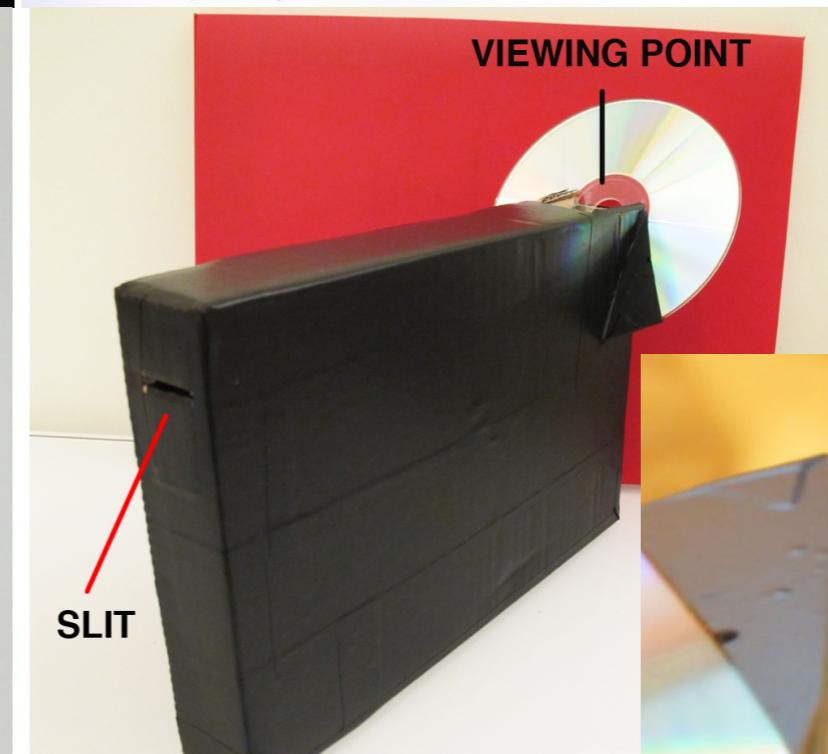
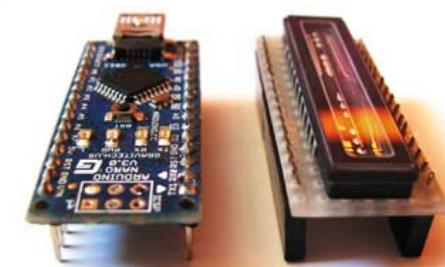
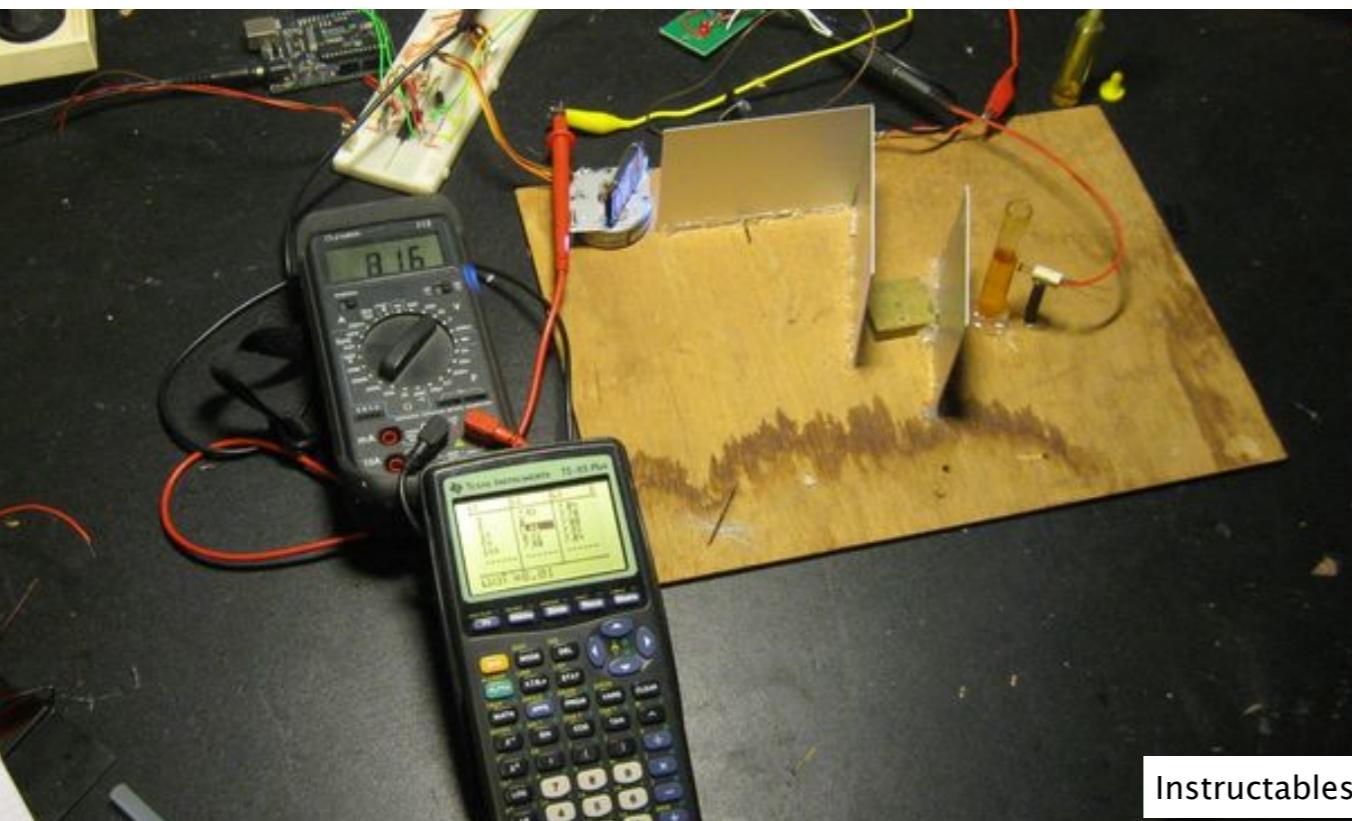


