



waag society

institute for art, science and technology

Waag Society – CC-BY-SA 3.0



BioHack Academy Bioreactor Controller Design



Biohacking



Center for Genomic Gastronomy- CC-BY-SA 3.0



Malthe M Borch- CC-BY-SA 3.0



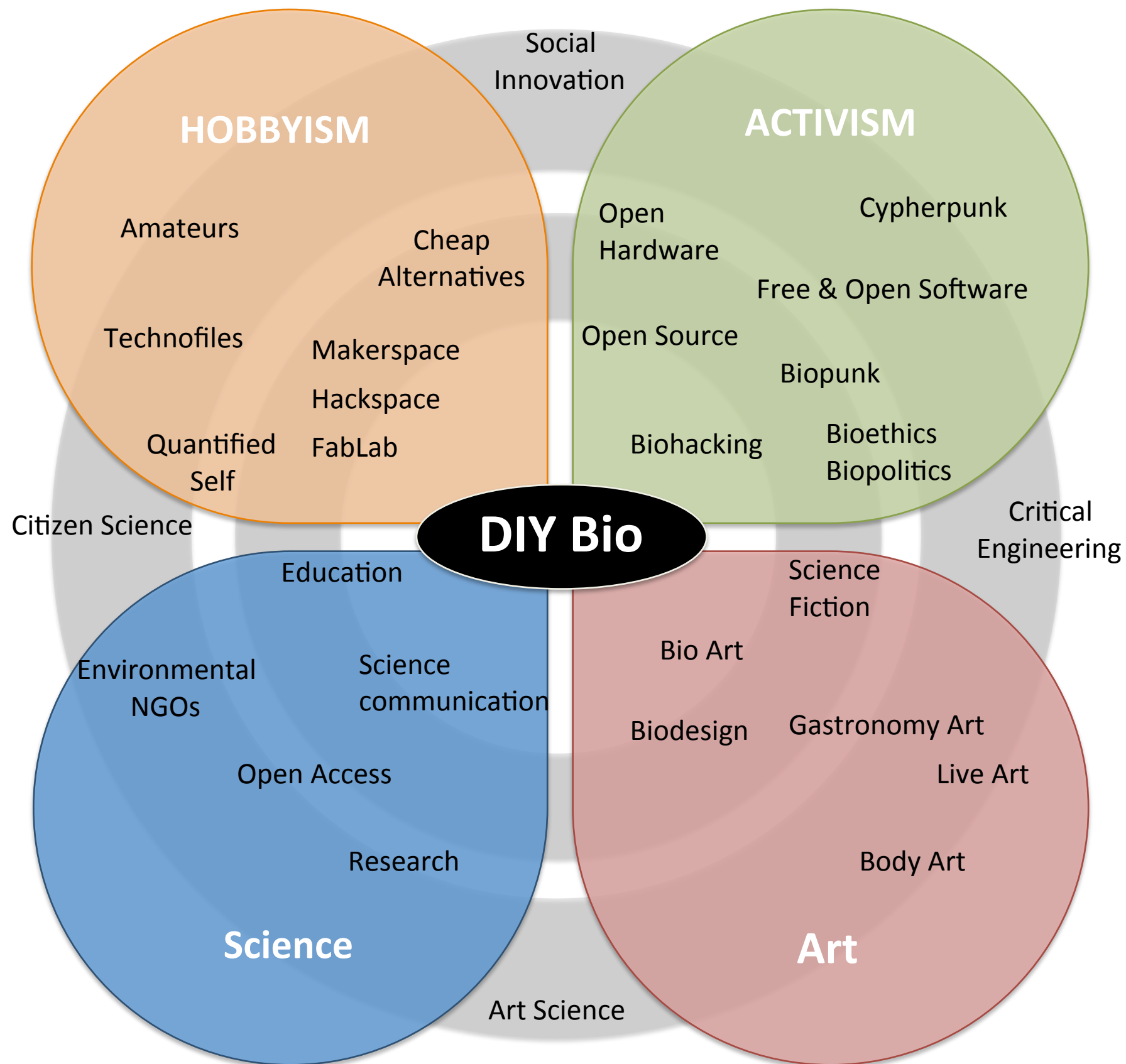
Future Everything- CC-BY-SA 3.0



Waag Society - CC-BY-SA 4.0



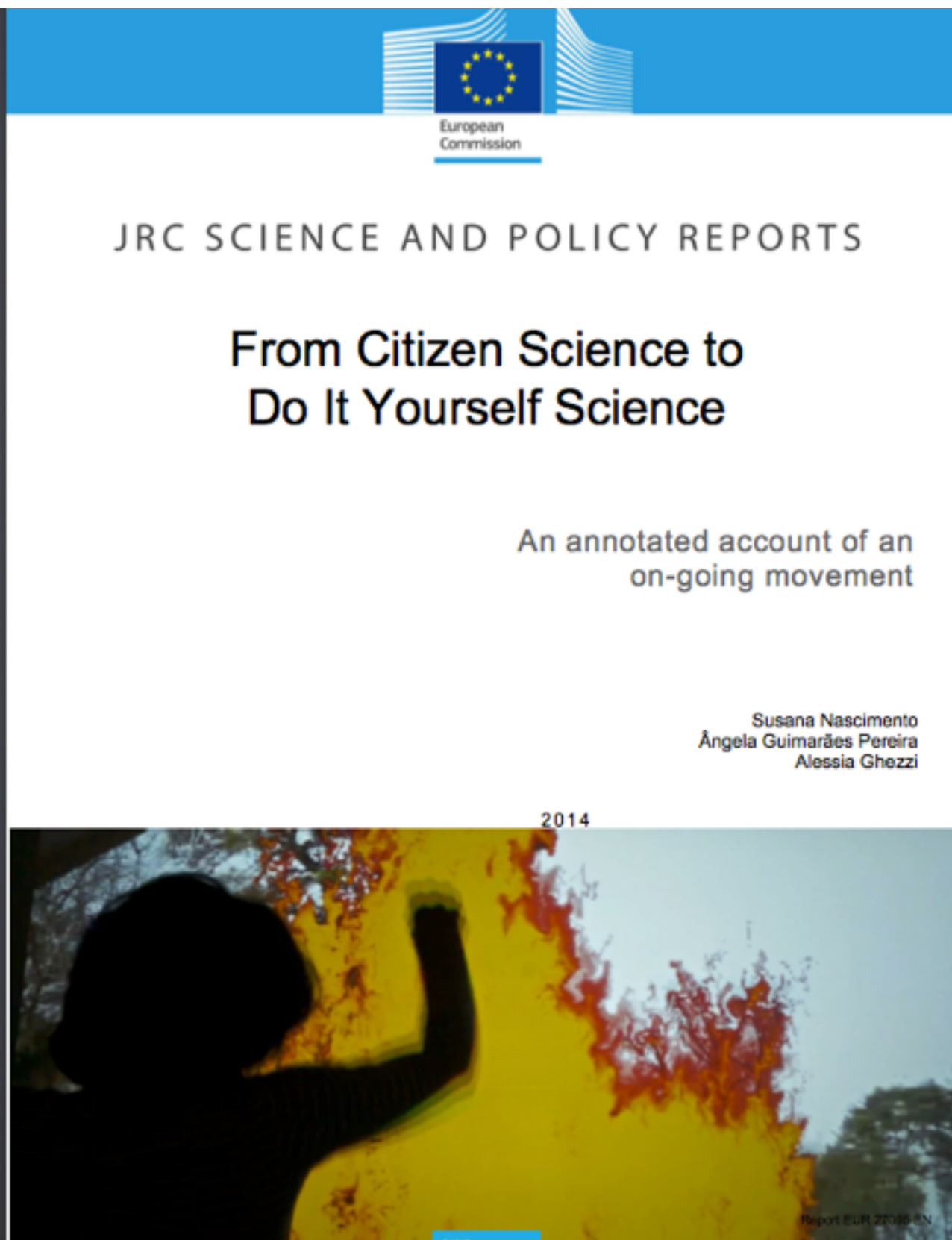
What is DIYBio?





Report of Citizen /DIY Science

<http://publications.jrc.ec.europa.eu/repository/handle/JRC93942>





Note of caution





Possible improvements?

