



ablauf

Mo → Intro und Basics Hardware

Di → Konzept & Experimente

Mi → Finalisieren und Abschluss

about me

max hans

11 studium arch + id

16 freelance designer & technologist

18 diplom ,hack the planet‘

18 intuity media lab

25 freelance designer & technologist

14

surface



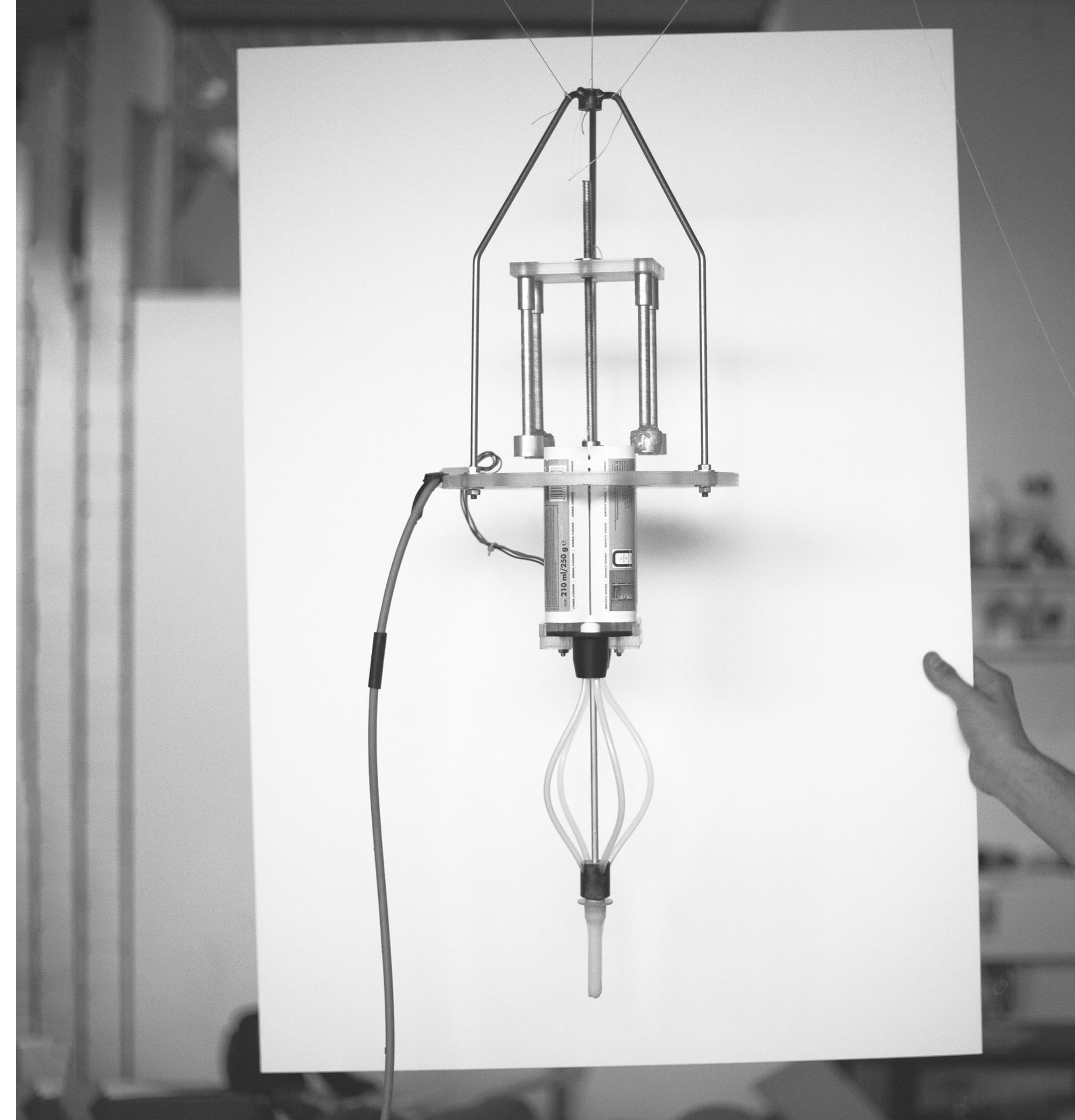