Dilution series





Previous hacks



<http://www.cs.cmu.edu/~zhuxj/astro/html/spectrometer.html>

PublicLab



Content

- Webcam spectrometer
- CCD Array spectrometer



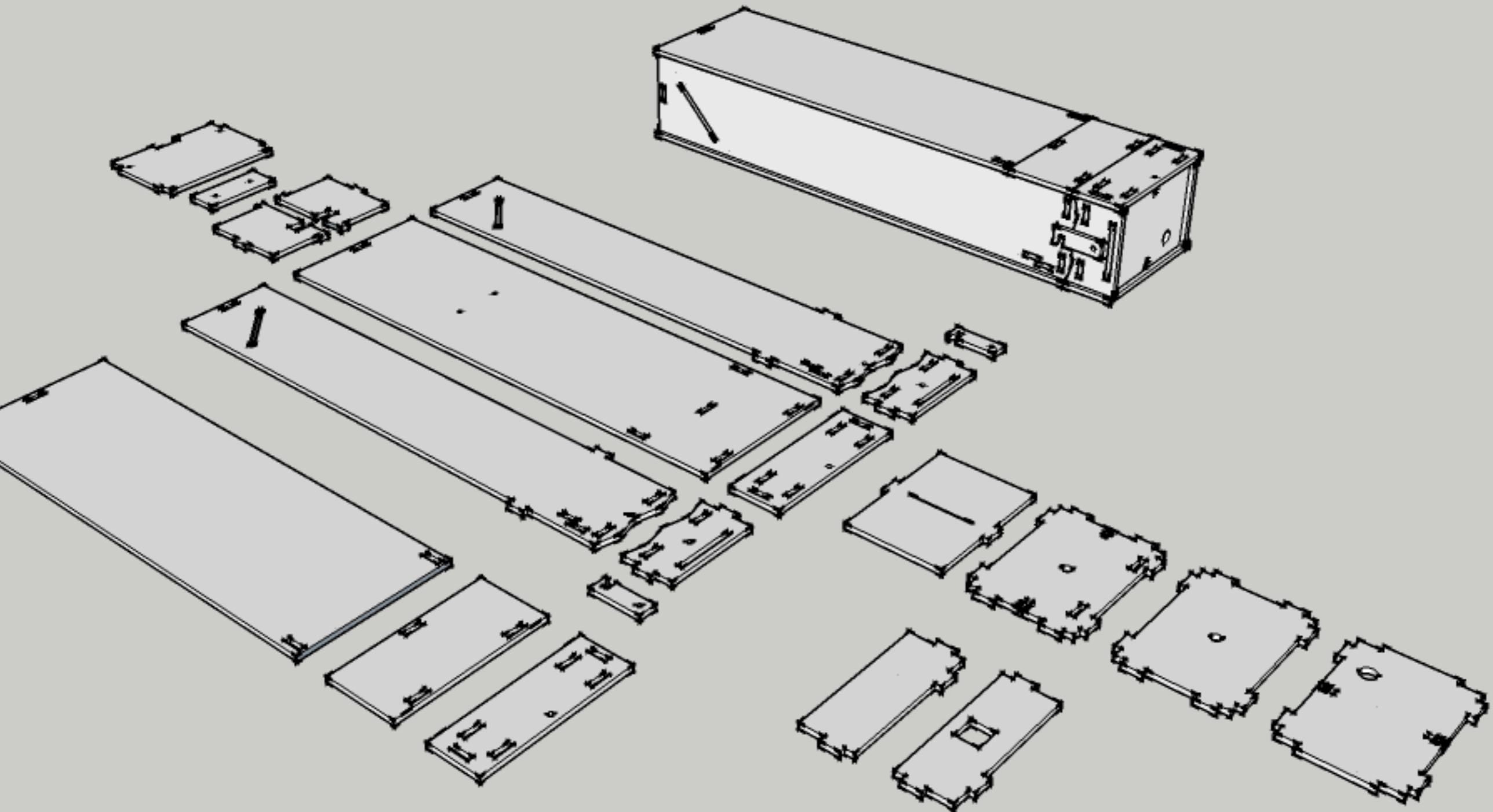
waag society

institute for art, science and technology

Webcam Spectrometer

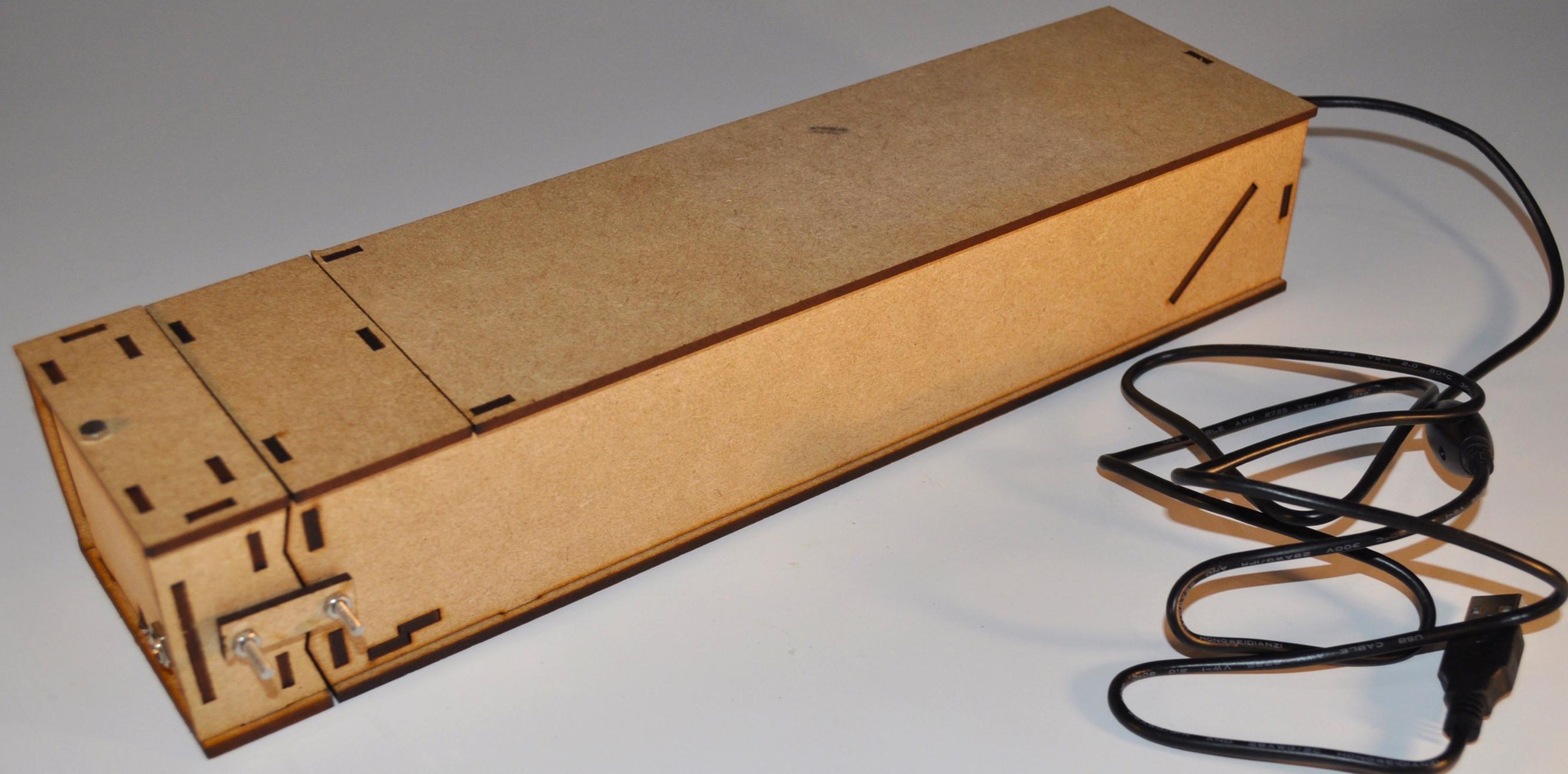


BioHack Academy design



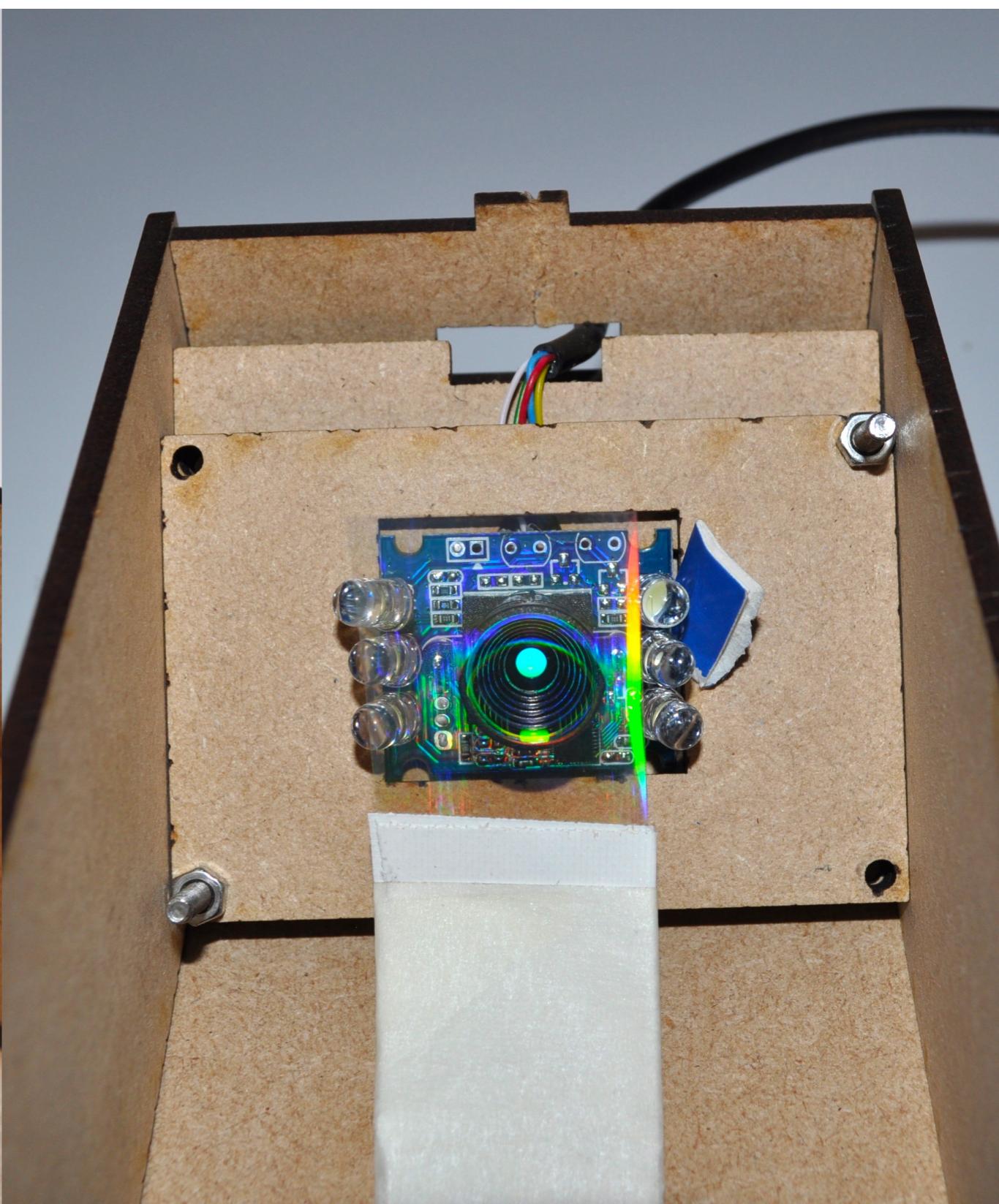
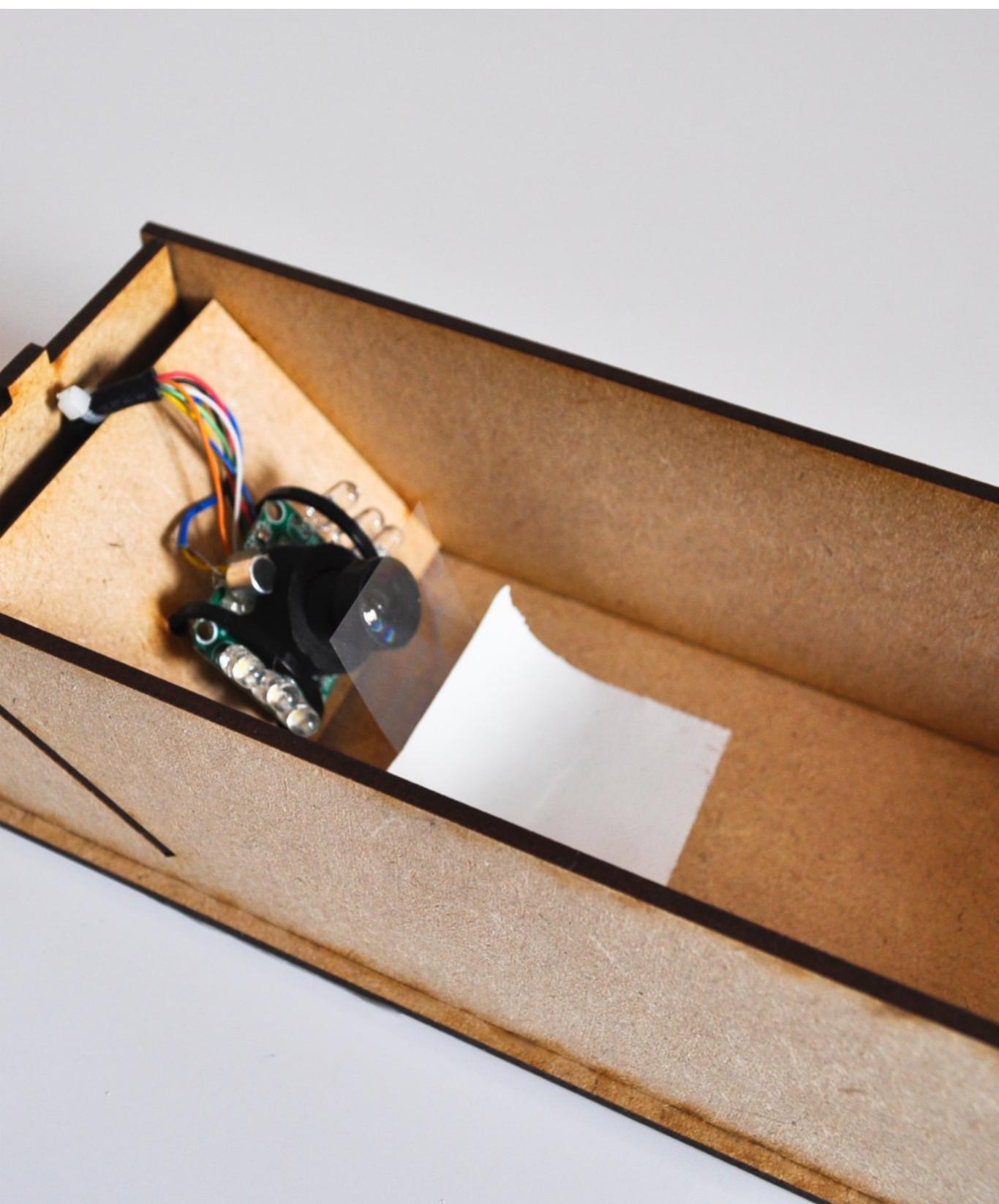