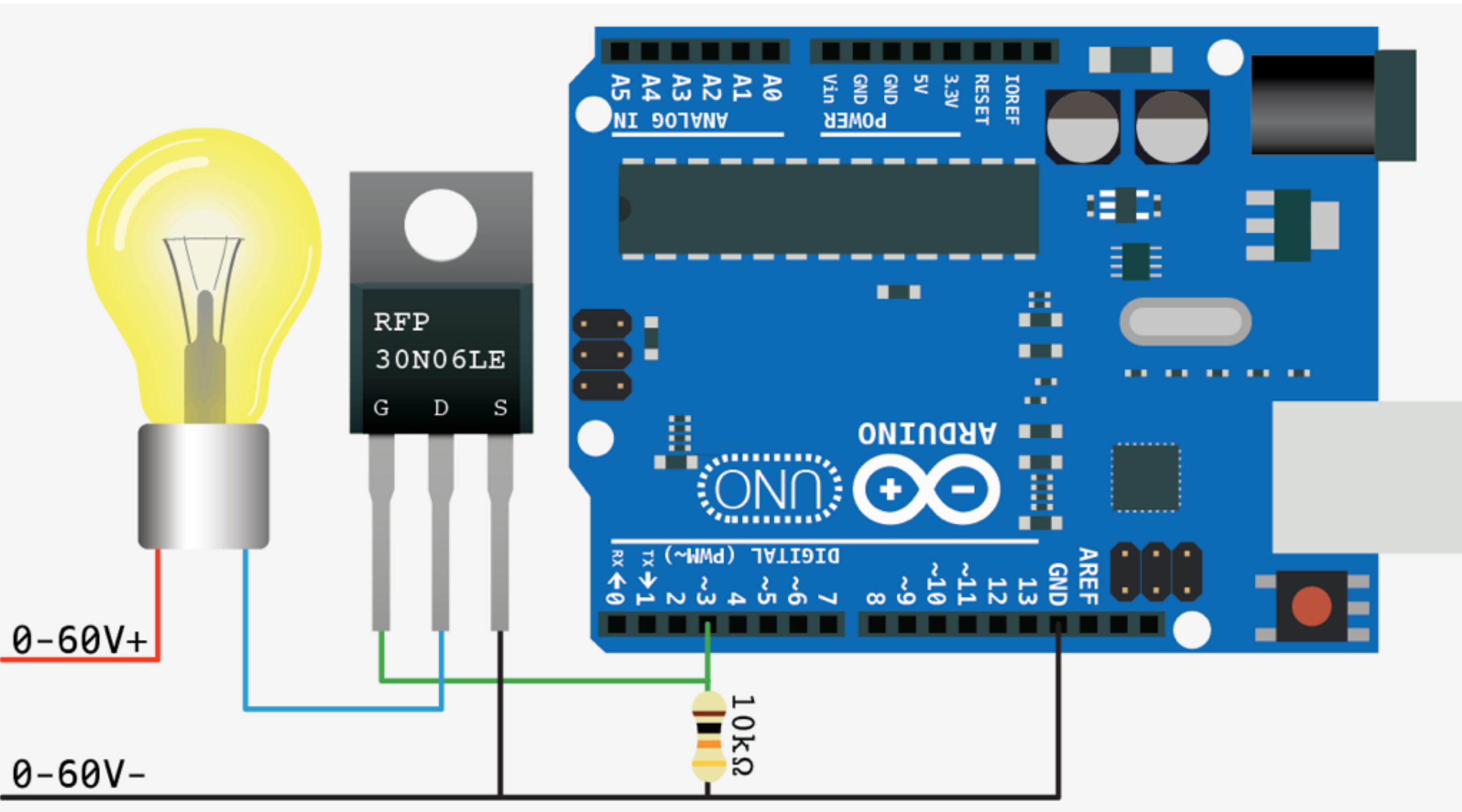
- Mechanical failure of relay?
- Use a power MOSFET instead?
- Solid state relay?
- Improve the code?





Power mosfet circuit

<http://bildr.org/2012/03/rfp30n06le-arduino/>

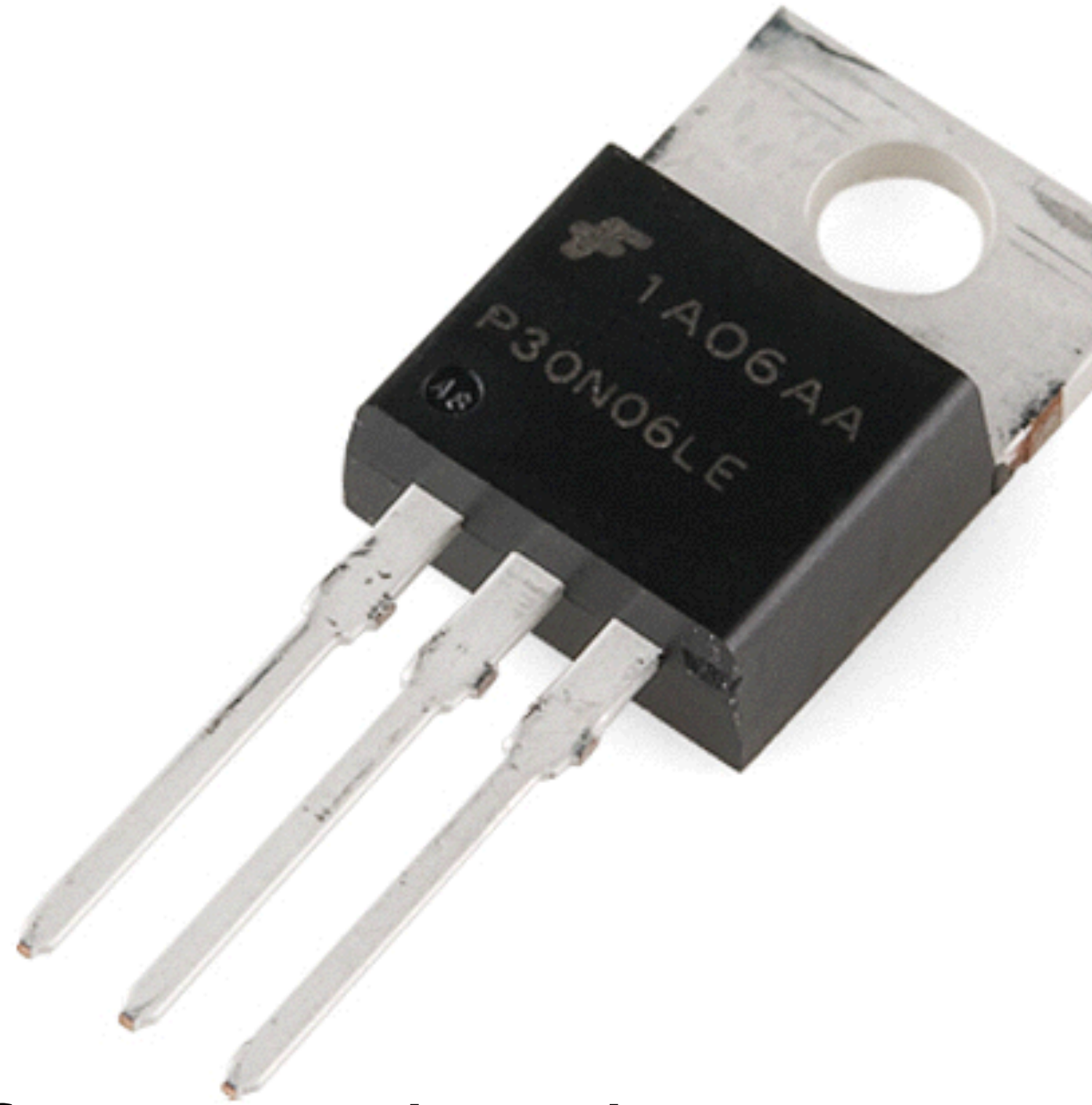




Power mosfet circuit

Important notes:

- RFP30N06LE
- Only DC
- $< 60\text{ V}$
- $< 30\text{ A}$

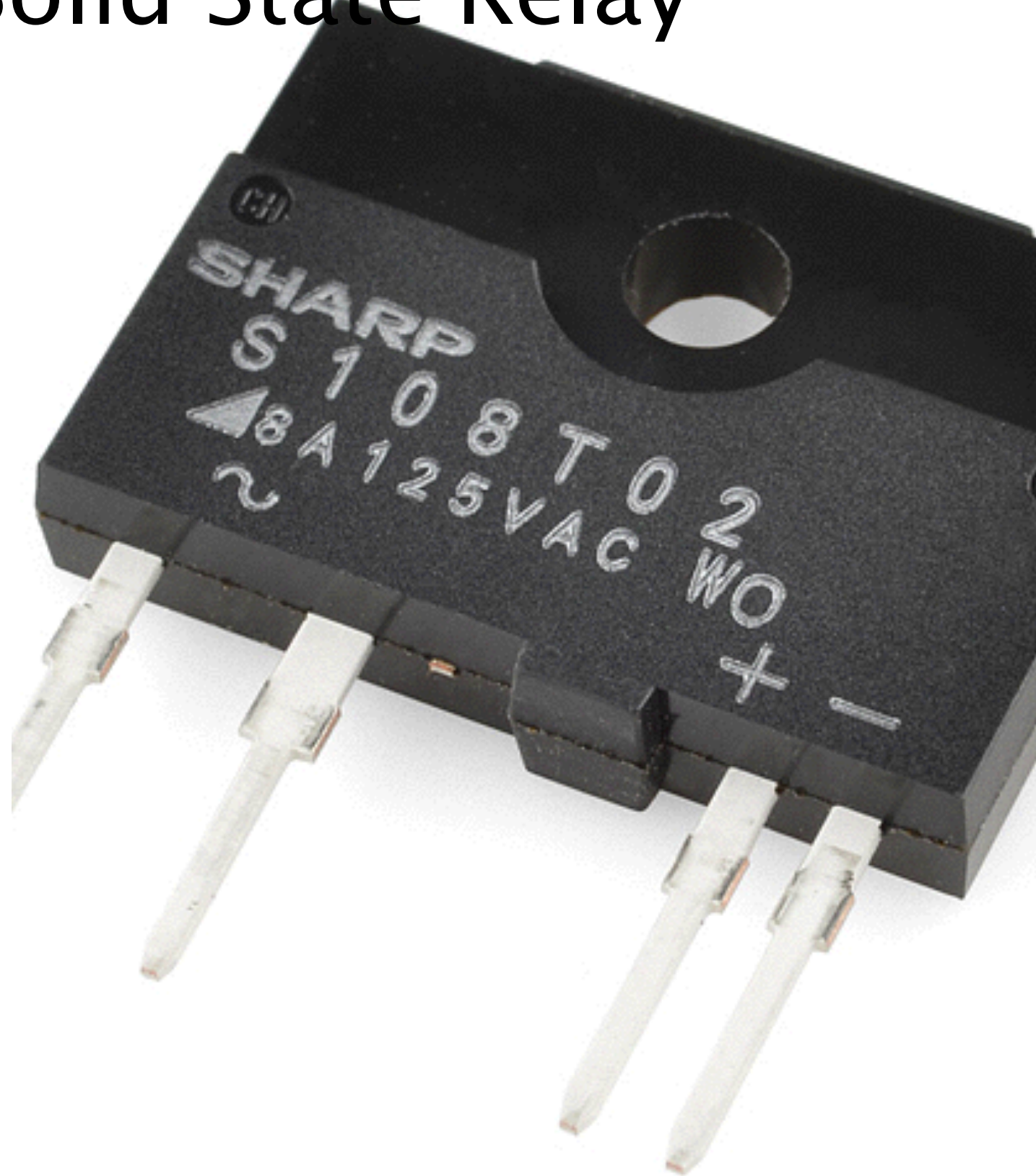


<https://www.sparkfun.com/datasheets/Components/General/RFP30N06LE.pdf>



Alternative 2: Solid State Relay

- S108T02
 - 400VAC
 - 8A





Fix the code

```
long switch_time = 3000;
[....]
if((millis() - begin_switch_time) > switch_time) {
    // Turn the 100W infrared light on or of depending on
    temperature
    if (temp < targetTemp) {
        digitalWrite(relayPin, HIGH);
        begin_switch_time = millis();
    }
    else {
        digitalWrite(relayPin, LOW);
        begin_switch_time = millis();
    }
}
[....]
```