14

surface



15

foam printer



15

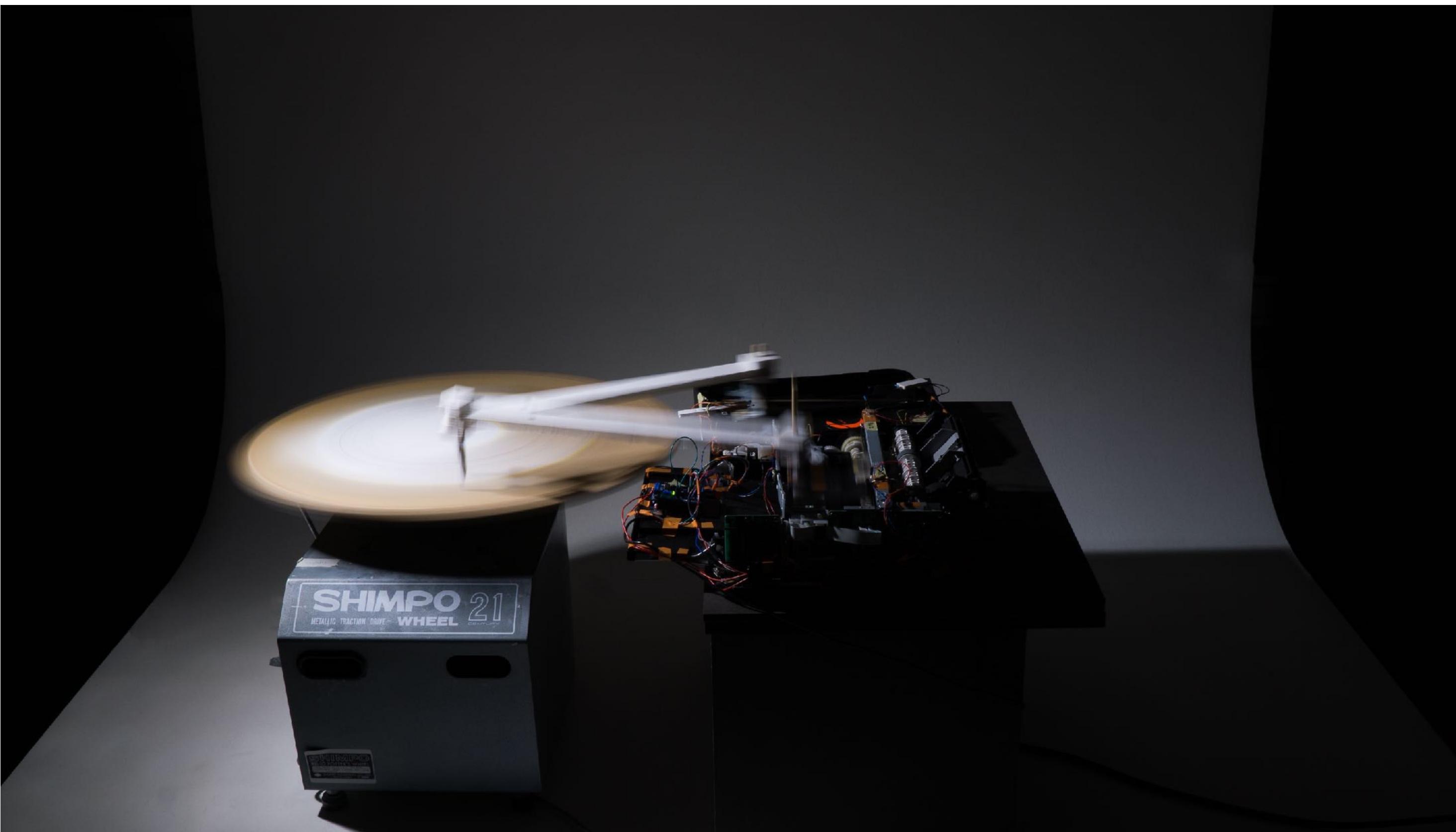
foam printer



17

workshop

creative hacking



AKA
INTERSTELLAR

workshop '17
creative hacking

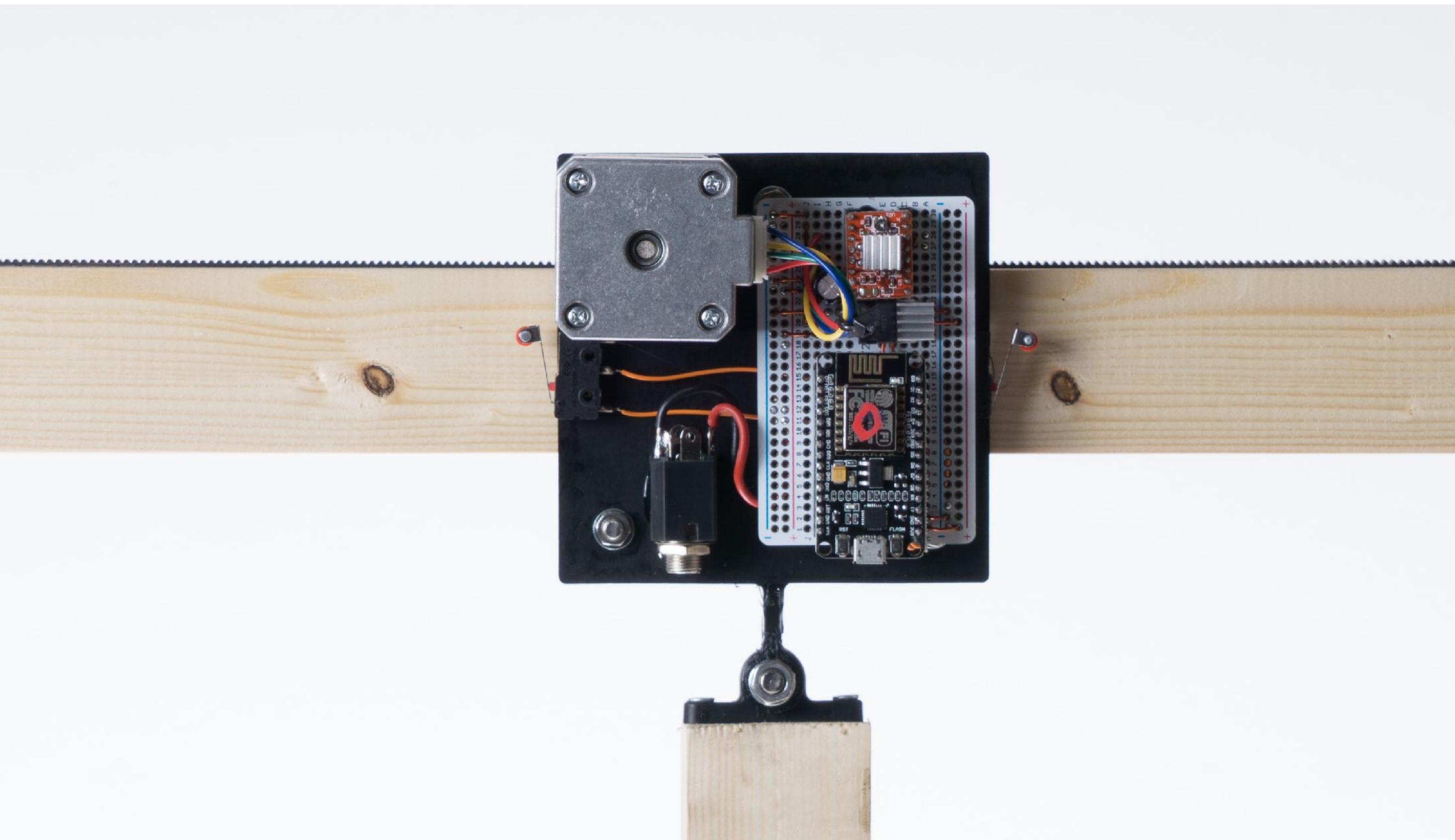
18
diplom
hack the planet



18

diplom

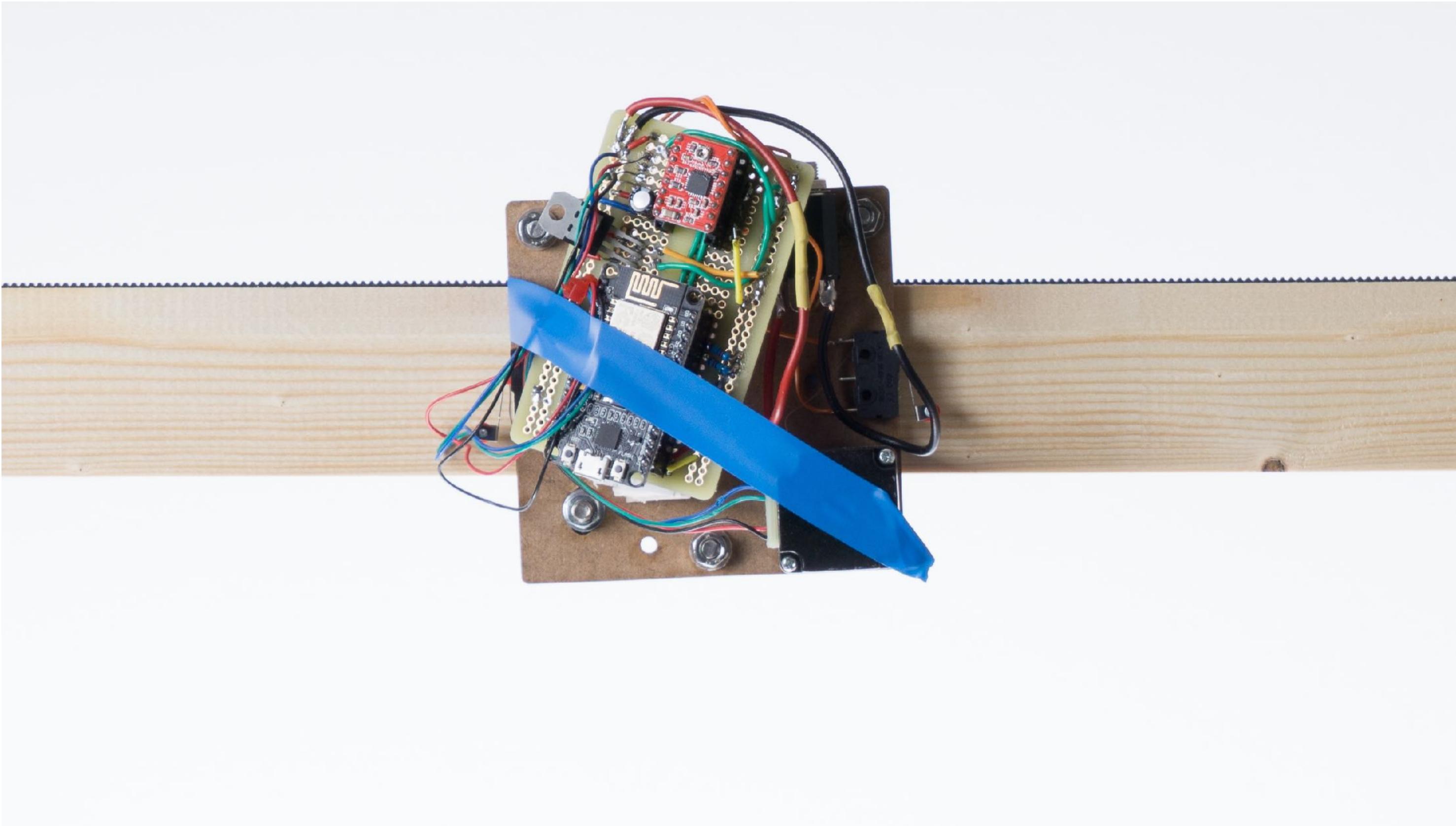
hack the planet



18