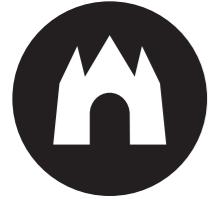
BioHack Academy design





BioHack Academy design





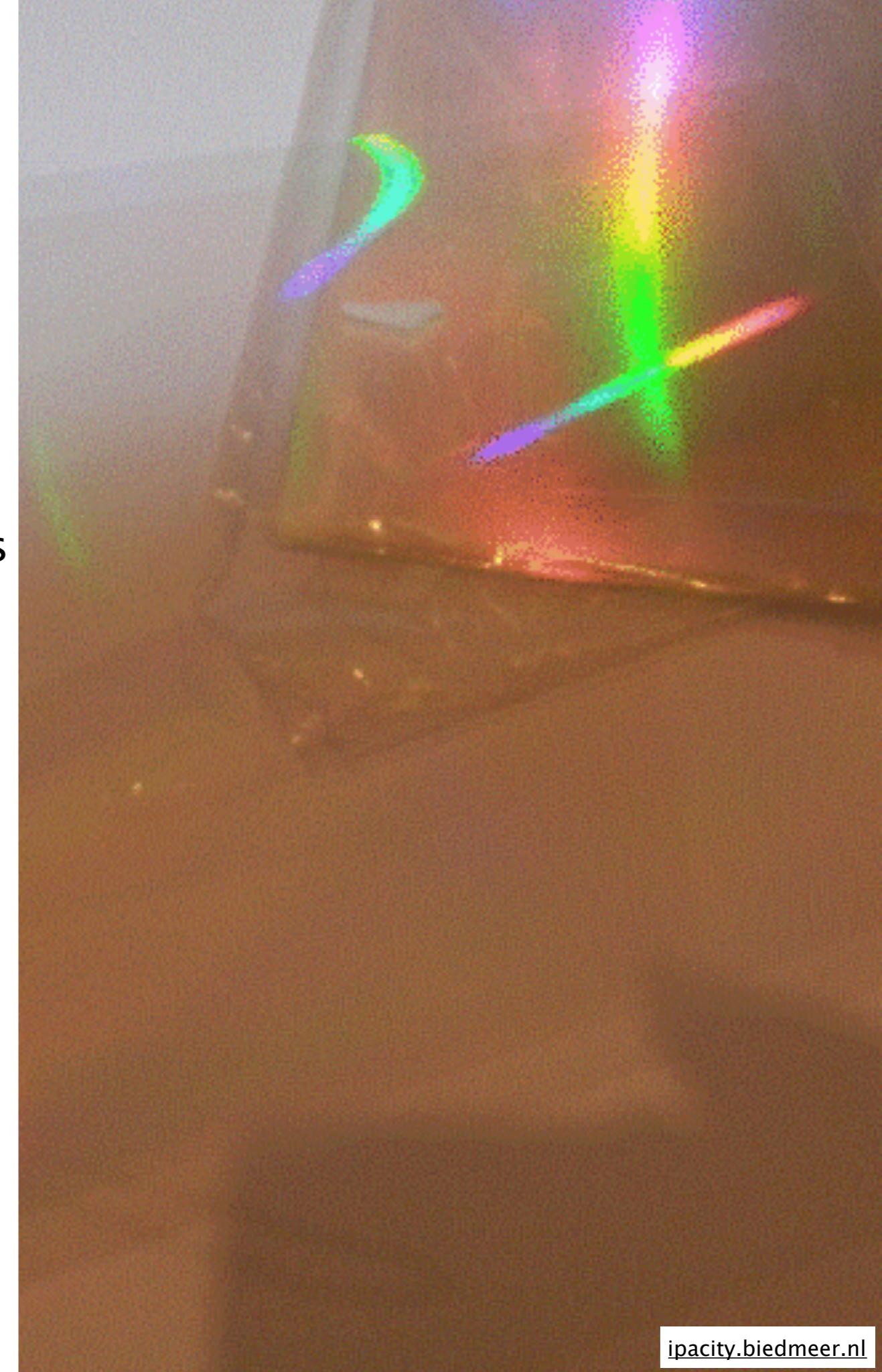
Bill of Materials

#	Amount	Description
1	1	Webcam
2	1	3mm MDF
3	1	White LED
4	1	220 Ohm resistor
5	1	7.5 W power supply
6	1	Jack Adapter
7	1	4.7 GB DVD-R (track pitch 0.74 microns) or 1,000 lines/mm grating



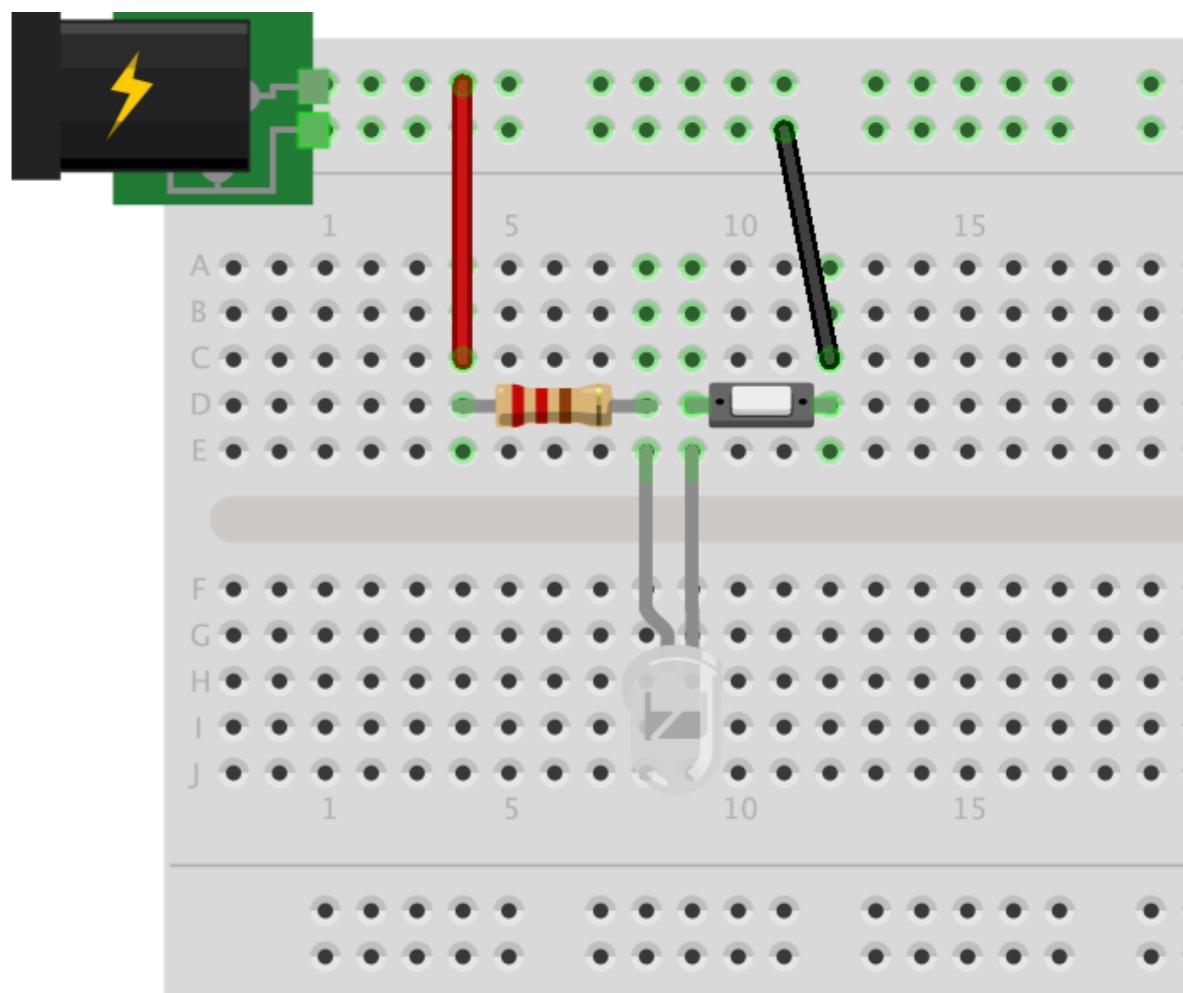
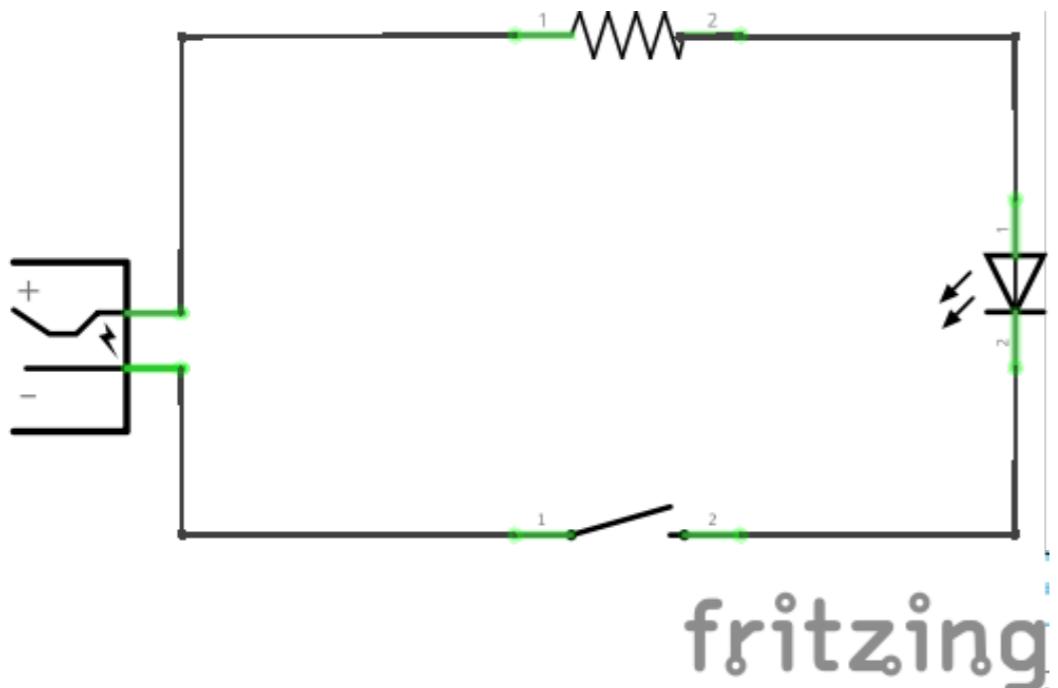
Foil

- Angle 45 degrees
 - Grating foil
 - 1000 lines per mm
 - DVD-R 4.7 GB:
 - Track pitch of 0.74 microns
 - 1351 lines per mm
 - DVD-R 3.95 GB:
 - Track pitch of 0.8 microns
 - 1250 lines per mm
 - CD:
 - track pitch 1.6 microns
 - 625 lines per mm