waag society

institute for art, science and technology

Bioreactor Controller



Industry standard

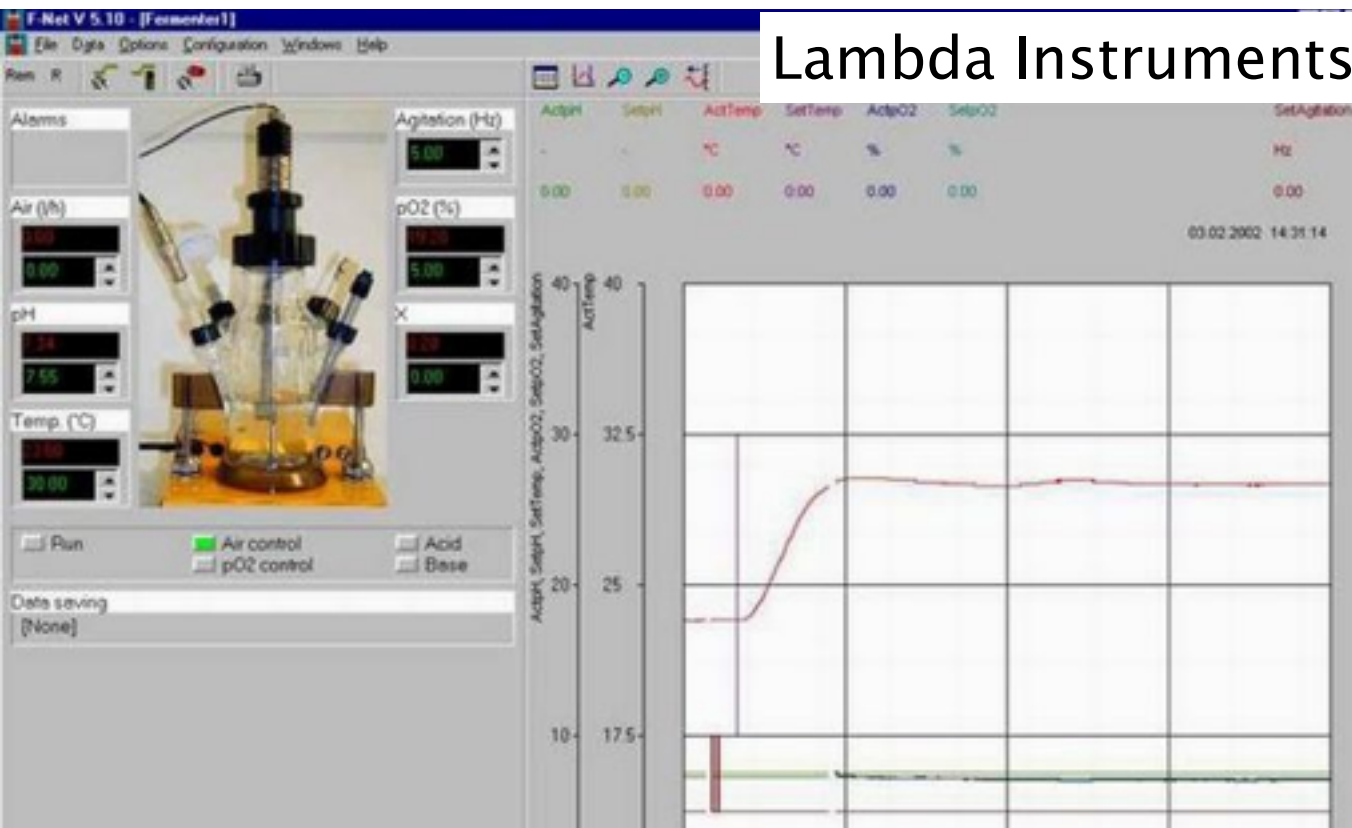
Applikon



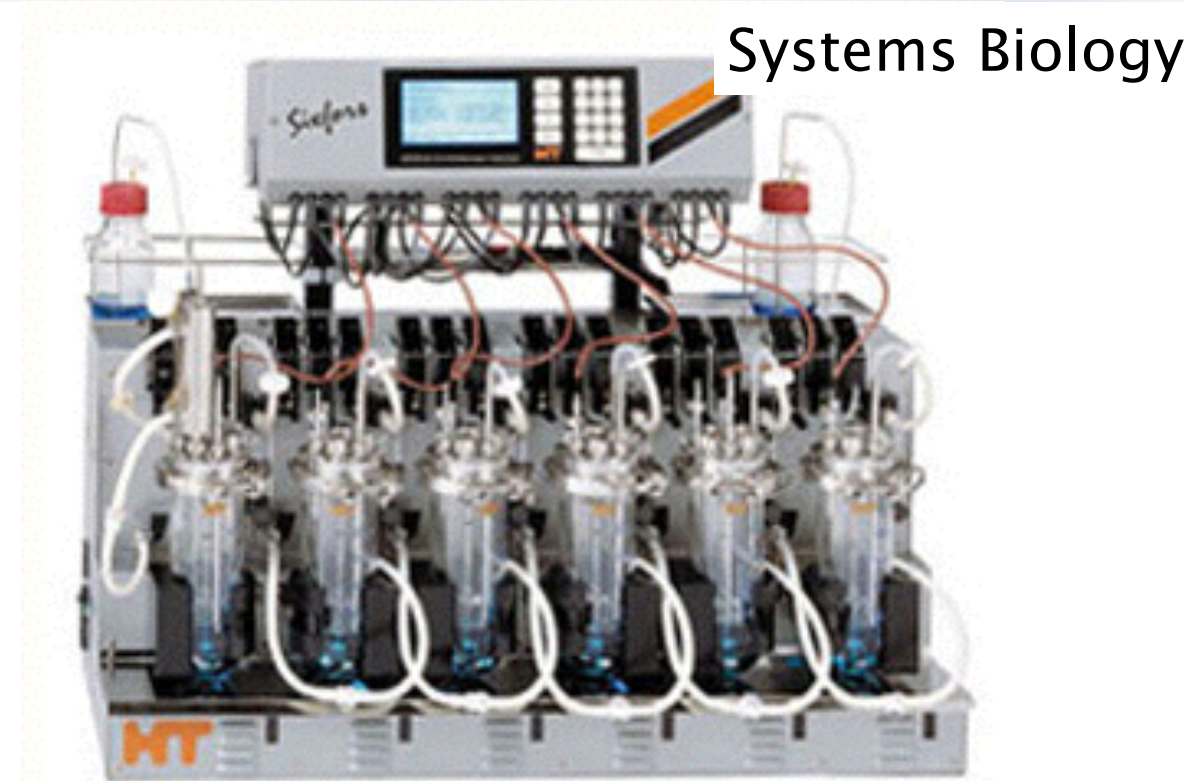
Medorex

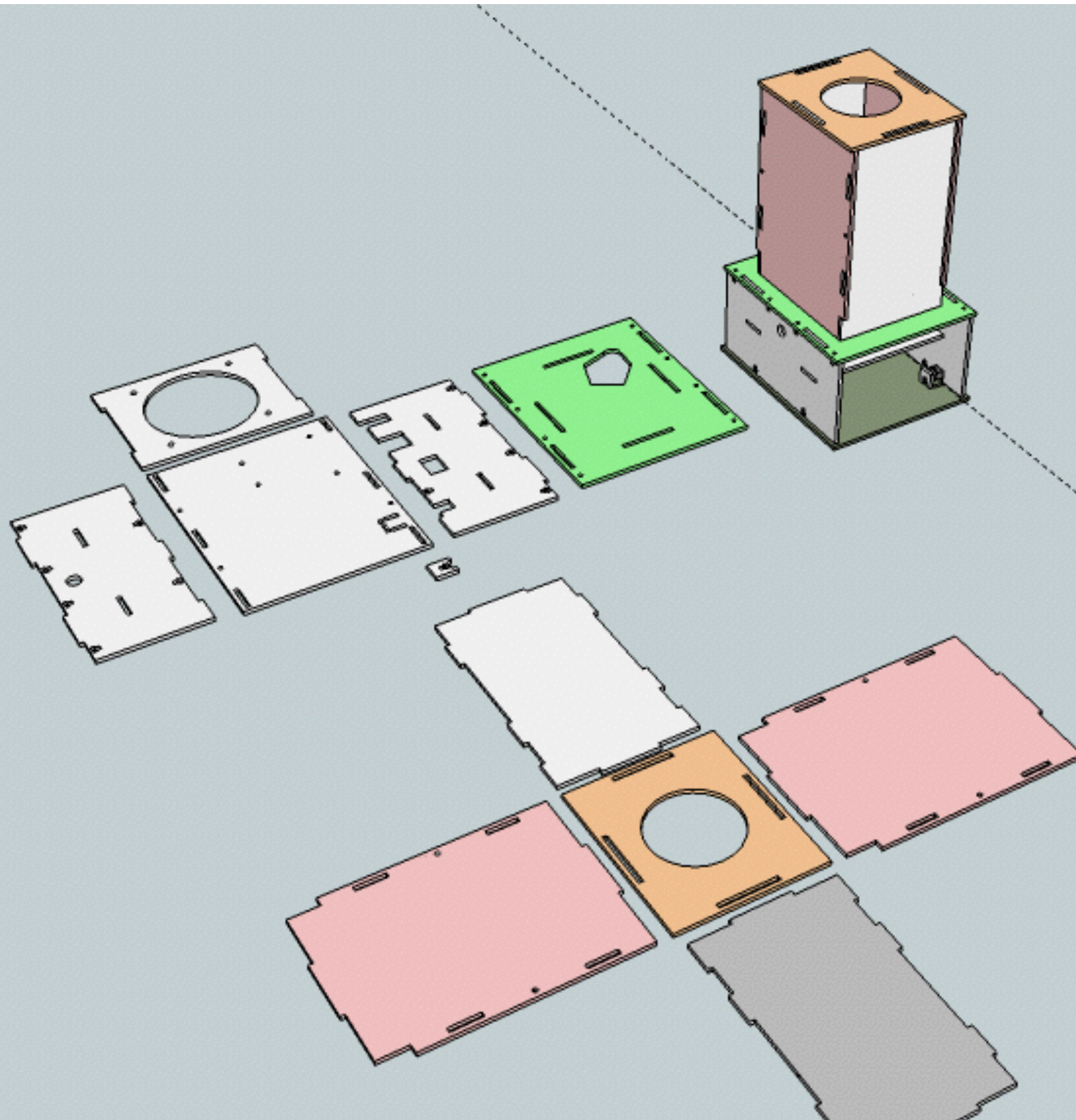


Lambda Instruments



Systems Biology