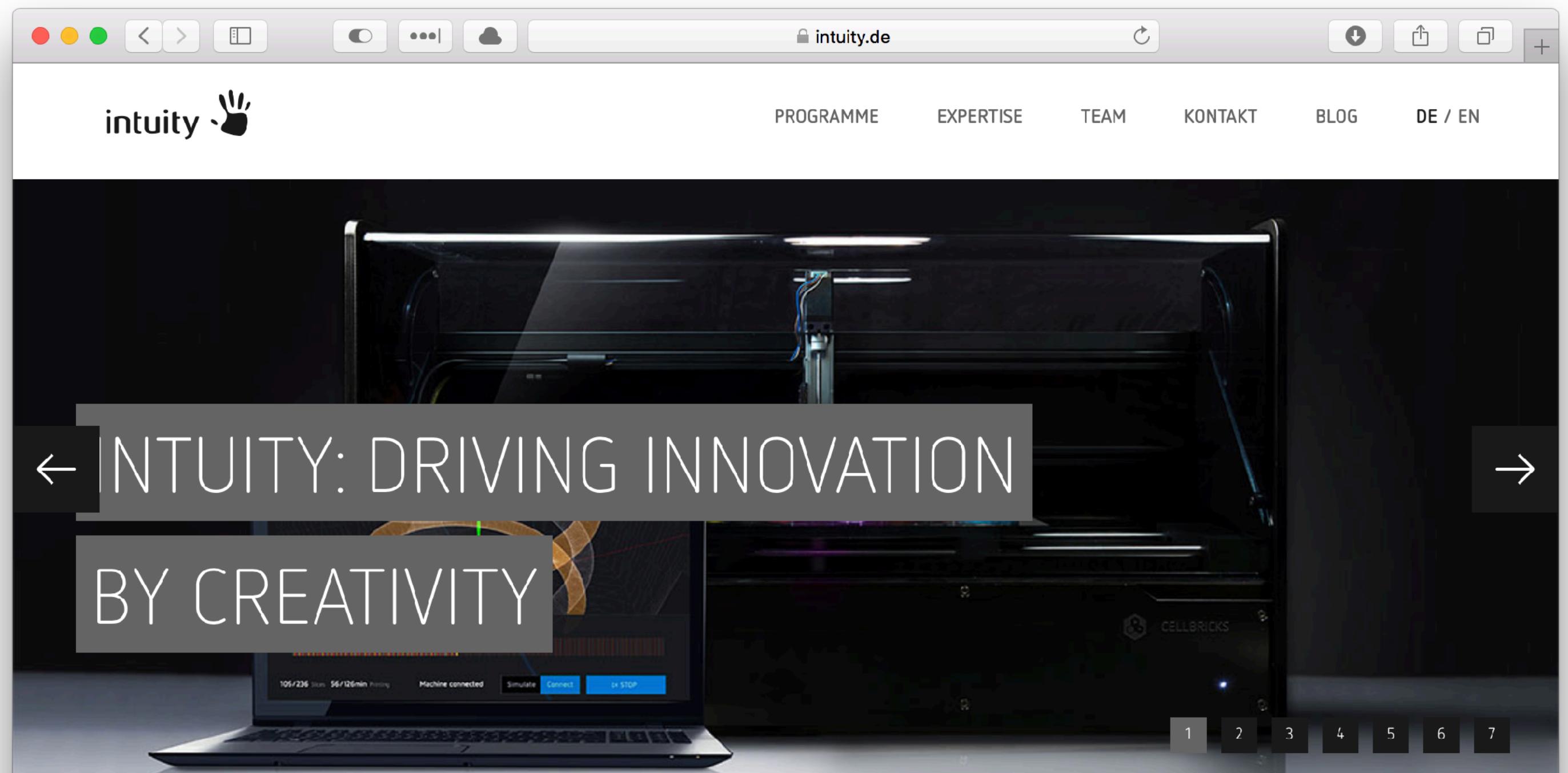
diplom

hack the planet



18

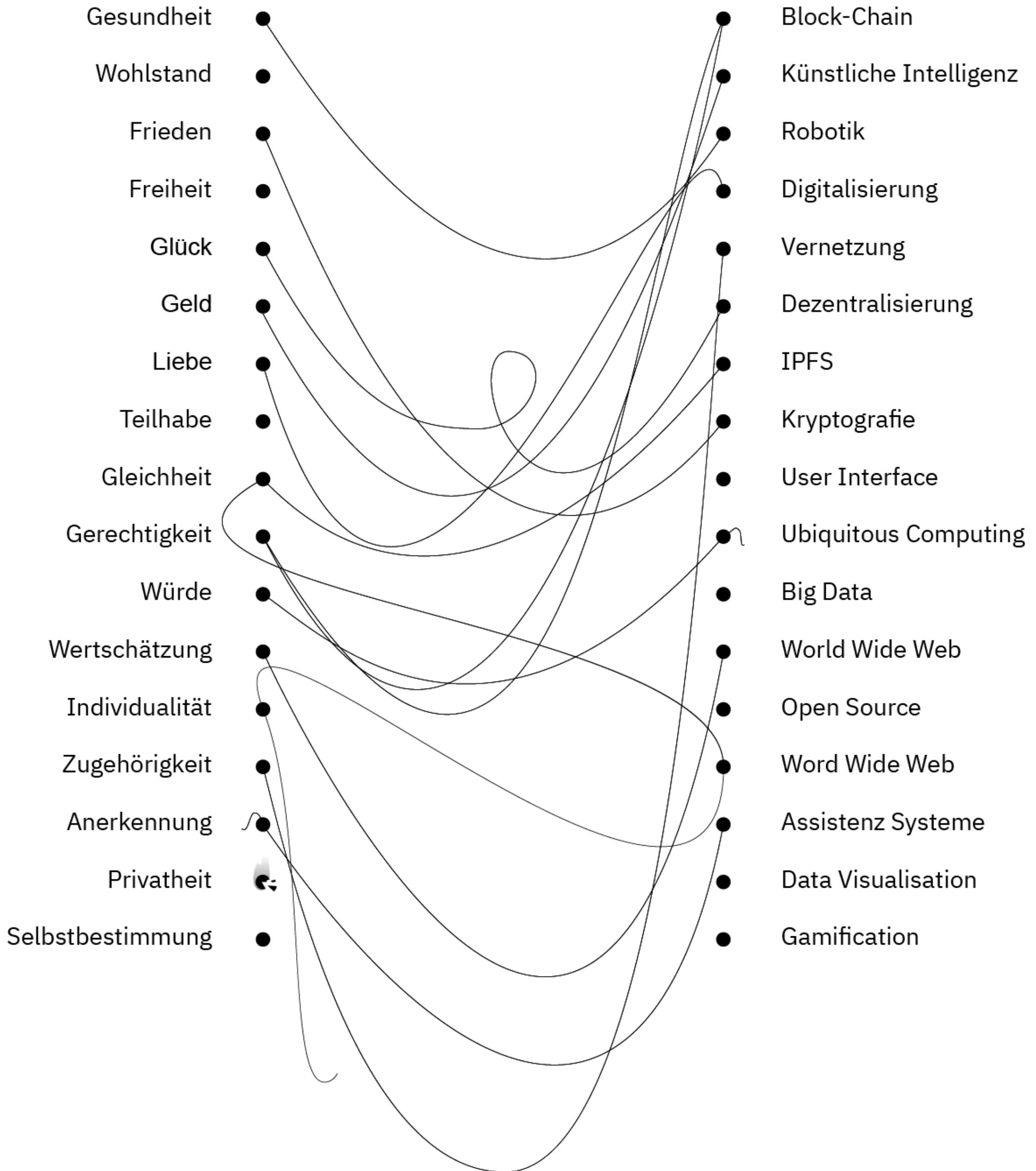
intuity media lab
stuttgart



Intuity entwickelt digitale Services und Produkte der nächsten Generation.
Wir kombinieren Kreativität, strategisch-systemisches Denken, Software-Hardware-
Prototyping und User-Experience-Design.