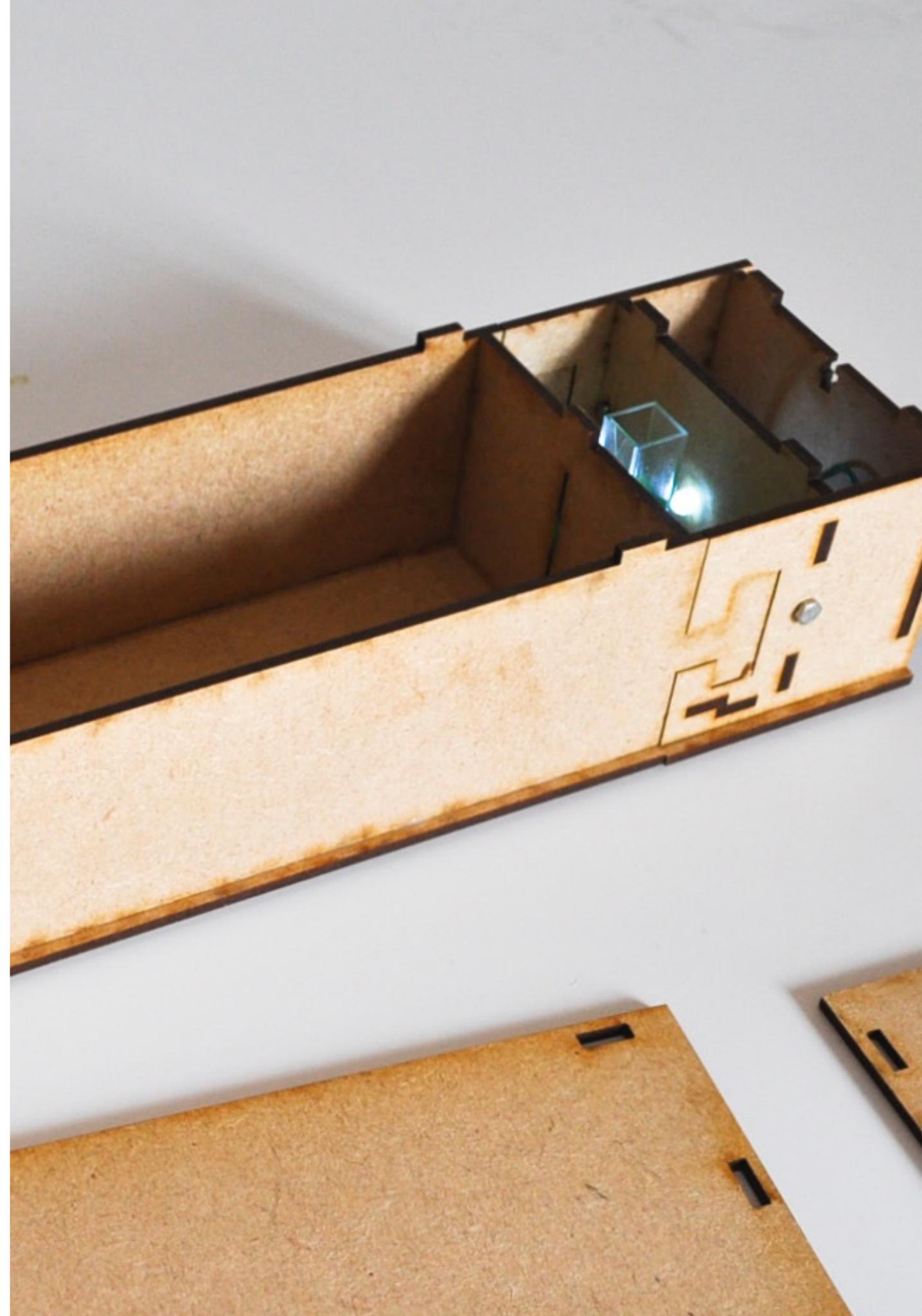
LED Wiring





Slit

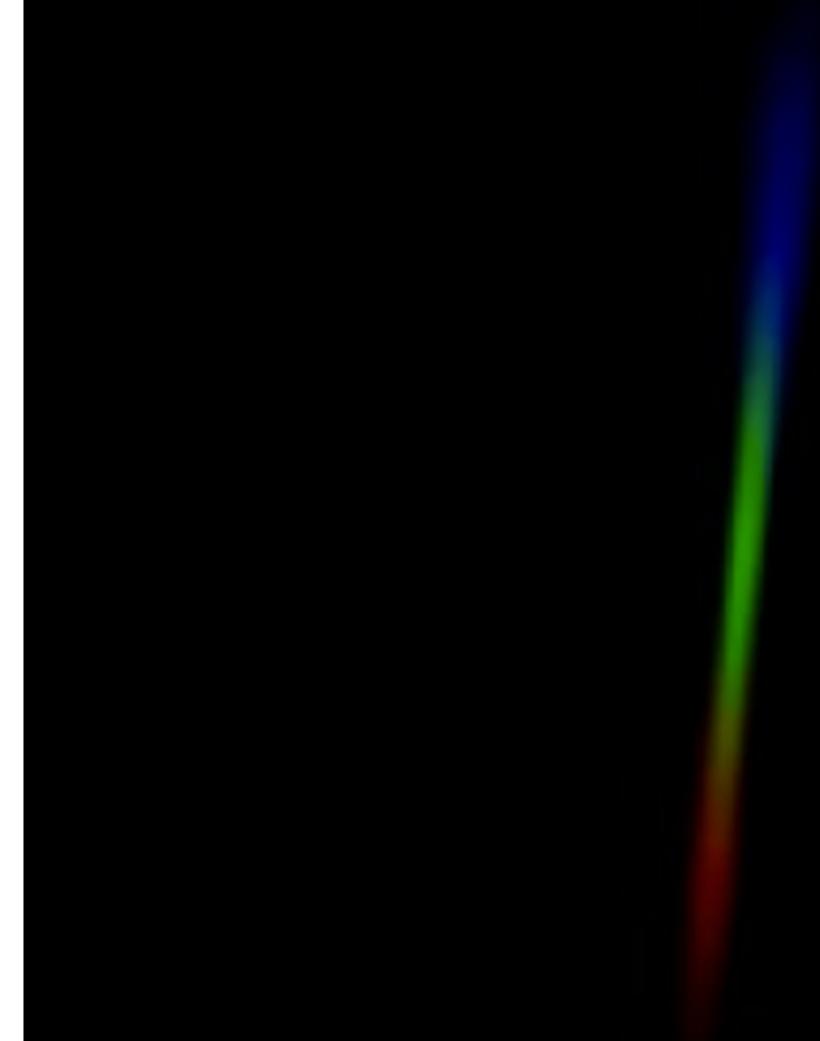
- Slit size: 1mm
- Lower is better resolution, but less sensitive
- Distance to slit: 22 cm
 - Focus webcam on this distance





Spectrum

- Day light





Spectral Workbench analysis

The figure shows the Spectral Workbench interface. At the top, there is a navigation bar with icons for Home, Search, Browse, Learn, User Profile (pieter), and a blue button for 'Capture spectra'. Below the navigation bar is a color calibration target consisting of a white card with various colored patches. A corresponding spectral plot is displayed below the target, showing a graph of relative intensity (%) from 0% to 100% on the y-axis against wavelength (nm) from 250 nm to 850 nm on the x-axis. The plot shows a broad emission band centered around 550 nm, with a small peak at approximately 568 nm. A legend in the upper right corner of the plot area indicates 'BHA Test = 16%'.

BHA Test

uploaded by pieter | March 31, 2015 07:26

Tags: [upload](#) x [device:custom](#) x



Notes

 Like 0  Tweet 0

 0 likes  Edit  Delete

Comments (0)

Comments (0)

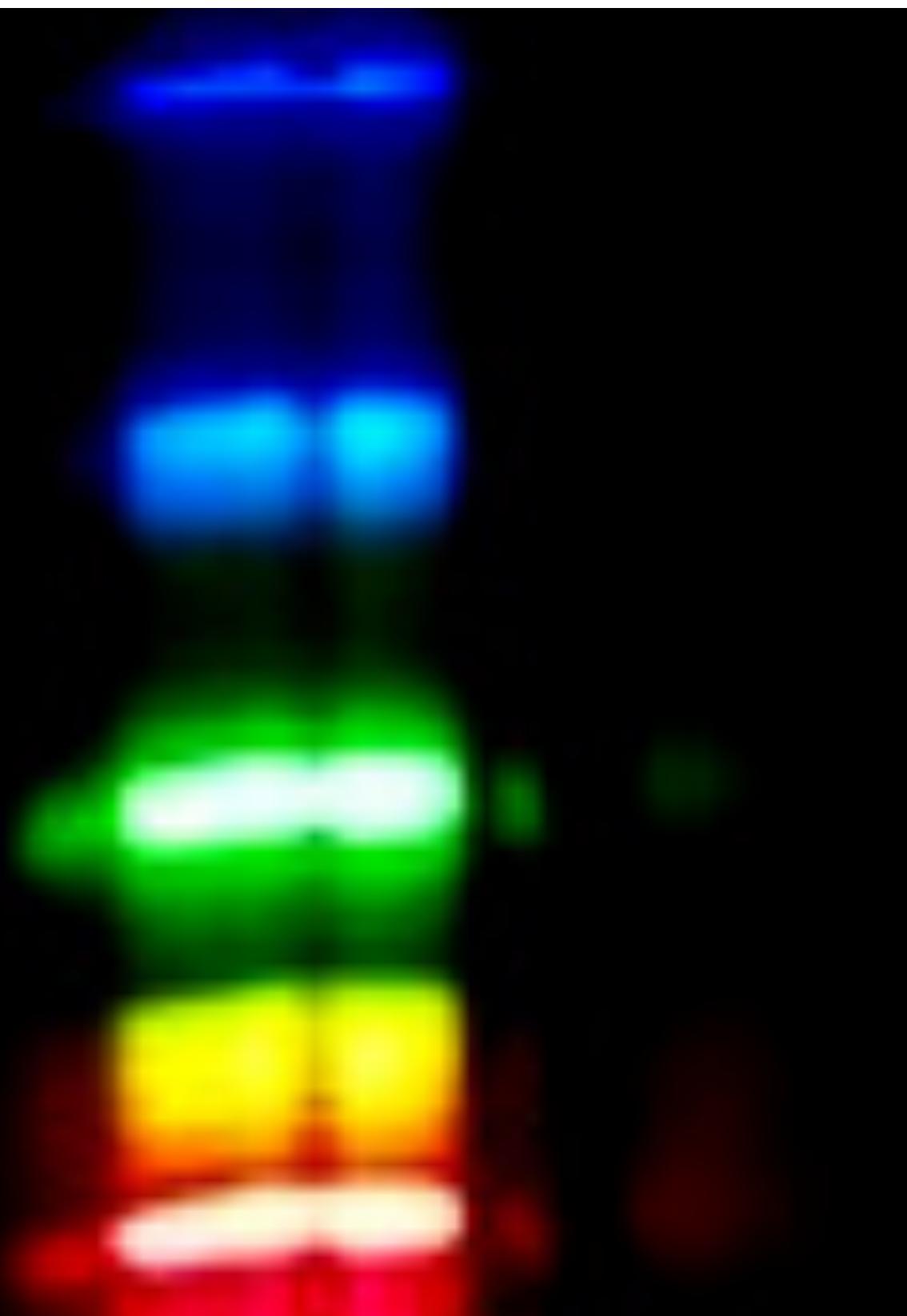
No comments

Post a comment

[JSON](#) [CSV](#) [CSV \(raw\)](#) [XML](#) [?](#)