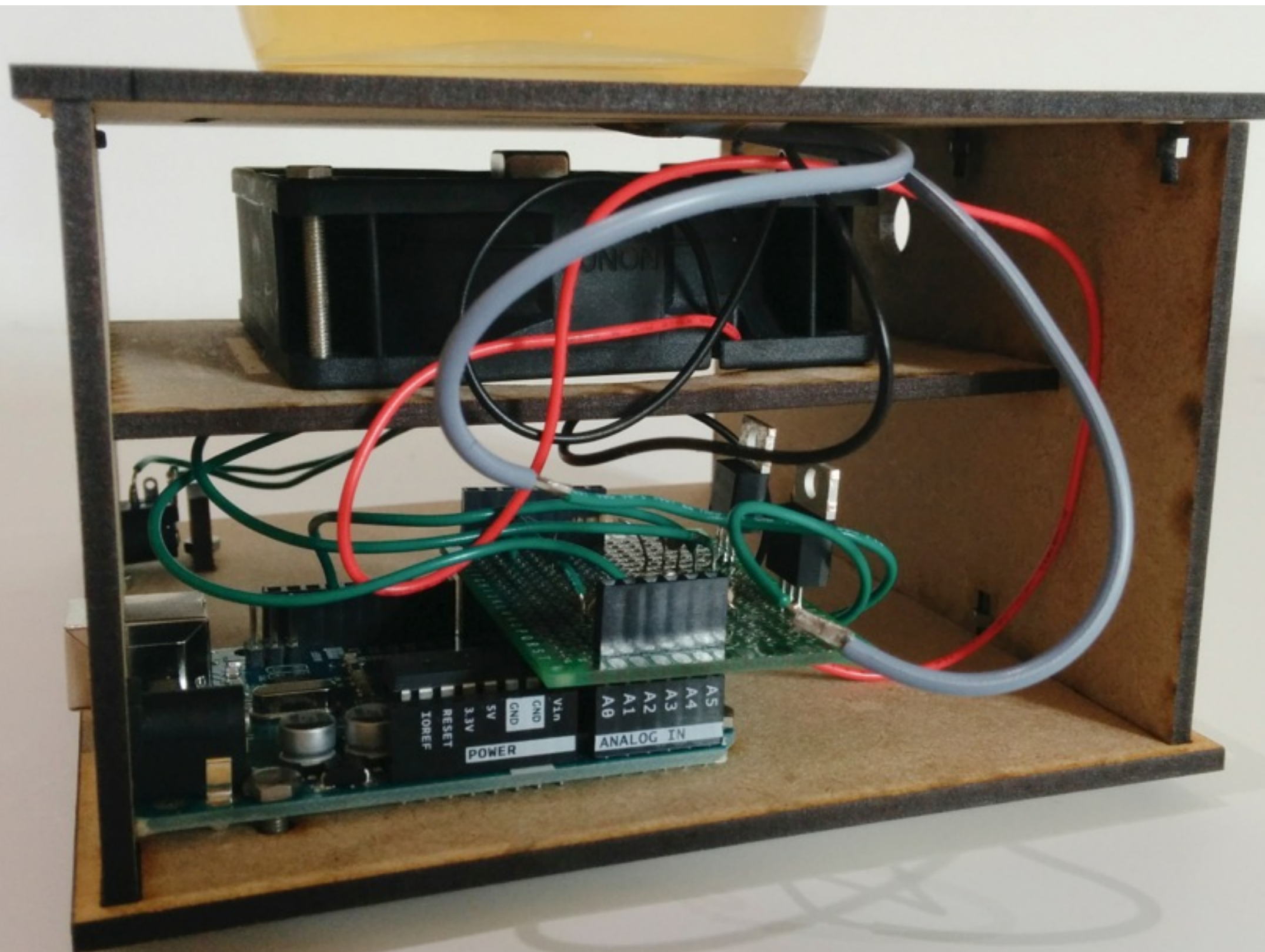


BioHack Academy Design





BioHack Academy design



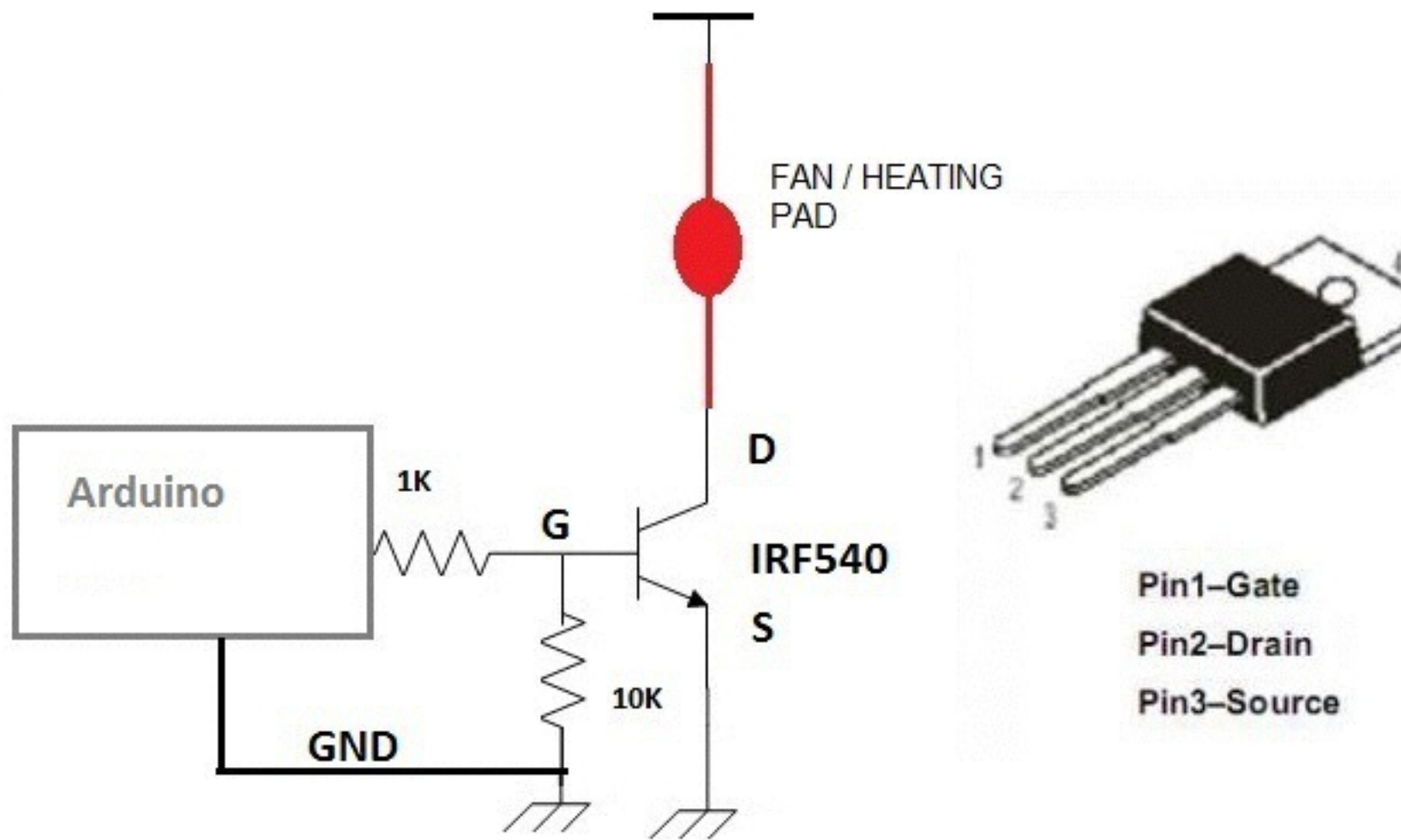


Bill of Materials

#	Amount	Description
1		1 Heating pad
2		1 Fan
3		2 Permanent magnets
4		2 MOSFET
5		2 10K resistor
6		2 1K resistor
7		1 OneWire Thermometer