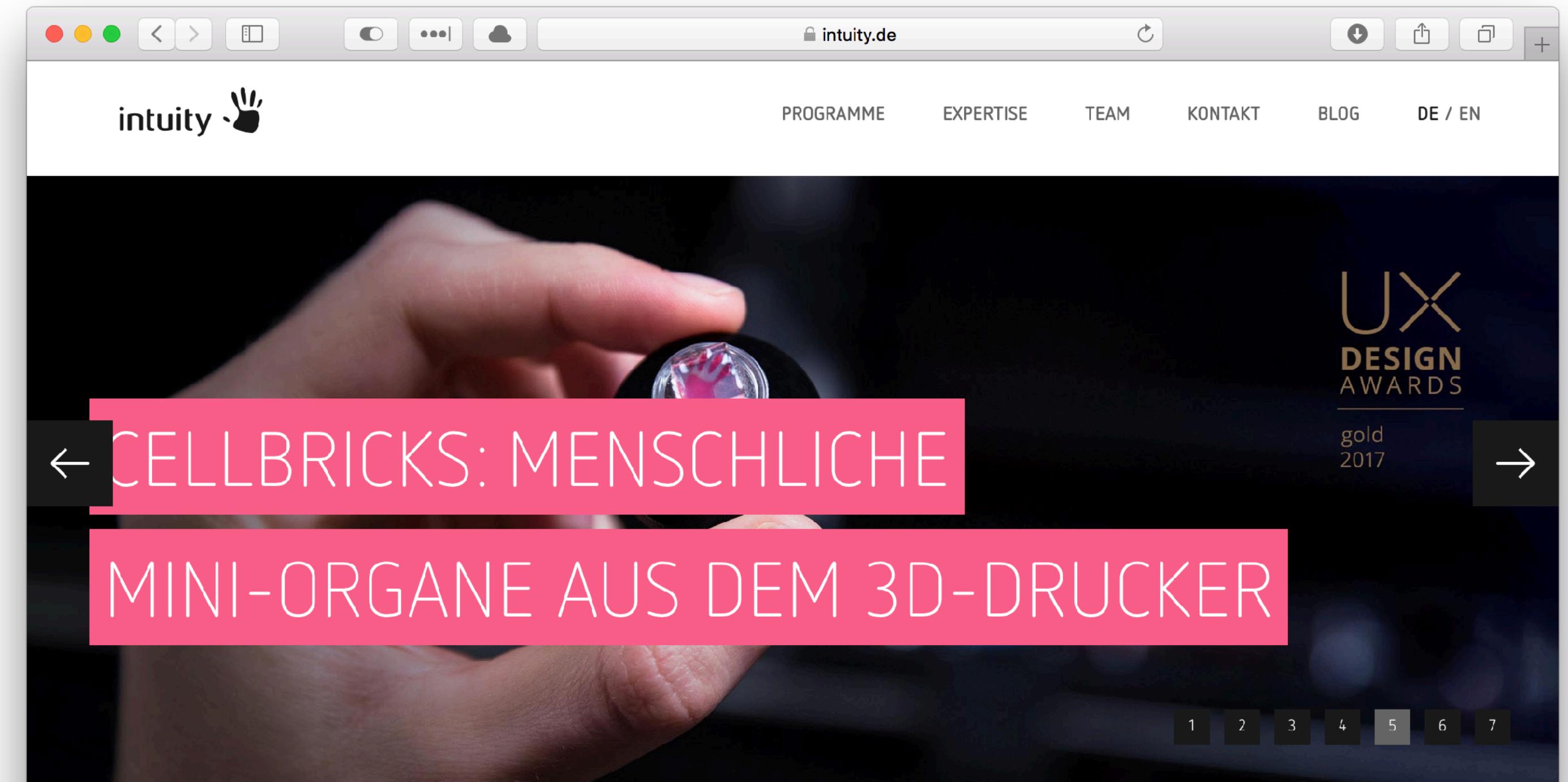
Wir beraten und begleiten Unternehmen, Chancen der digitalen Vernetzung zu erkennen und zu nutzen.
Unsere Erfahrung reicht von der Etablierung von Design-Thinking-Prozessen über IoT-Strategien für globale Konzerne bis zur Entwicklung von Inkubatoren zur systematischen Entwicklung neuer Geschäftsideen.

Komplexe Herausforderungen in neue Geschäftschancen und attraktive Produkte zu verwandeln ist eine hohe



cellbricks

3d printing organoids



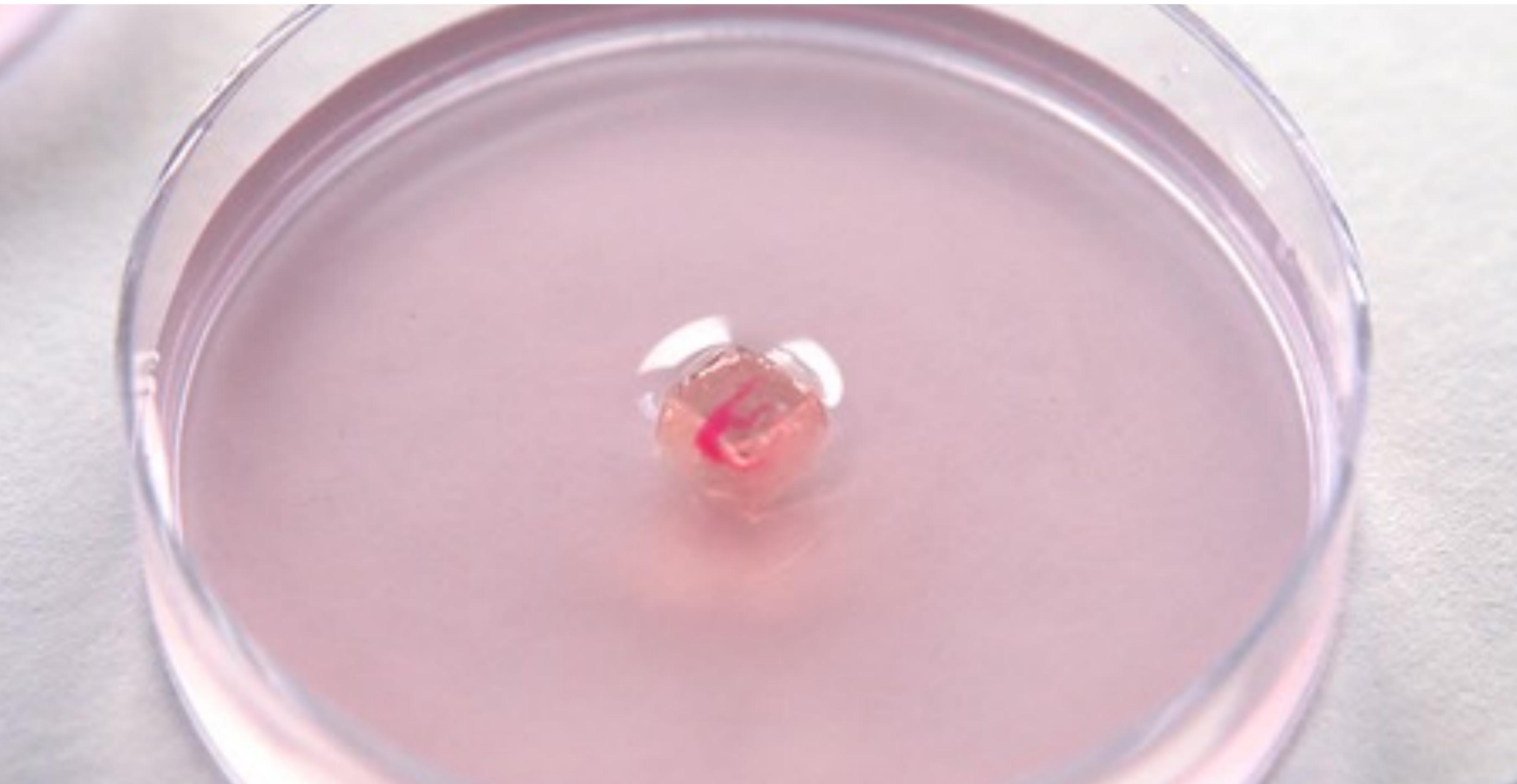
Intuity entwickelt digitale Services und Produkte der nächsten Generation.
Wir kombinieren Kreativität, strategisch-systemisches Denken, Software-Hardware-
Prototyping und User-Experience-Design.

Wir beraten und begleiten Unternehmen, Chancen der digitalen Vernetzung zu erkennen und zu nutzen.
Unsere Erfahrung reicht von der Etablierung von Design-Thinking-Prozessen über IoT-Strategien für globale
Konzerne bis zur Entwicklung von Inkubatoren zur systematischen Entwicklung neuer Geschäftsideen.