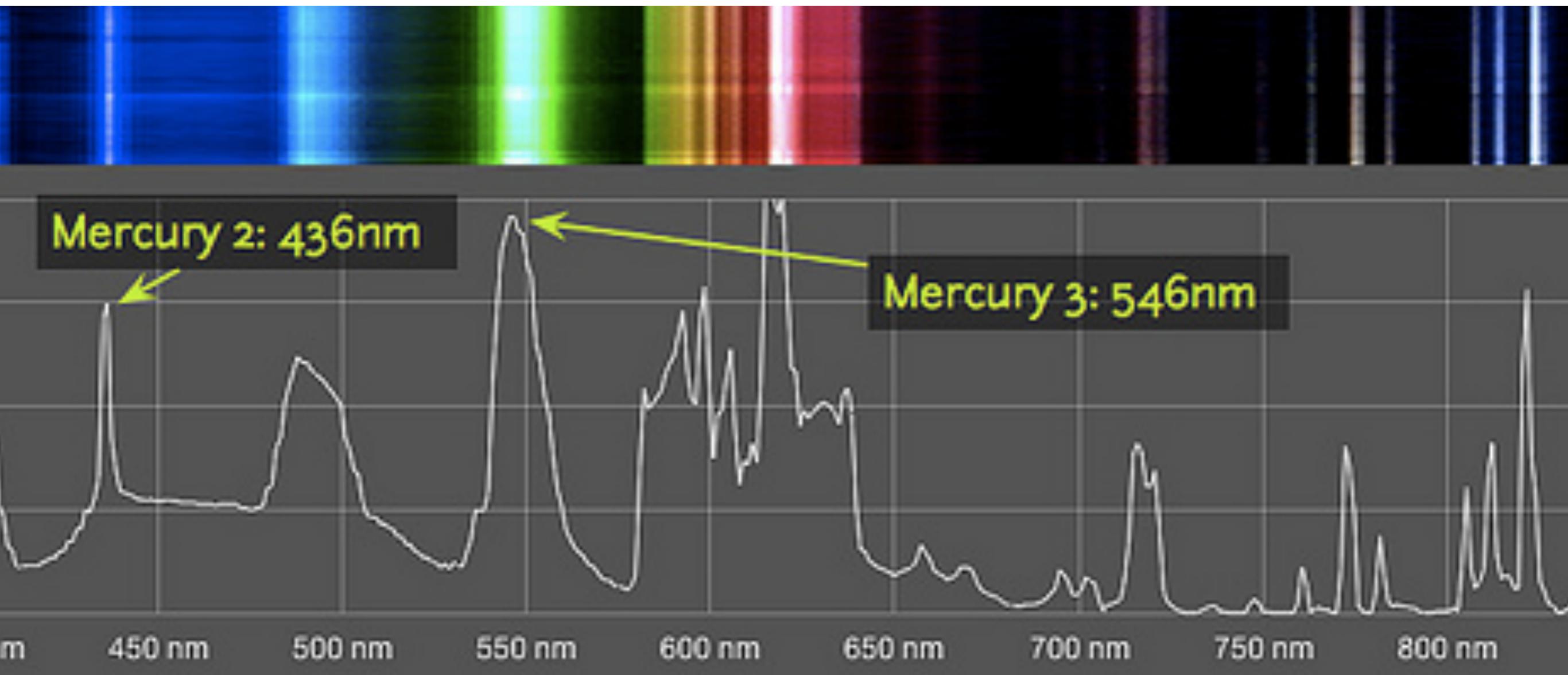


Calibration





Calibration





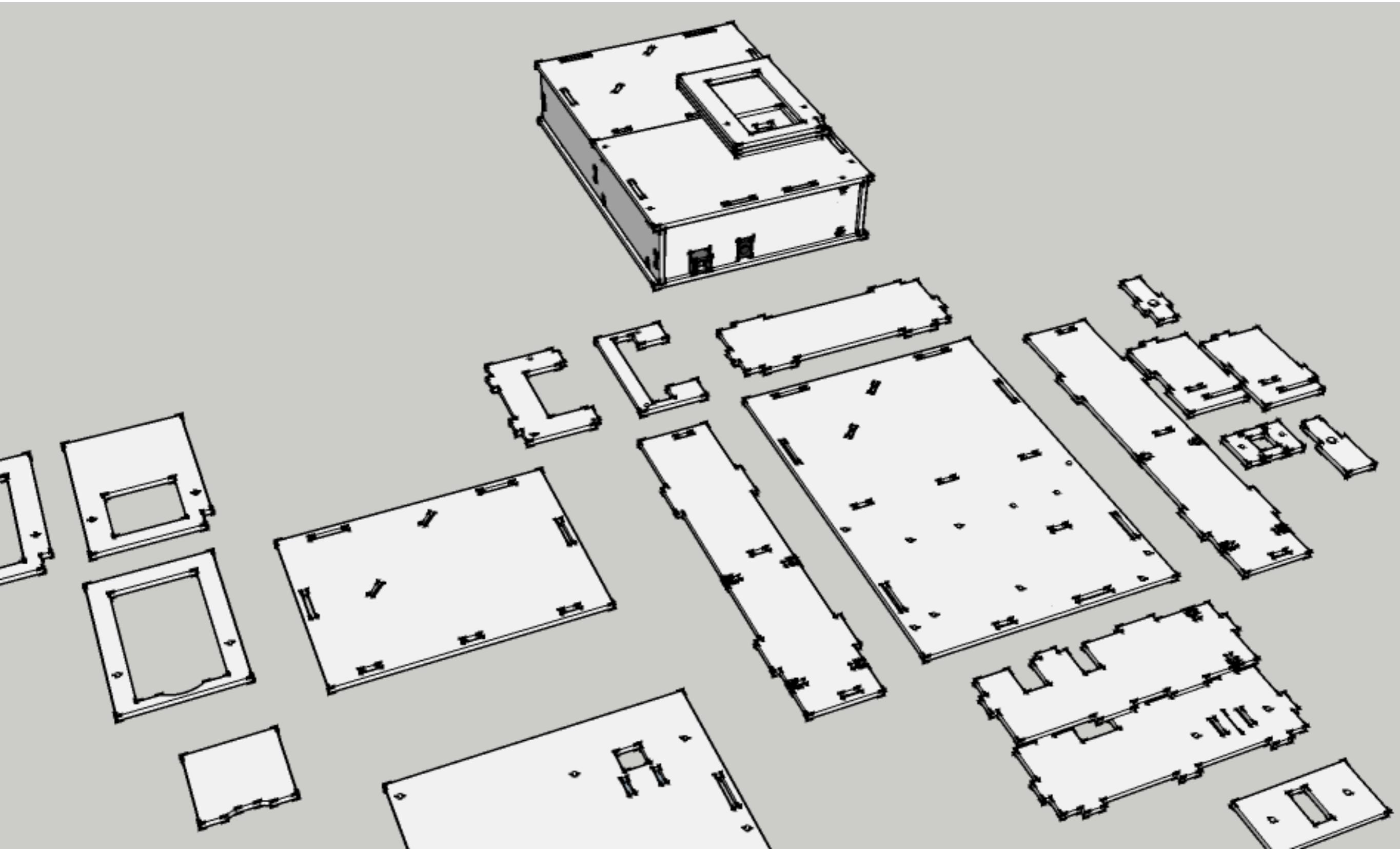
waag society

institute for art, science and technology

CCD Array Spectrometer

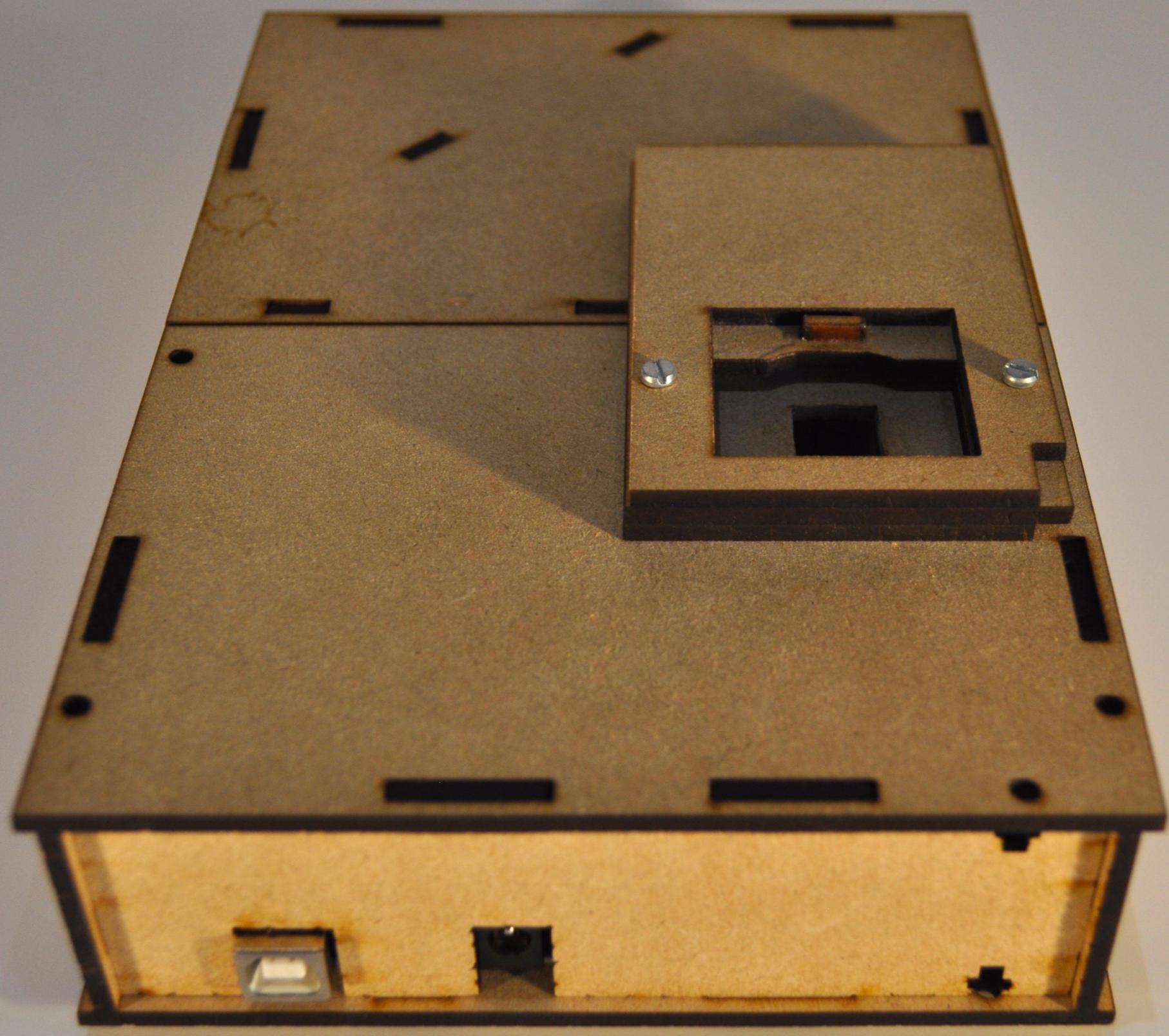


BioHack Academy design



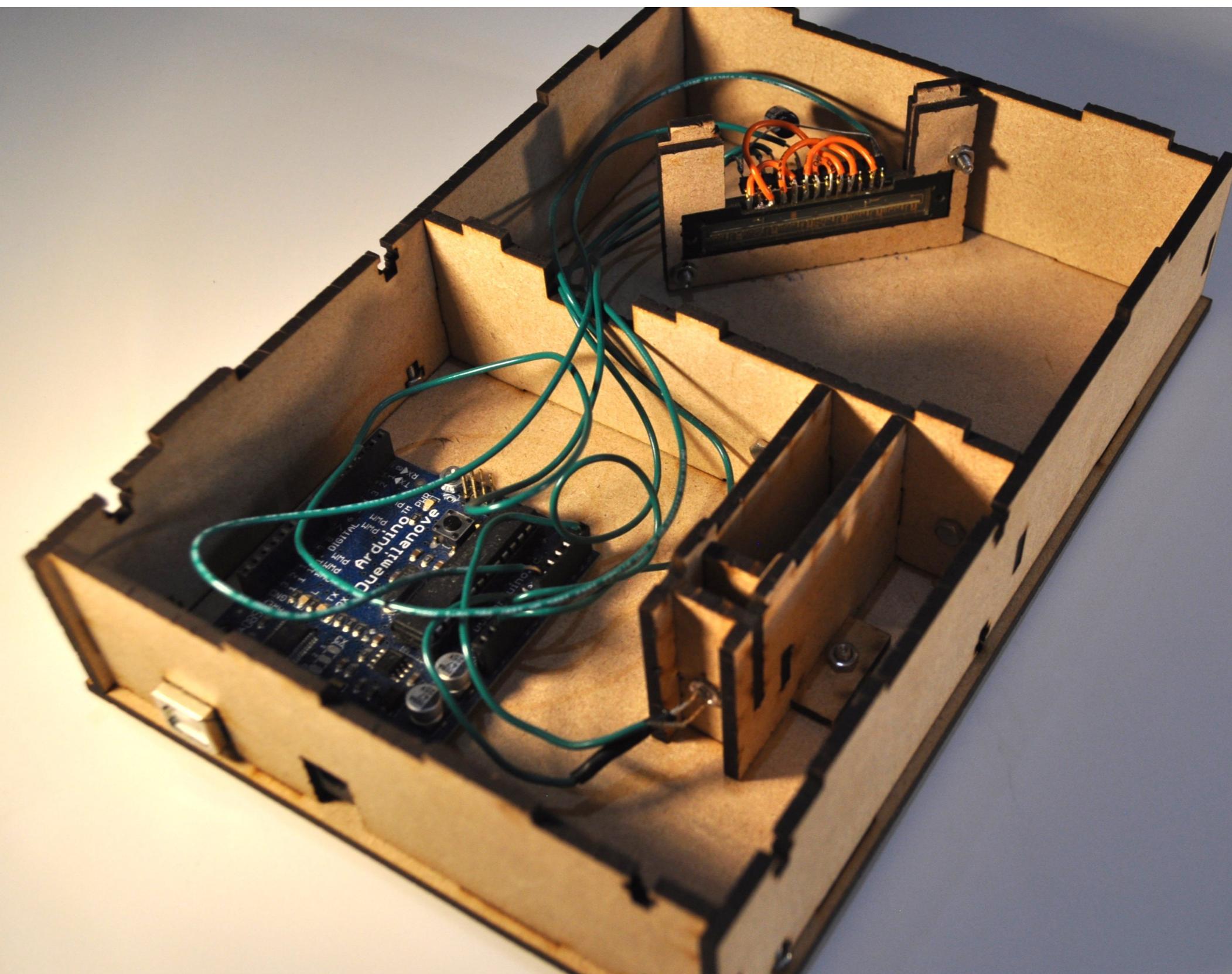


BioHack Academy design





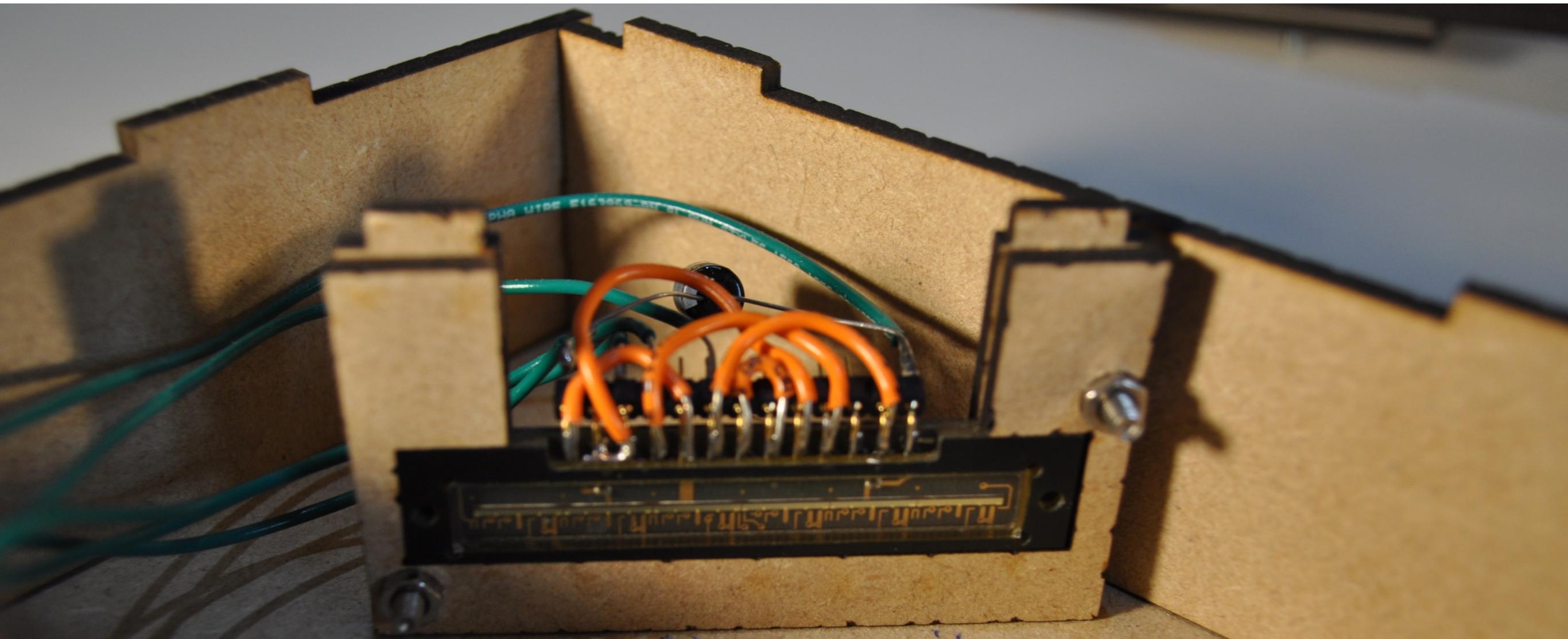
BioHack Academy design

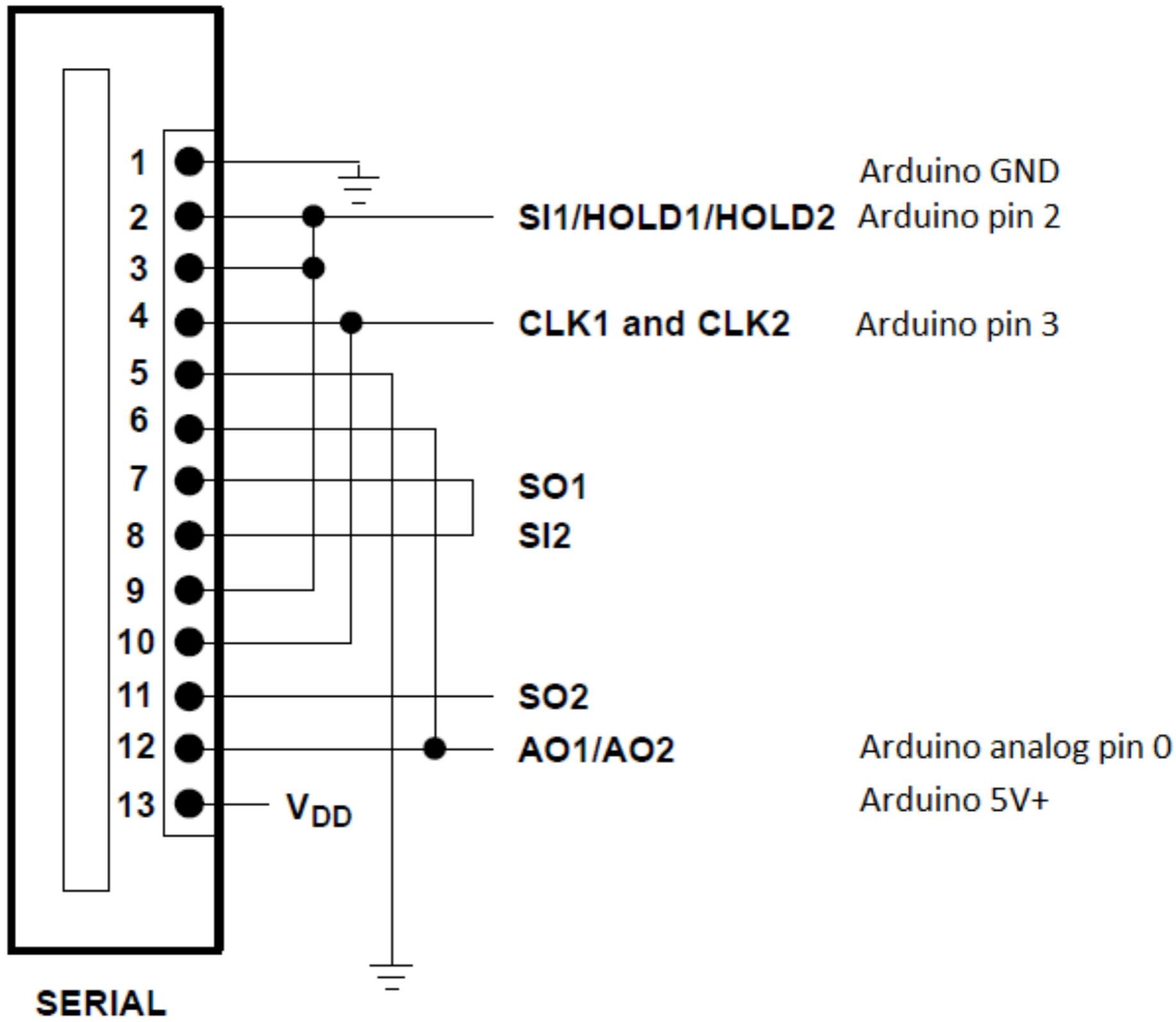




Bill of Materials

No	Amount	Description	Supplier NL	Cost