Component Wiring





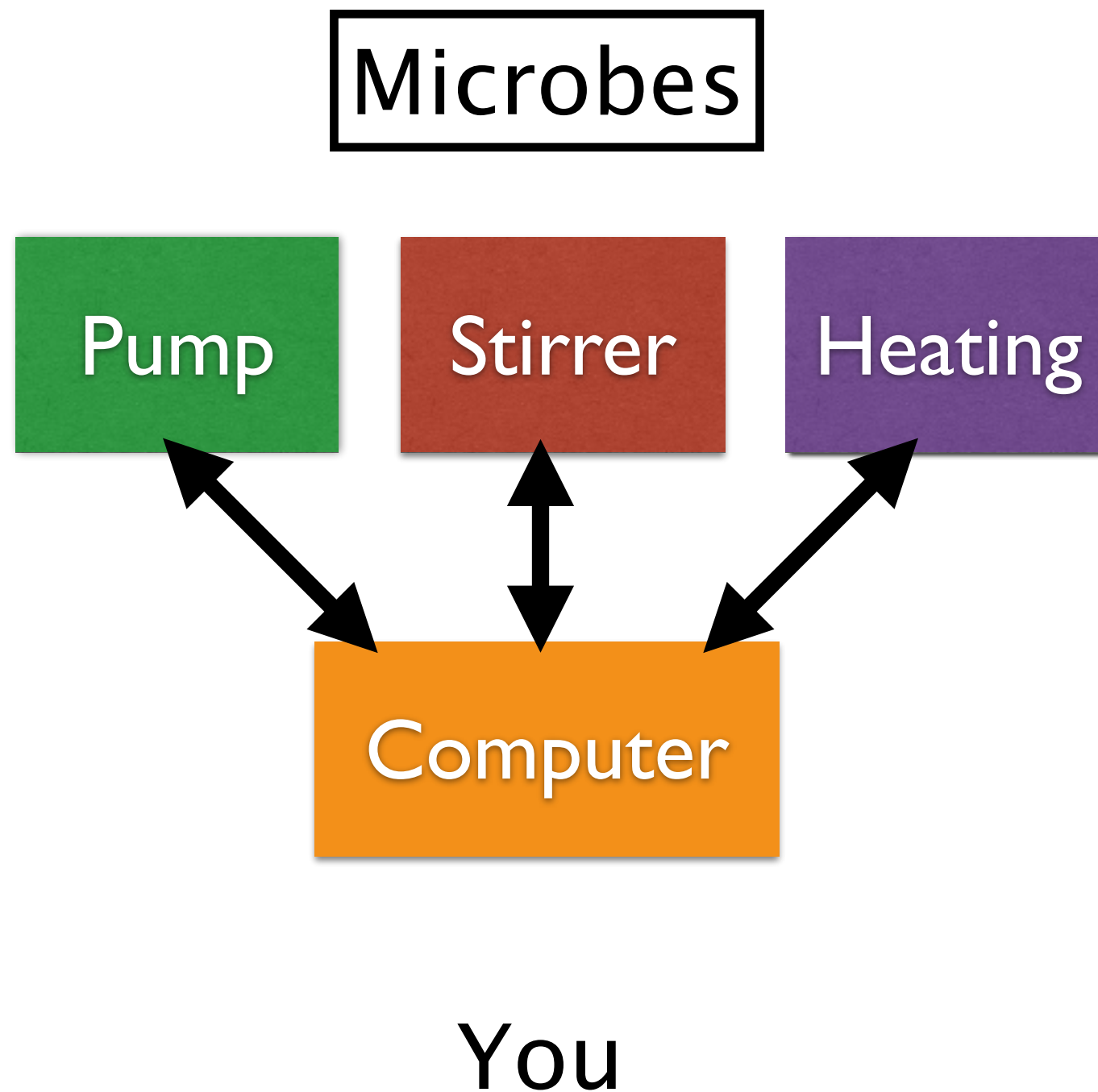
waag society

institute for art, science and technology

Coding

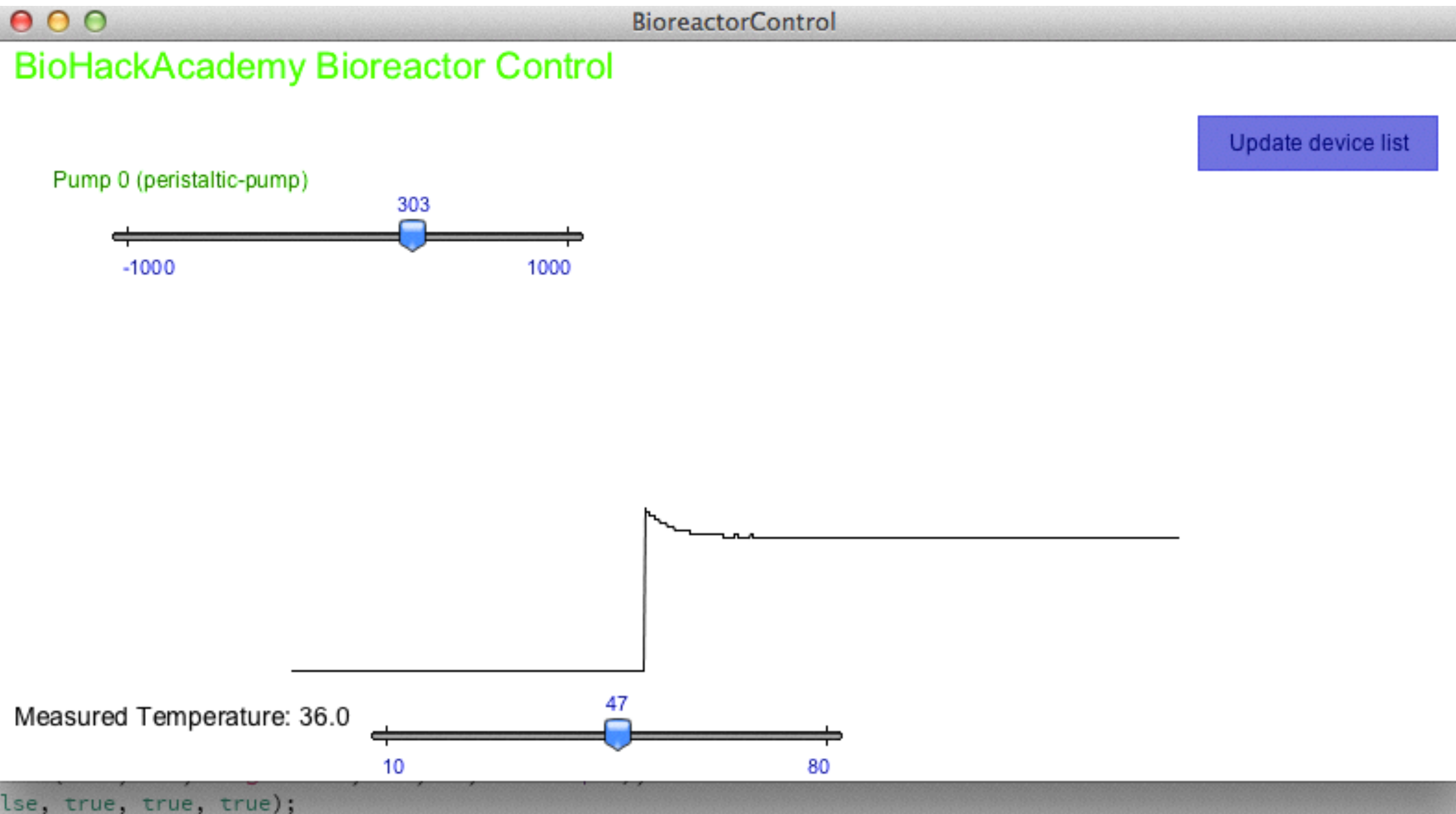


Connectivity diagram





Processing interface





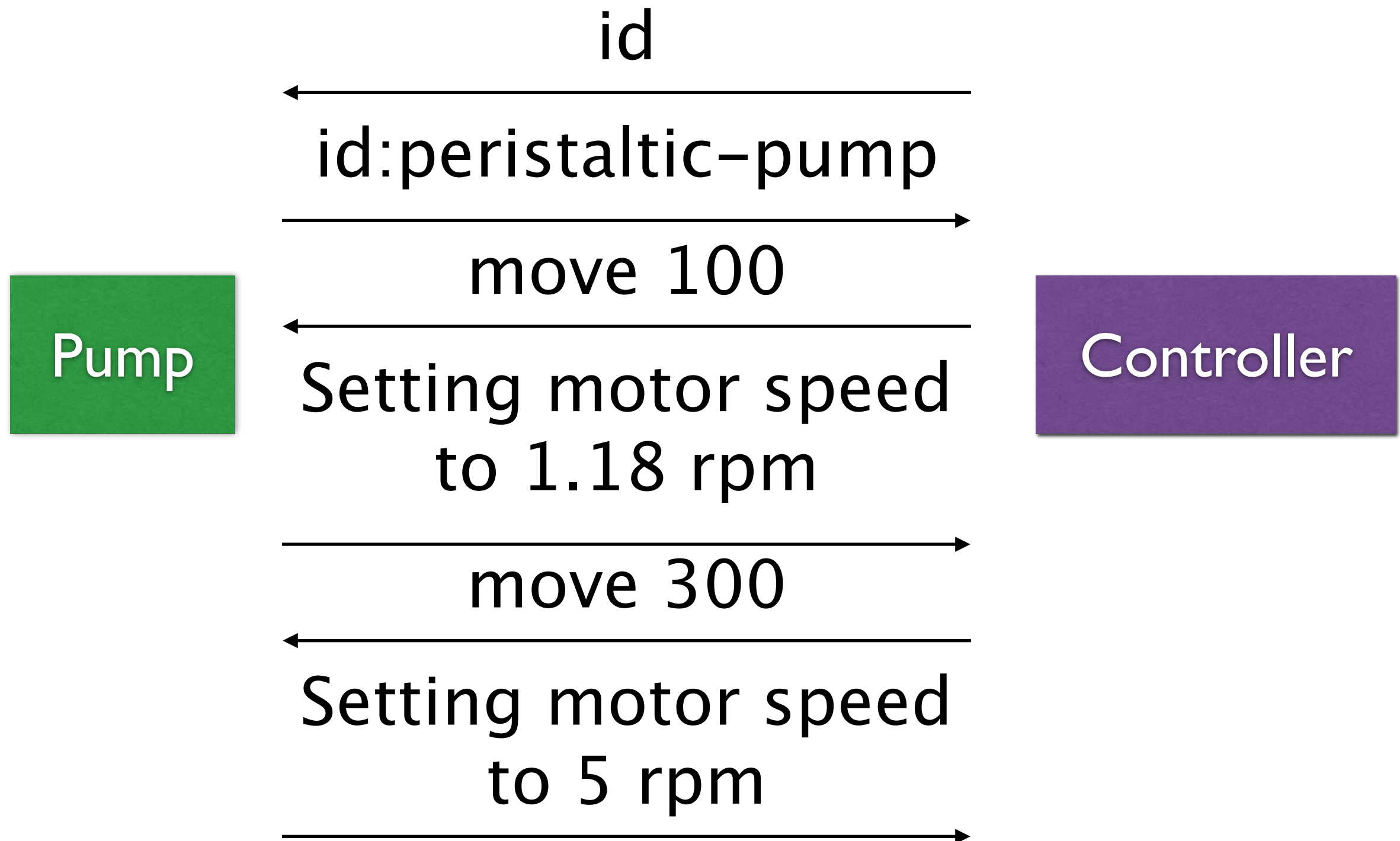
waag society

institute for art, science and technology

Demonstration



Communication scheme





Code of the pump

```
*/  
void loop() {  
  // Update clock  
  uint32_t time = millis(); // current time since start of sketch  
  uint16_t dt = time-lastTick; // difference between current and previous time  
  lastTick = time;  
  
  ledstate = 1-ledstate;  
  digitalWrite(LED_PIN, ledstate); // alternate between 0 and 1  
  
  if (time > lastUpdate + 100) {  
    if (encoderValue != 0) {  
      setMotorSpeed(motorSpeed + encoderValue);  
      encoderValue=0;  
    }  
    lastUpdate=time;  
    updateLCD();  
  }  
  
  while (Serial.available()>0) {  
    char c = (char)Serial.read();  
    if (c == '\n') {  
      if(buffer.startsWith("id")) {  
        Serial.println("id:peristaltic-pump"); // so the bioreactor can figure out what is connected  
      } else if (buffer.startsWith("move")) {  
        float sp = buffer.substring(4).toInt()/60.0f;  
        Serial.print(F("Setting motor speed to "));  
        Serial.print(sp, 2);  
        Serial.println(F(" rpm"));  
        setMotorSpeed(sp);  
      }  
      else {  
        Serial.println("Unknown cmd.");  
      }  
      buffer="";  
    } else buffer+=c;  
  }  
}
```



Identification

[....]

```
while (Serial.available() > 0) {  
  char c = (char)Serial.read();  
  if (c == '\n') {  
    if(buffer.startsWith("id")) {  
      Serial.println(„id:peristaltic-pump");  
    }  
  }  
}
```

[....]



Code on Computer

```
void serialEvent(Serial serial) {  
  
    SerialPortBuffer spb = null;  
    for (SerialPortBuffer s : serialPorts.values ())  
        if (s.port == serial) {  
            spb=s;  
            break;  
        }  
  
    if (spb != null) {  
        while (serial.available () >0) {  
            char c = serial.readChar();  
            if (c == '\n' && spb.buffer.length() > 0) {  
  
                if (serial == bioreactorSerial) {  
                    println("bioreactor: " + spb.buffer);  
  
                    if (spb.buffer.startsWith("temp")) {  
                        measuredTemperature = Float.parseFloat(spb.buffer.substring(5));  
                        addTemp( measuredTemperature );  