Komplexe Herausforderungen in neue Geschäftschancen und attraktive Produkte zu verwandeln ist eine hohe

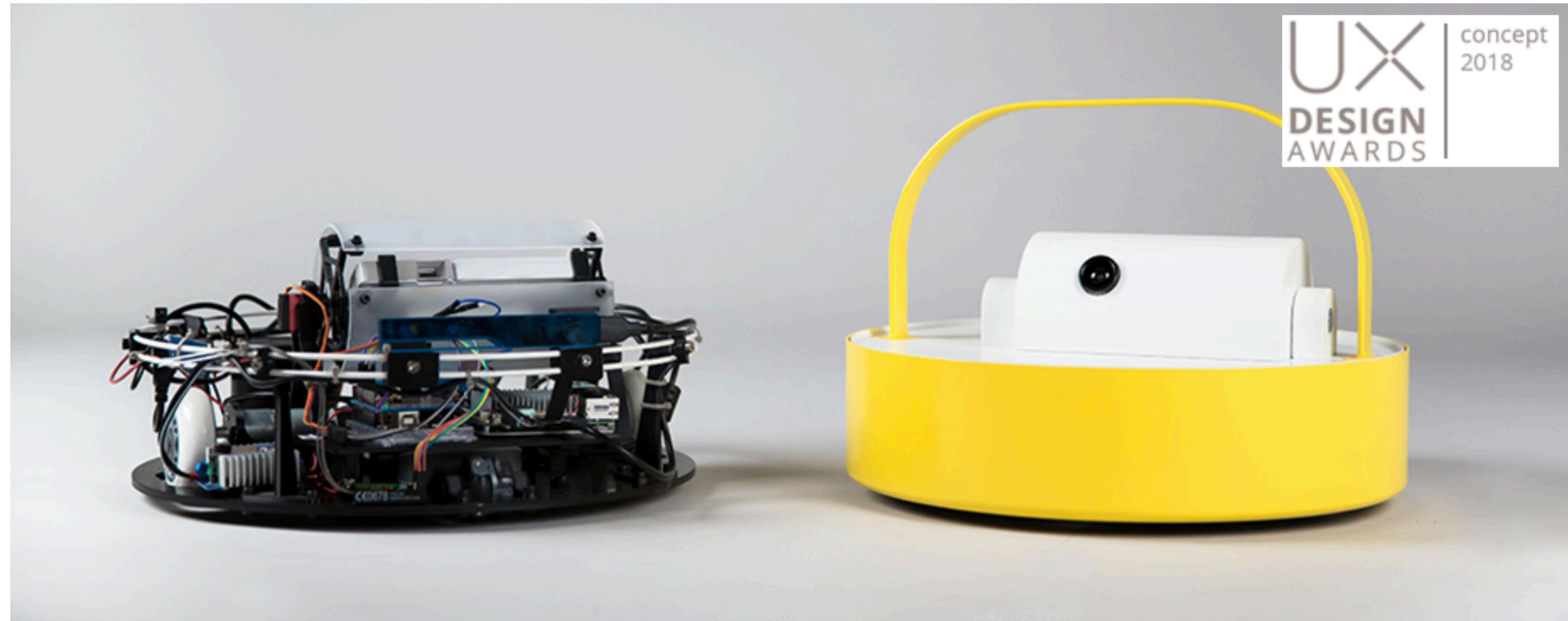
cellbricks

organ generator



kopernikus

an experimental prototyping platform
to test and evaluate
the vision of home robotics.



minimic

Super cheap, intelligent microscope

Automated sample scanning

Sample evaluation tool



various projects

focus on exploring the potential
of emerging technologies
in various fields of robotics,
software and AI