1	1	White 5mm LED	iPrototype, EOO, Hackerstore	0.18
2	1	TSL1406R Photo diode array	Mouser, DigiKey	22.57
3	1	1,000 lines/mm grating	ipacity, Carolina	14.00
4	1	100 uF capacitor	Farnell	0.14
5	4	220 ohm resistor	Farnell, iPrototype, EOO	0.02
6	1	Array of 13 right angle header pins	iPrototype	0.99
7	1	RGB LED	Farnell, iPrototype	0.95







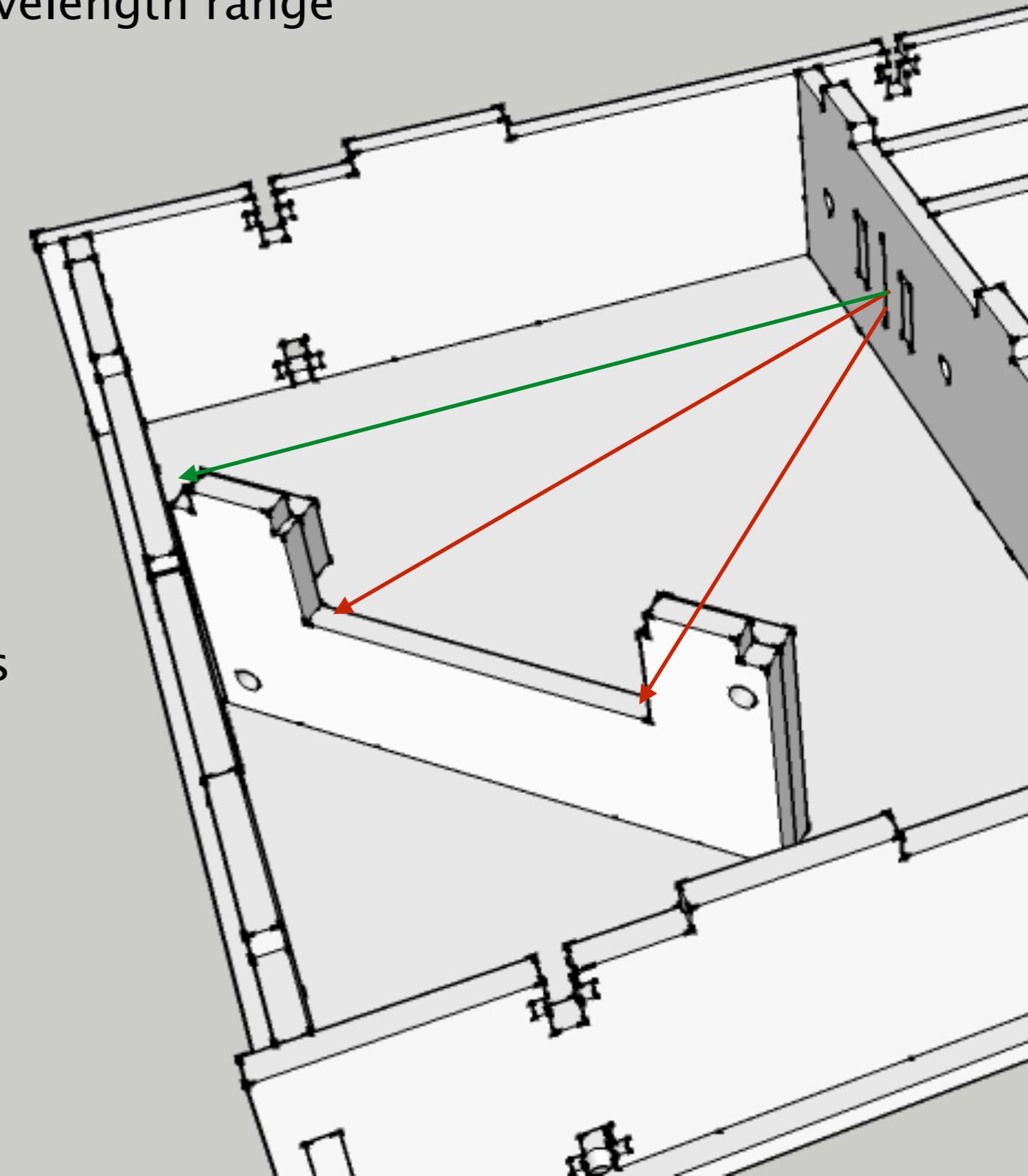
CCD array

- 400 nm to 900 nm target wavelength range
- d = grating 0.001 mm / line
- λ = wavelength
- ϕ = diffraction angle
- m = order of diffraction = 1