                    }  
                } else if (pumps.containsKey(serial)) {  
                    pumps.get(serial).processSerialLine(spb.buffer);  
                } else {  
                    if (spb.buffer.startsWith("id:")) {  
                        initDevice(spb.buffer.substring(3).trim(), serial);  
                    } else  
                        serial.write("\nid\n");  
                    spb.buffer="";  
                } else {  
                    spb.buffer+=c;  
                }  
            }  
        }  
    }  
}
```



Code on computer

```
void initDevice(String deviceId, Serial serial)
{
    println("id: " + deviceId);
    if (deviceId.equals("bioreactor")) {
        bioreactorSerial = serial;
        println("bioreactor connected.");
    }

    if (deviceId.equals("peristaltic-pump")) {
        addPump(deviceId, serial, 0);
        println("peristaltic pump connected.");
    }

    if (deviceId.equals("syringe-pump")) {
        addPump(deviceId, serial, 1);
        println("syringe pump connected.");
    }
}
```



Code on computer

```
void addPump(String pumpType, Serial serialPort, int pumpldx)
{
    GCustomSlider sdr;

    sdr = new GCustomSlider(this, 60, 80 + 120 * pumpldx, 260, 50, "blue18px");
    // show      opaque ticks value limits
    sdr.setShowDecor(false, true, true, true);
    // there are 3 types
    // GCustomSlider.DECIMAL e.g. 0.002
    // GCustomSlider.EXPONENT e.g. 2E-3
    // GCustomSlider.INTEGER
    sdr.setNumberFormat(G4P.INTEGER, 0);
    sdr.setLimits(0.5f, 0f, 1.0f);

    if (pumpType.equals("peristaltic-pump")) {
        sdr.setLimits(-1000, 1000);
    }

    if (pumpType.equals("syringe-pump")) {
        sdr.setLimits(-1000, 1000);
    }

    Pump p = new Pump(sdr, serialPort);

    p.label = new GLabel(this, 10, 65 + pumpldx * 120, 180, 20);
    p.label.setText("Pump " + pumpldx + " (" + pumpType + ")");
    p.label.setLocalColorScheme(GCScheme.GREEN_SCHEME);

    pumps.put(serialPort, p);
}
```



Communicating to the pump

```
void handleSliderEvents(GValueControl slider, GEvent event) {  
    for (Pump p : pumps.values ()) {  
        if (p.slider == slider) {  
            p.continuousRotation(slider.getValueF());  
        }  
    }  
}
```

[...]

```
class Pump {
```

[...]

```
    void continuousRotation(float rpm) {  
        port.write("move" + (int)rpm + "\n");  
        println("move" + (int)rpm);  
    }  
};
```