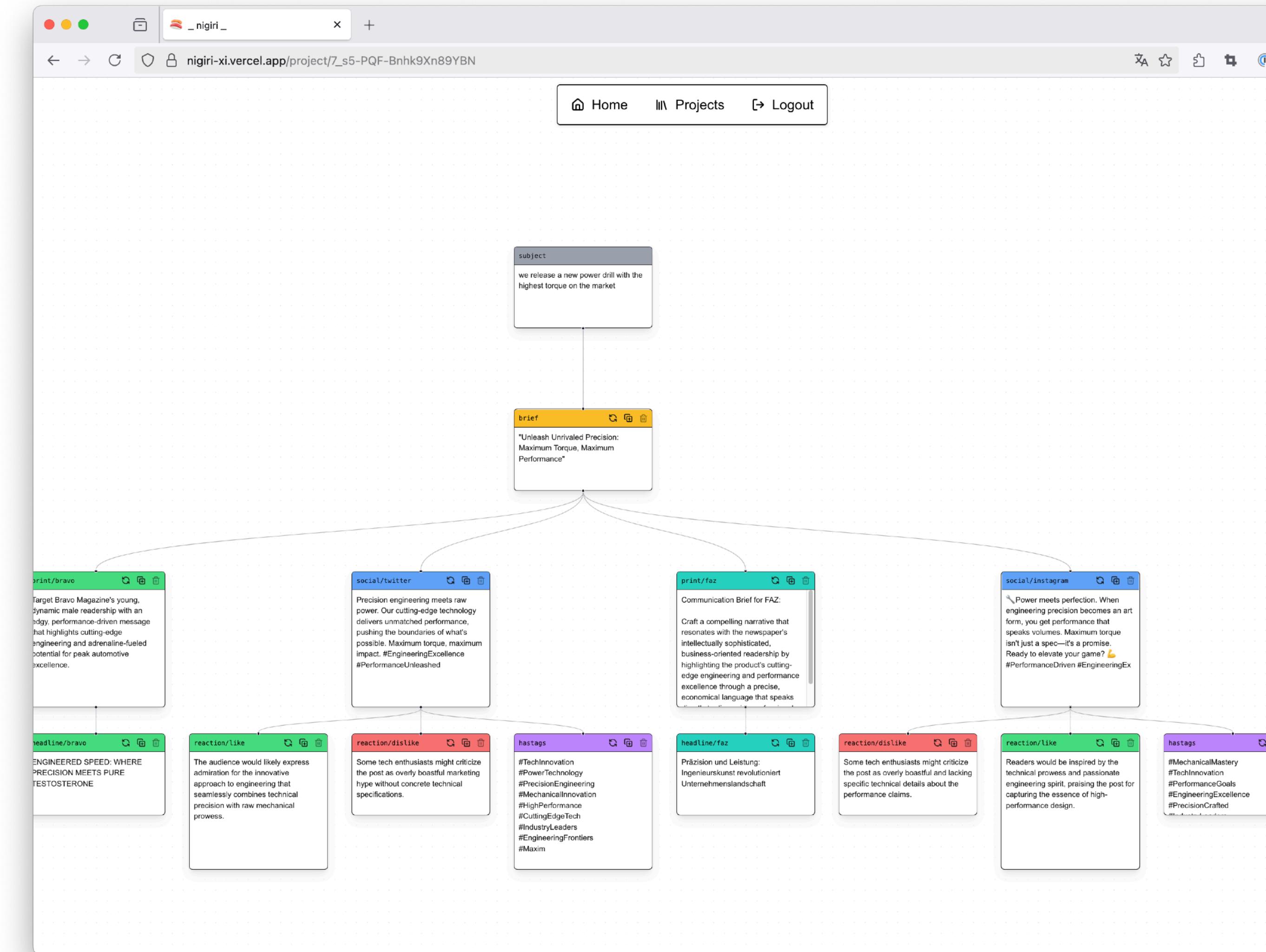
FESTOOL

ARRI

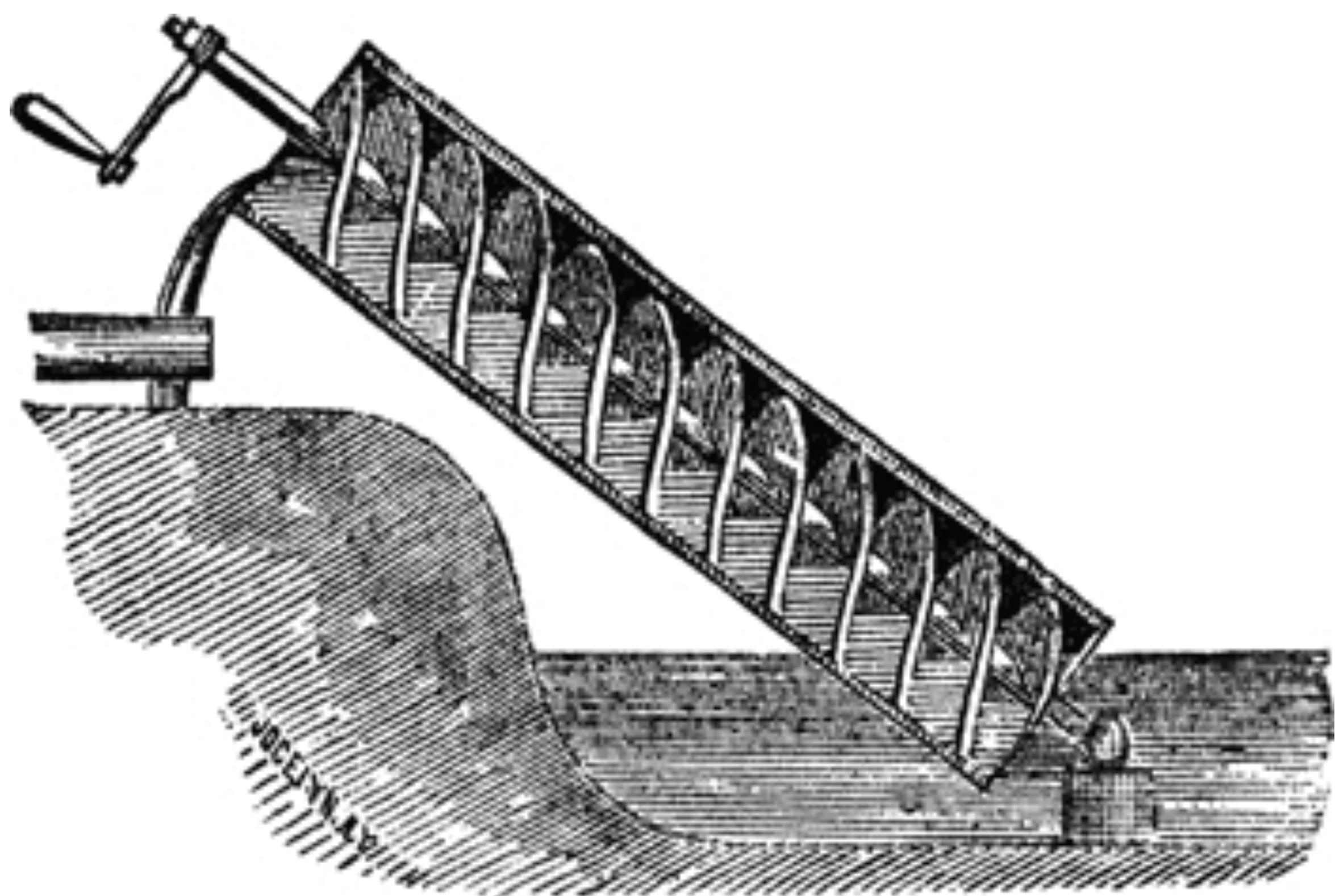


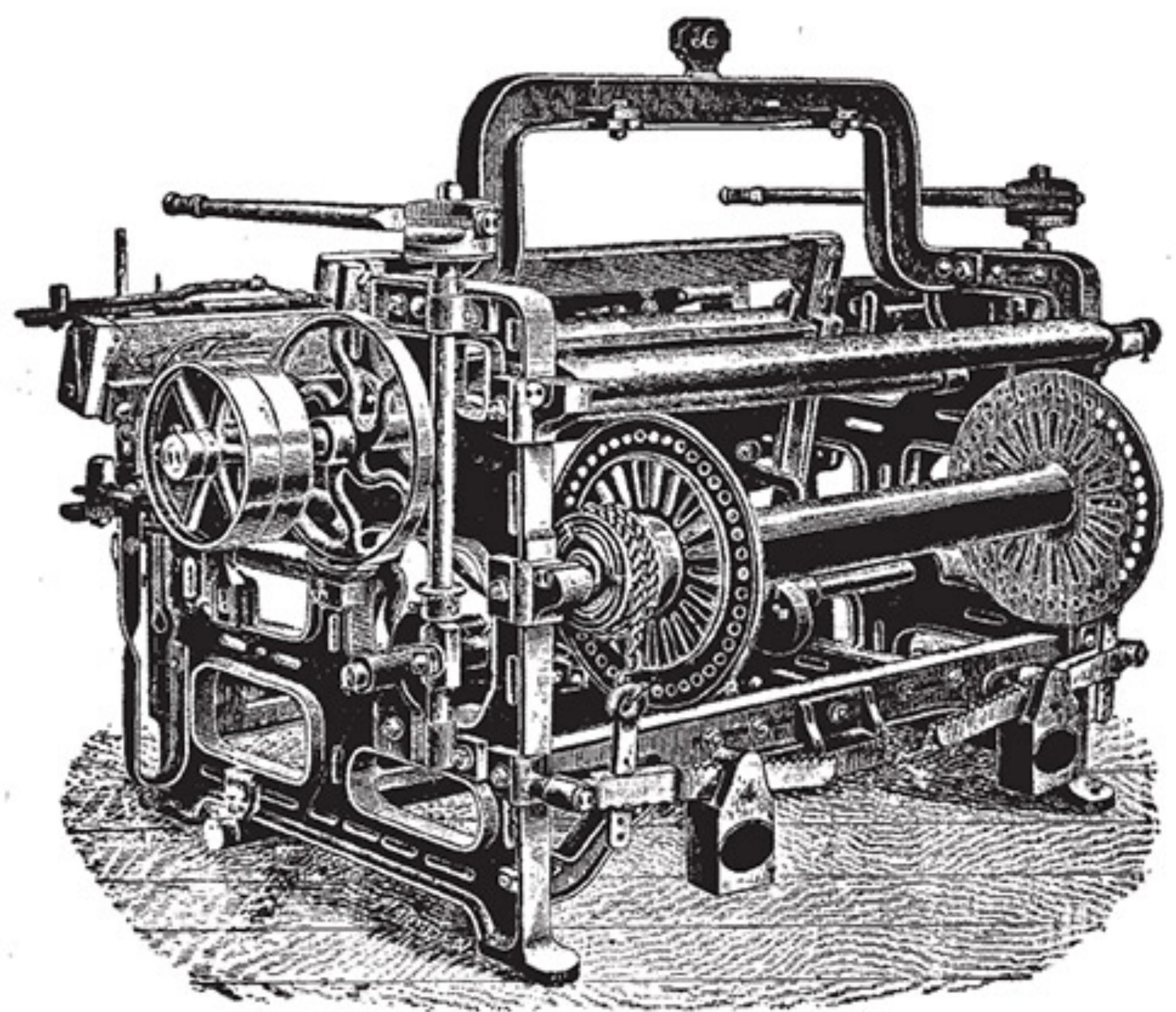
BOSCH

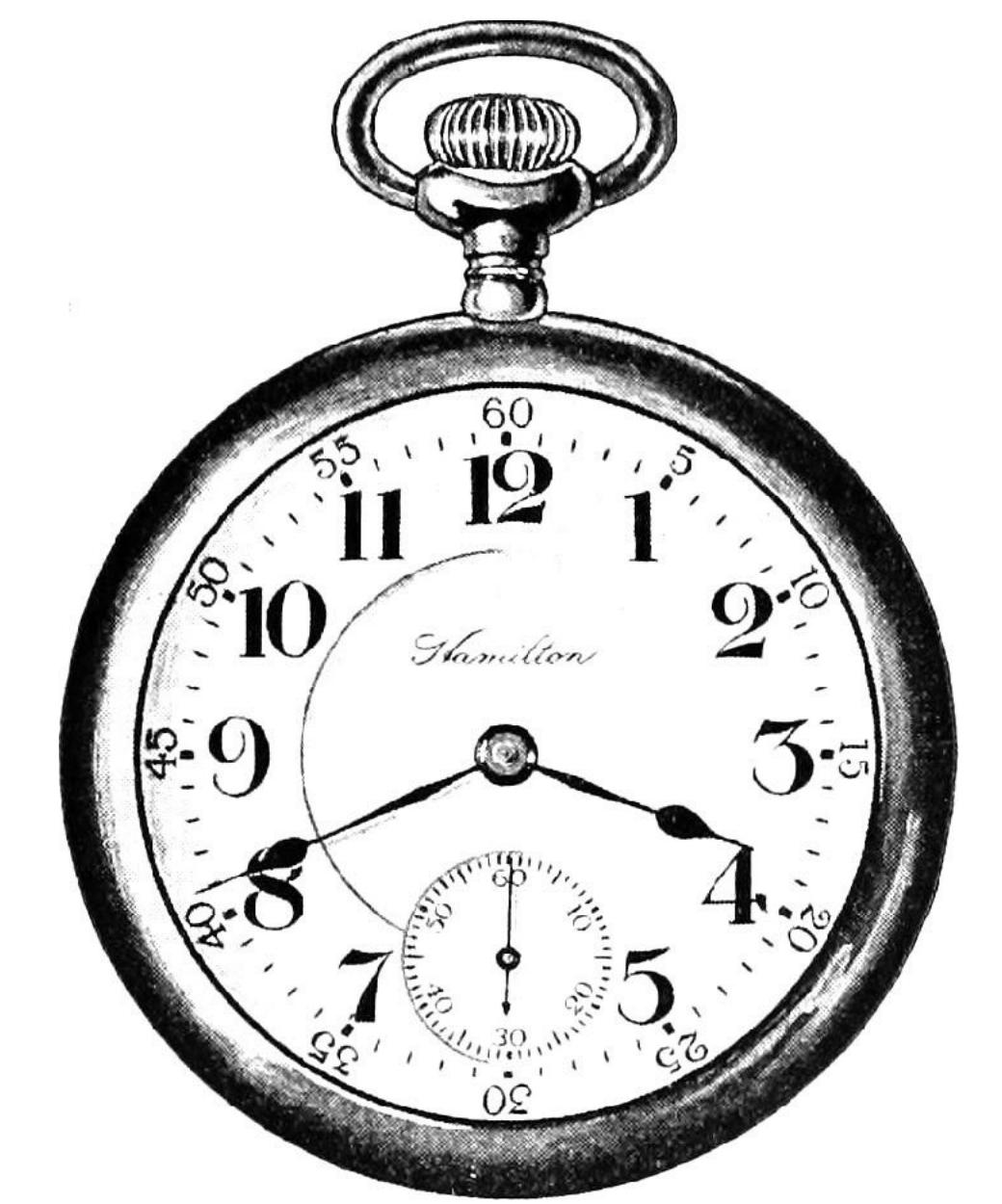
FESTO



computing

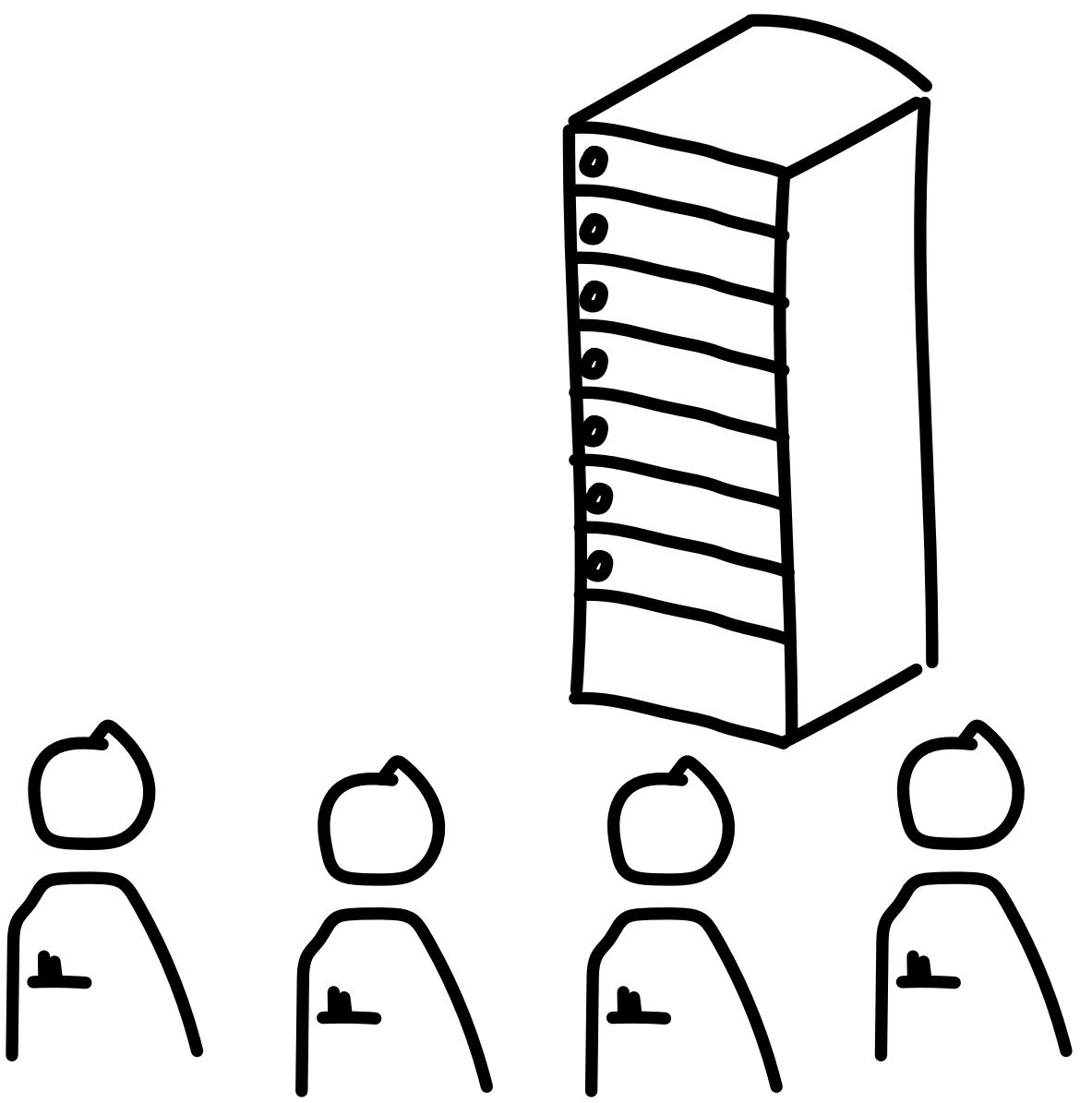


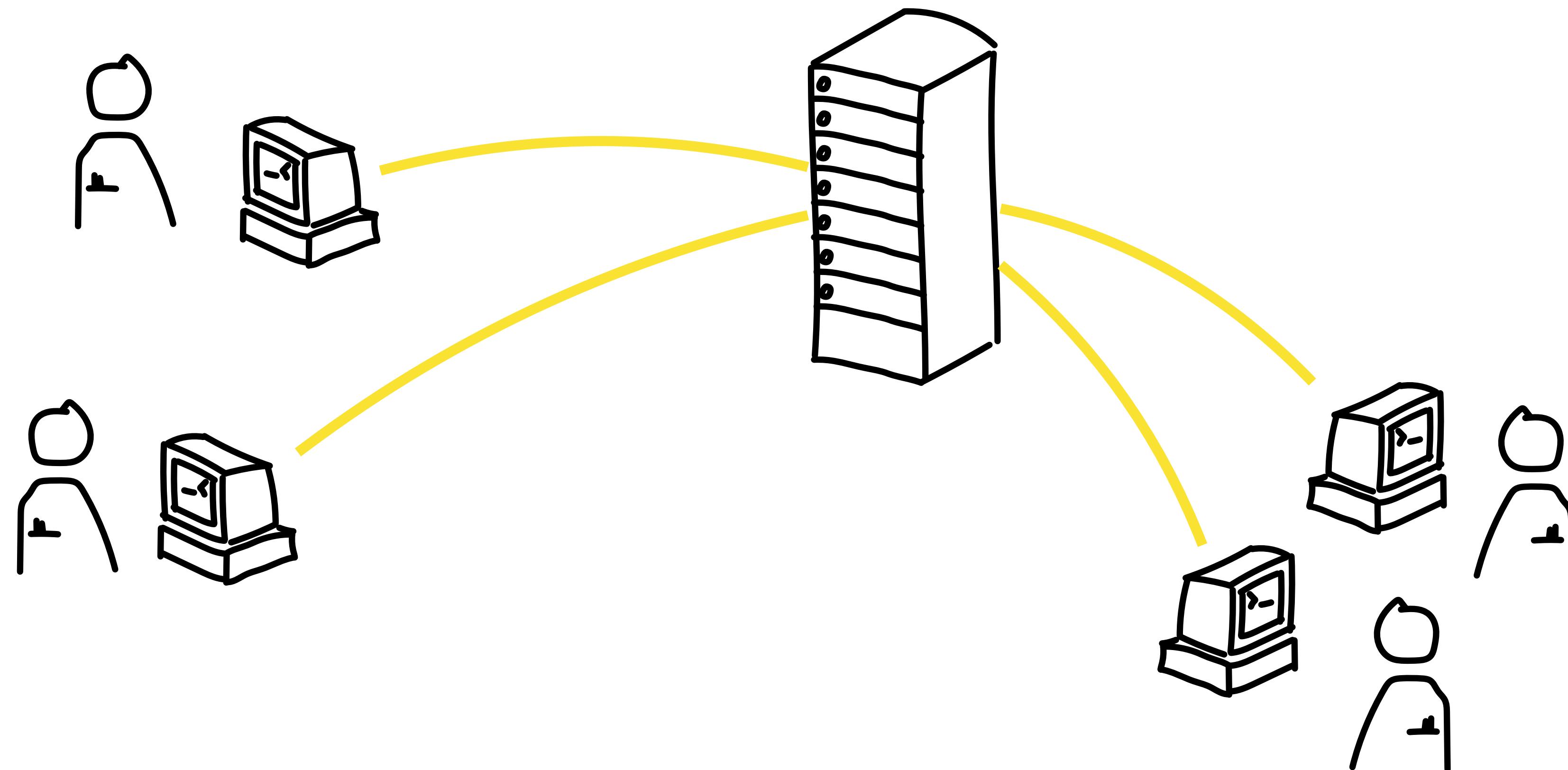












Introducing Macintosh. For the rest of us.

In the older days, before 1984,
not very many people used computers.
For a very good reason.



Not very many people knew how.
And not very many people wanted
to learn.

After all, in those days, it meant
listening to your stomach growl through