$$d \times \sin(\theta) = m \times \lambda$$

$$\theta = \sin^{-1}\left(\frac{m \times \lambda}{d}\right)$$

- 400 nm: $\phi = 23.58$ degrees
- 900 nm $\phi = 64.16$ degrees





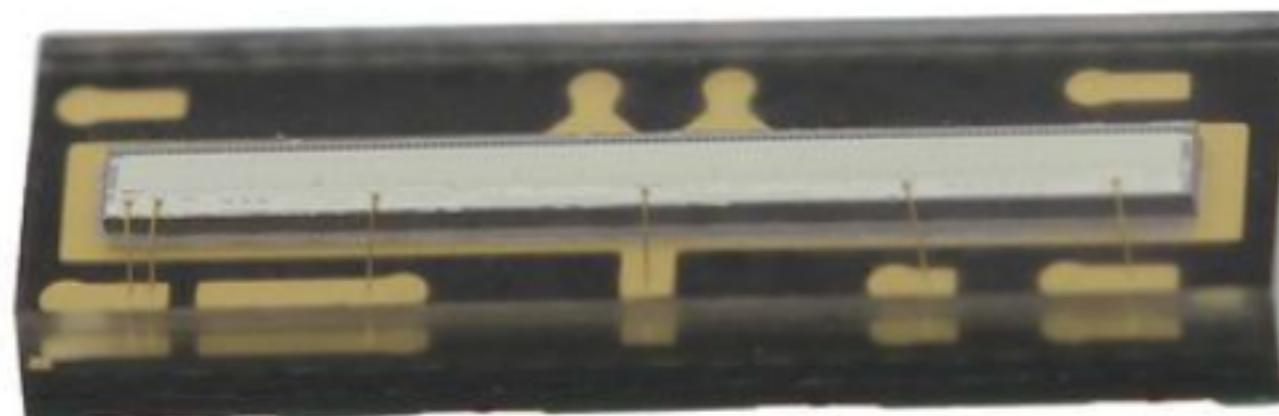
Alternative to TSL1406R?

**ams AG TSL1401CL Visible Light 128-Element Photo
1000nm, Surface Mount CL package**

RS-stocknr.: 785-7692

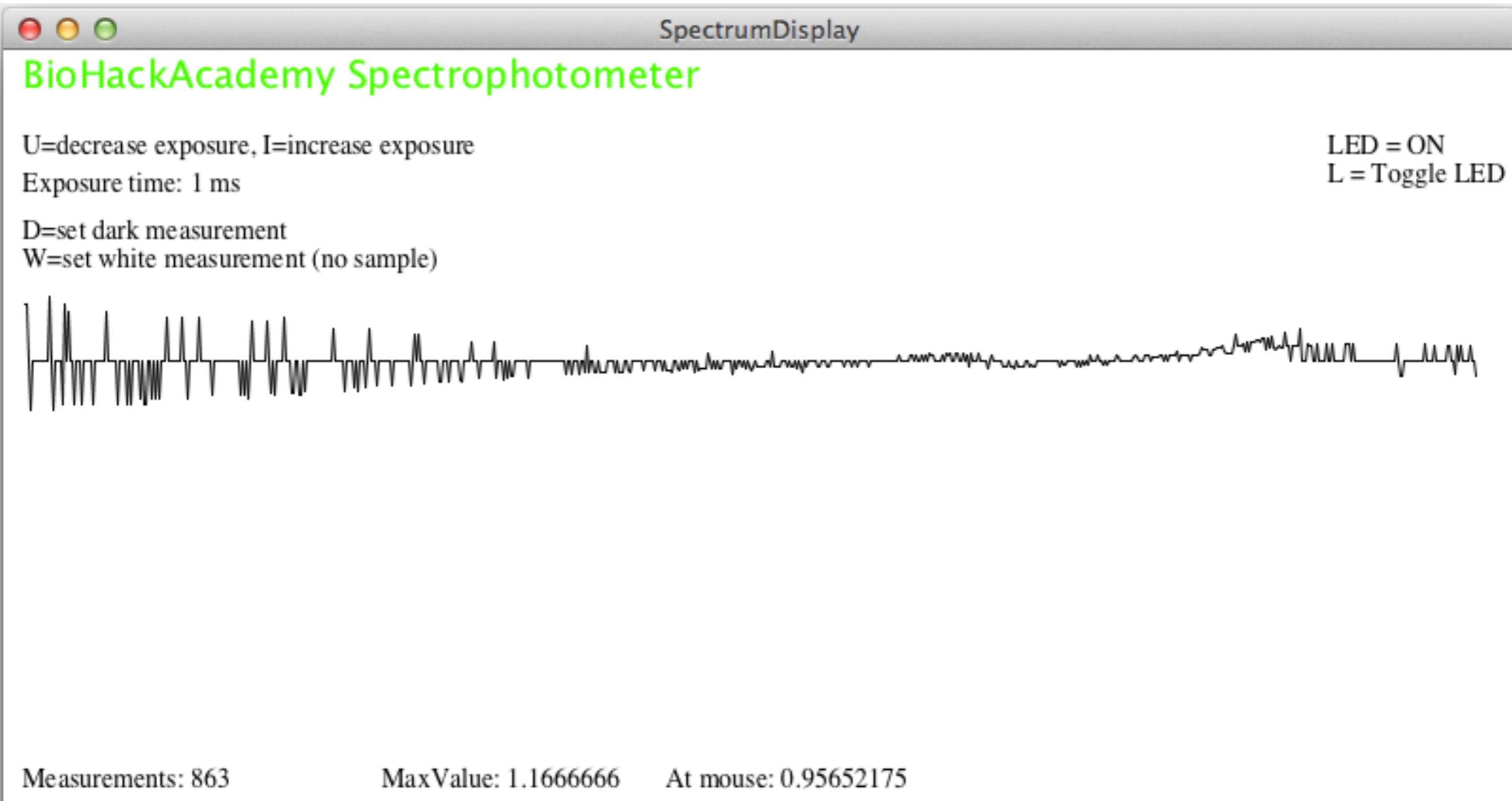
Fabrikantnummer: **TSL1401CL**

Fabrikant: **ams**





Processing Software – SpectrumDisplay





Source Code – setup

```
import processing.serial.*;

Serial port;      // The serial port
int exposureTime = 1; // 1 ms
float updateSpeed = 2000; // 1000 ms

int lastSpectrumUpdate=0;

boolean ledState=true;
PFont defaultFont;
PFont titleFont;

String buffer;

int SpectrumSize = 768;
float[] rawSpectrumData, correctedSpectrumData;
float[] darkReadout, whiteReadout;
int spectrumValueIndex=0;
int spectraCount=0;
```

```
void setup() {
    size(800, 400);
    // create a font with the third font available to the system:
    defaultFont = createFont(PFont.list()[0], 14);
    titleFont = createFont(PFont.list()[4], 20);
    textFont(defaultFont);

    // List all the available serial ports:
    printArray(Serial.list());

    String portName = Serial.list()[0];
    port = new Serial(this, portName, 57600);

    correctedSpectrumData = new float[SpectrumSize];
    rawSpectrumData = new float[SpectrumSize];
    darkReadout = new float[SpectrumSize];
    whiteReadout = new float[SpectrumSize];
    for(int i=0;i<SpectrumSize;i++)
        whiteReadout[i]=1.0f;

    lastSpectrumUpdate=millis();

    ledState=true;
    port.write("led 1\n");
}
```



Source Code – spectrum

```
void readSpectrum() {
    println("Updating spectrum");
    port.write("read\n");
}

void drawSpectrum() {
    int xstart=10, ystart=height-30;

    float maxVal=0.0f;
    for (int i=0;i<correctedSpectrumData.length;i++)
        maxVal = max(correctedSpectrumData[i],maxVal);

    float yscale = (height - 160)/maxVal;
    for (int i=1;i<correctedSpectrumData.length;i++) {
        line(i+xstart, ystart - correctedSpectrumData[i-1] * yscale,
              i+xstart+1, ystart - correctedSpectrumData[i] * yscale);
    }

    int indexAtMousePos = max(0, min(SpectrumSize-1, mouseX - xstart));
    text("At mouse: " + correctedSpectrumData[indexAtMousePos], 350, 39
    text("MaxValue: " + maxVal, 200, 390);
}

/*
Compute correctedSpectrumData based on rawSpectrumData, whiteReadout */
void computeSpectrum() {
    for (int i=0;i<SpectrumSize;i++) {
        correctedSpectrumData[i] = (rawSpectrumData[i] - darkReadout[i])
    }
}

void draw() {
    int time = millis();

    if (lastSpectrumUpdate + updateSpeed < time) {
        lastSpectrumUpdate=time;
        readSpectrum();
    }

    background(255);
    text("U=decrease exposure, I=increase exposure", 10, 55);
    text("Exposure time: " + exposureTime + " ms", 10, 75);
    text("D=set dark measurement", 10, 100);
    text("W=set white measurement (no sample)", 10, 115);

    text("Measurements: " + spectraCount, 10, 390);

    text("LED = " + ( ledState ? "ON" : "OFF" ), 700, 55);
    text("L = Toggle LED", 700, 70);

    textAlign(CENTER);
    textFont(titleFont);
    fill(0, 255, 0);
    text("BioHackAcademy Spectrophotometer", 10, 20);
    textFont(defaultFont);
    fill(0);

    drawSpectrum();
}
```