Pump obeying the command

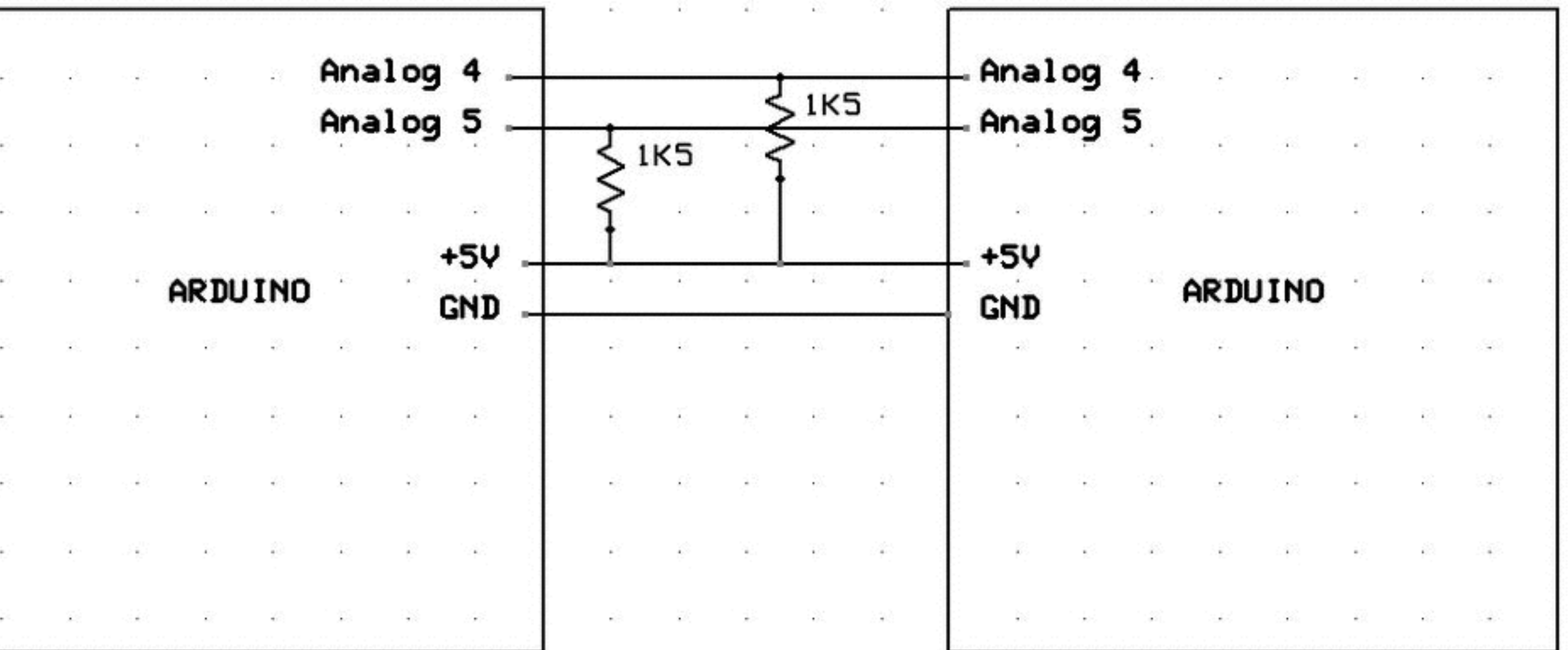
[....]

```
while (Serial.available() > 0) {  
  char c = (char)Serial.read();  
  if (c == '\n') {  
    if(buffer.startsWith("id")) {  
      Serial.println("id:peristaltic-pump");  
    } else if (buffer.startsWith("move")) {  
      float sp = buffer.substring(4).toInt()/60.0f;  
      Serial.print(F("Setting motor speed to "));  
      Serial.print(sp, 2);  
      Serial.println(F(" rpm"));  
      setMotorSpeed(sp);  
    }  
  }  
}
```

[....]



Alternative: arduino 2 arduino





Master Code

```
#include <Wire.h>

#define LED_PIN 13
byte x = 0;

void setup()
{
  Wire.begin(); // Start I2C Bus as Master
  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, LOW);
}

void loop()
{
  Wire.beginTransmission(9); // transmit to device #9
  Wire.send(x);               // sends x
  Wire.endTransmission();     // stop transmitting
  x++;
  if (x > 5) x=0;
  delay(450);
}
```



Slave Code

```
#include <Wire.h>

#define LED_PIN 13
#define LED_1 12
#define LED_2 11

int x;

void setup() {
  Wire.begin(9);           // Start I2C Bus as a Slave (Device Number 9)
  Wire.onReceive(receiveEvent); // register event

  pinMode(LED_PIN, OUTPUT);
  pinMode(LED_1, OUTPUT);
  pinMode(LED_2, OUTPUT);

  digitalWrite(LED_PIN, LOW);
  digitalWrite(LED_1, LOW);
  digitalWrite(LED_2, LOW);

  x = 0;
}

void loop() {
  //If value received is 0 blink LED 1
  if (x == 0) {
    digitalWrite(LED_1, HIGH);
    delay(200);
    digitalWrite(LED_1, LOW);
    delay(200);
  }
  //If value received is 1 blink LED 2
  if (x == 1) {
    digitalWrite(LED_2, HIGH);
    delay(200);
    digitalWrite(LED_2, LOW);
    delay(200);
  }
}
```



waag society

institute for art, science and technology

Graduation Show

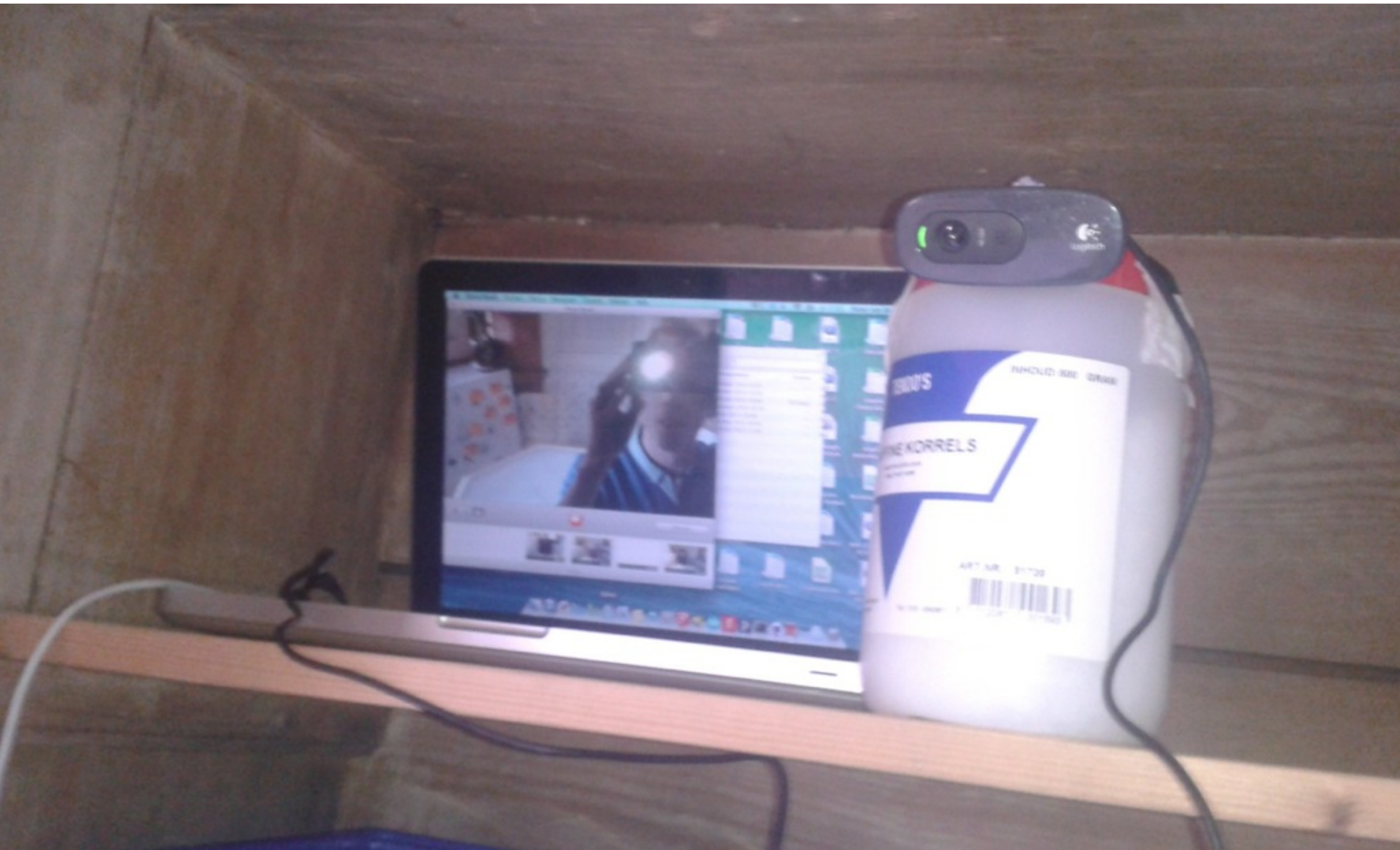


Program

- 18:30 Doors of TA open
- 19:00 Opening
- 19:10 5 Presentations of participants
- 20:10 Presentation of the certificates
- 20:30 Graduation ceremony
- 20:45
Show-and-tell fair of all projects
in the MakersGuild and drinks



Truth Booth Videos





Continue your stay

- 2 more months in the Wetlab
- 2 days a week
- Supported by an assistant
- Reserve machine time





**some
rights
reserved**

These slides are published by Waag
Society under CC-BY-SA 4.0 license