Source Code – interaction

```
void serialEvent(Serial port) {
    while (port.available () >0) {
        char c = port.readChar();
        if (c == '\n' && buffer.length() > 0) {
            char first=buffer.charAt(0);
            if (first >= '0' && first <= '9') {
                int value = Integer.parseInt(buffer.trim()); //substring(0,bu
                rawSpectrumData[spectrumValueIndex++] = value/1024.0f;
                if (spectrumValueIndex==SpectrumSize) {
                    computeSpectrum();
                    spectrumValueIndex=0;
                    spectraCount++;
                }
            } else if(buffer=="start") {
                spectrumValueIndex=0; // align again
            }
            buffer="";
        } else {
            buffer+=c;
        }
    }
}

void keyPressed() {
    if (key >= 'A' && key <= 'Z')
        key += 'a'-'A'; // make lowercase

    if (key == 'i' || key == 'u') {
        if (key == 'i') exposureTime += 3;
        else exposureTime -= 3;
        exposureTime = max(exposureTime,1);

        port.write("exp " + max(1, (int)exposureTime) + "\n");
    }

    if (key == 'l') {
        ledState=!ledState;
        port.write("led " + ( ledState ? "1" : "0") + "\n");
    }

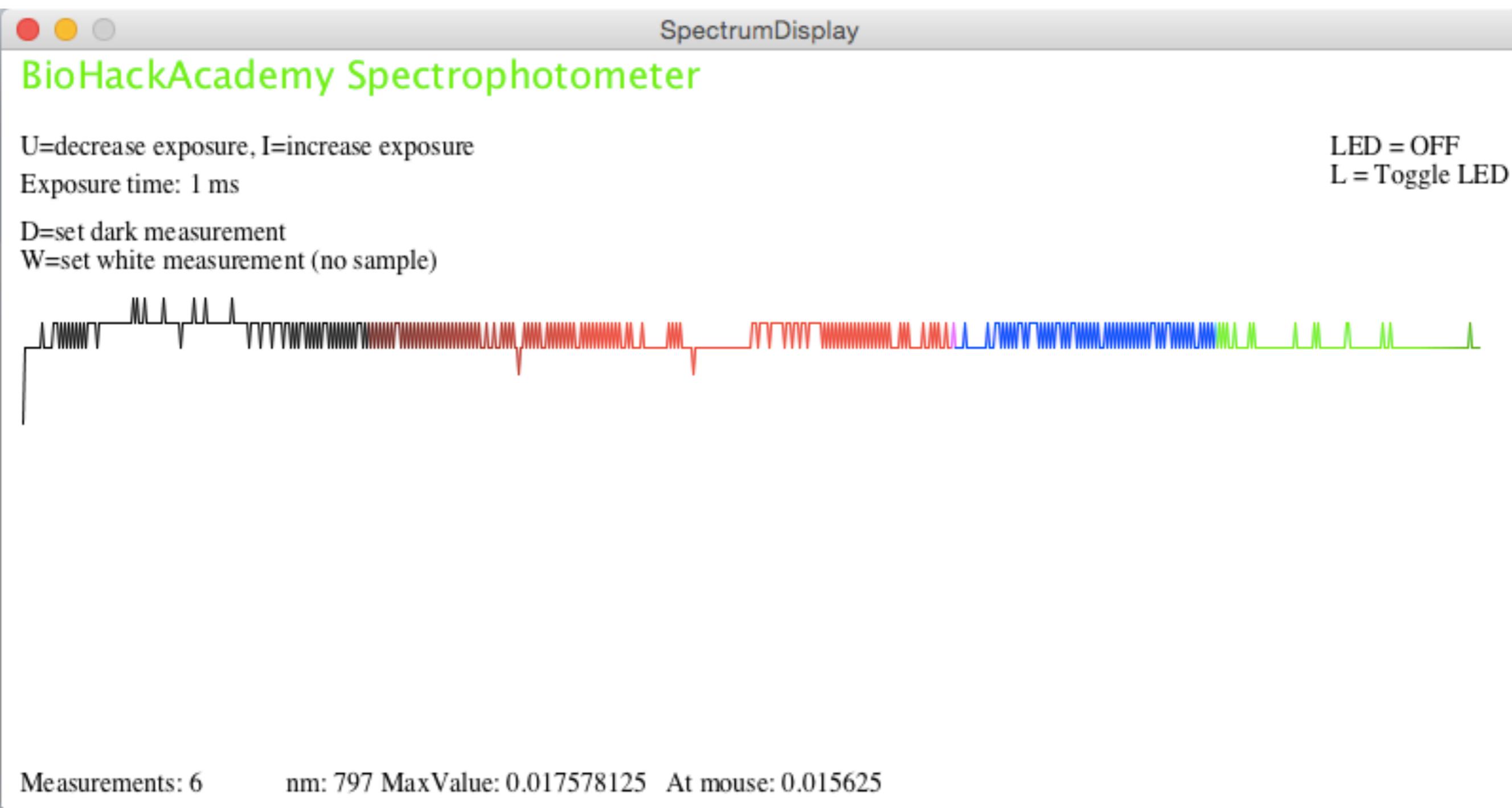
    if (key == ' ') {
        readSpectrum();
    }

    if (key == 'd') {
        for (int i=0;i<SpectrumSize;i++)
            darkReadout[i] = rawSpectrumData[i];
    }

    if (key == 'w') {
        for (int i=0;i<SpectrumSize;i++)
            whiteReadout[i] = rawSpectrumData[i];
    }
}
```



Processing Software – Empty





Red food dye

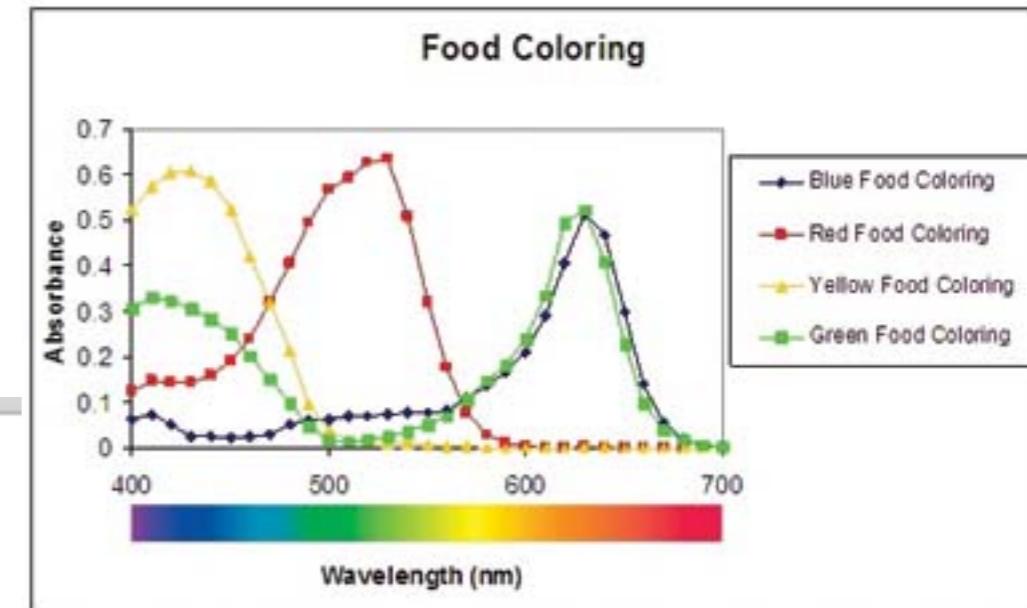
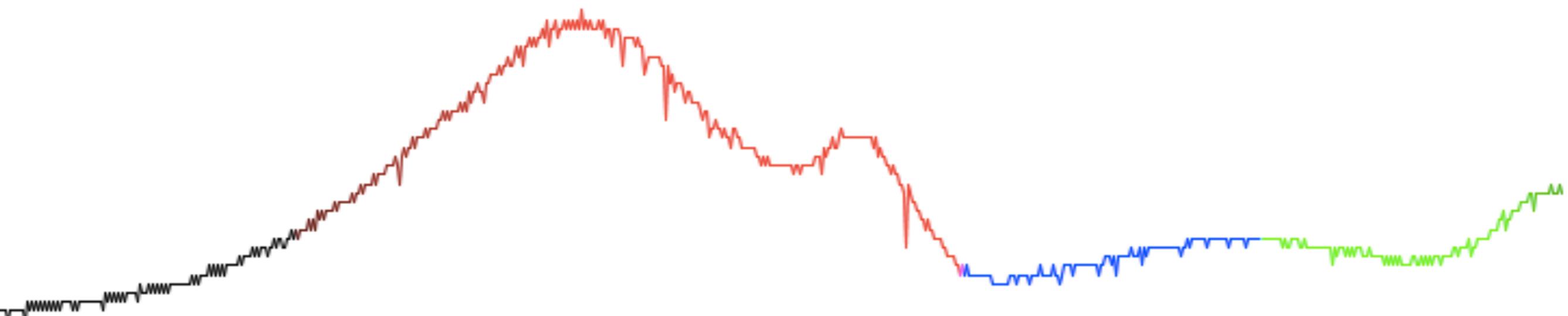
JackAcademy Spectrophotometer

↑=increase exposure, I=increase exposure

Exposure time: 1 ms

Dark measurement

White measurement (no sample)



L = Toggle LED



Green food dye

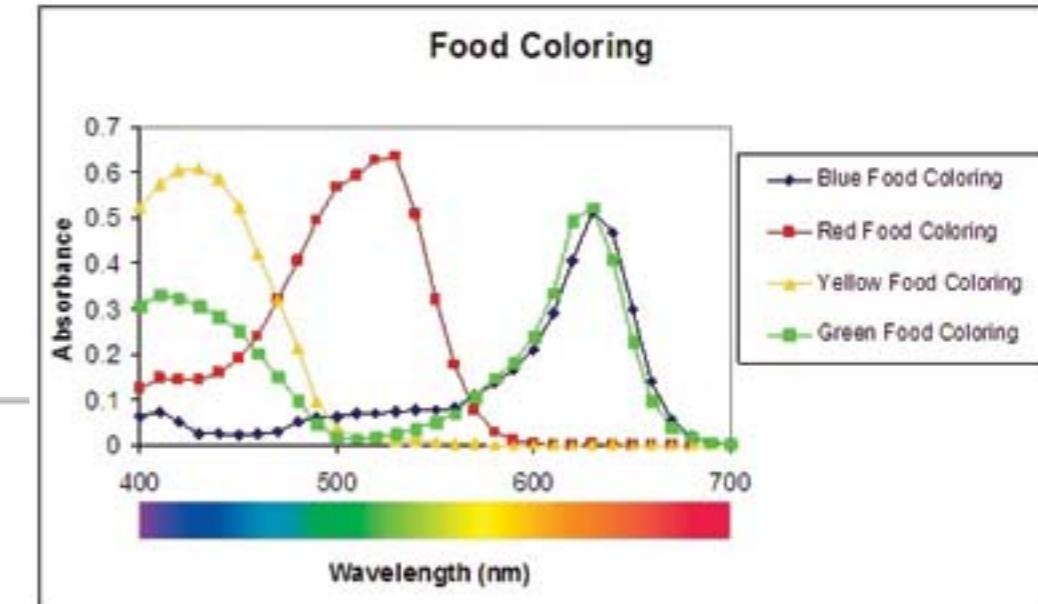
ackAcademy Spectrophotometer

ease exposure, I=increase exposure

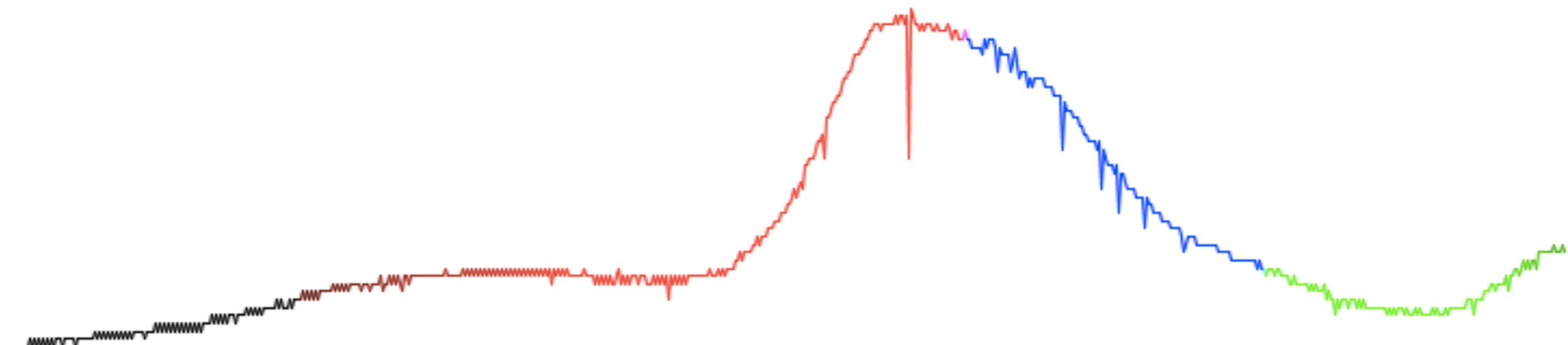
re time: 1 ms

ark measurement

white measurement (no sample)



L = Toggle LED





**some
rights
reserved**