computer seminars. Falling asleep over
computer manuals. And staying awake
nights to memorize commands so

complicated you'd have to be a computer
to understand them.

Then, on a particularly bright day
in Cupertino, California, some
particularly bright engineers
had a particularly bright idea:
since computers are so smart,
wouldn't it make more sense
to teach computers about
people, instead of teaching people about
computers?

So it was that those very engineers
worked long days and late nights and
a few legal holidays, teaching tiny
silicon chips all about people. How they
make mistakes and change their minds.
How they refer to file folders and save
old phone numbers. How they labor for
their livelihoods, and doodle in their
spare time.

For the first time in recorded
computer history, hardware engineers

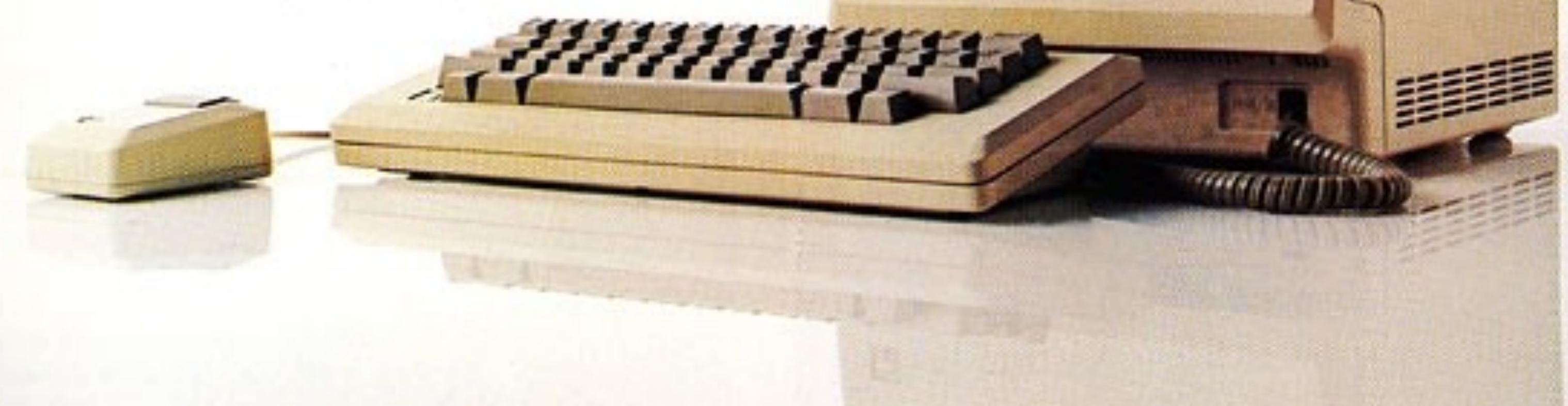
actually talked to software engineers
in moderate tones of voice, and both
were united by a common goal: to build
the most powerful, most portable, most
flexible, most versatile computer not very
much money could buy.

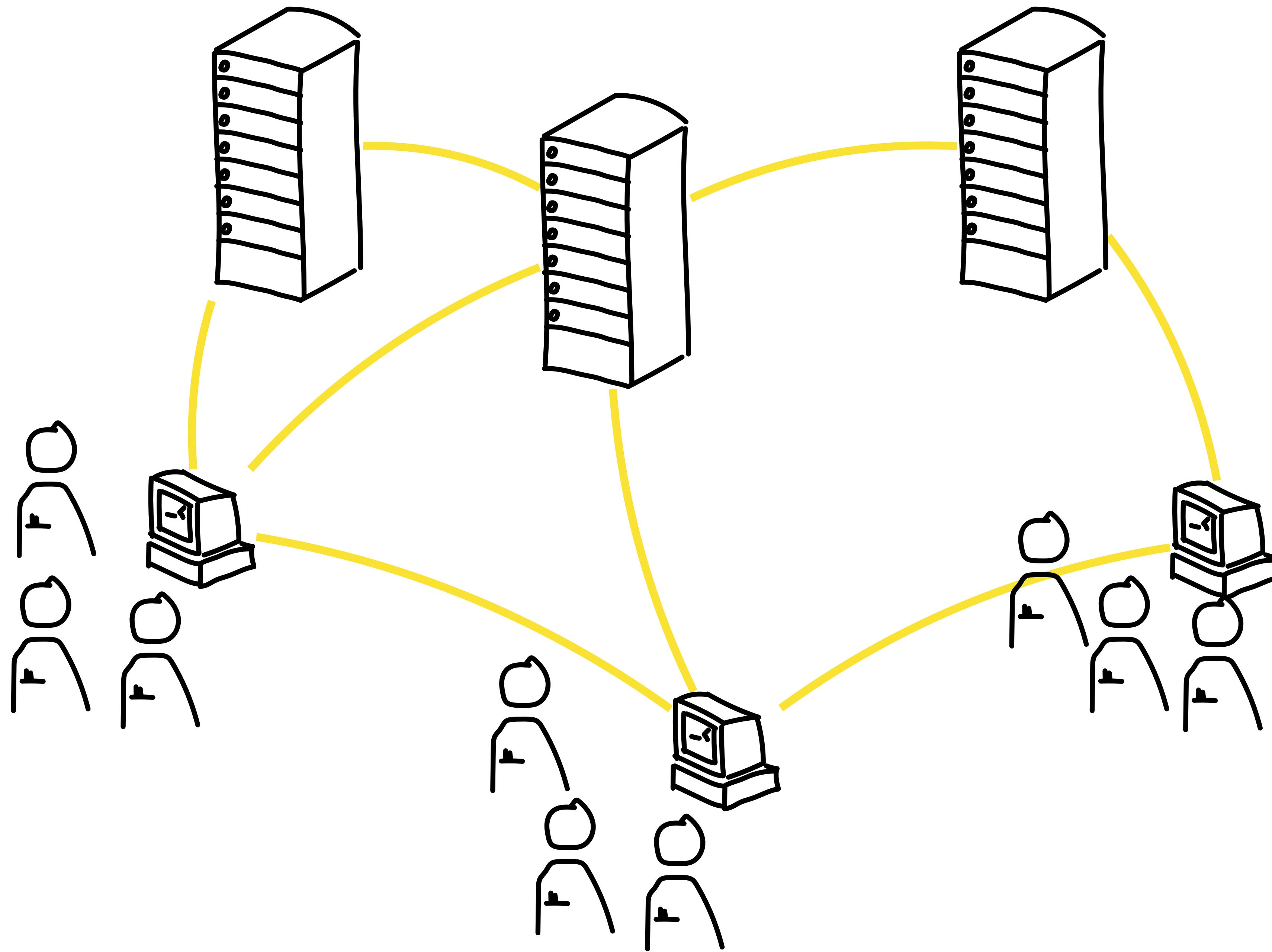
And when the engineers were
finally finished, they introduced us to
a personal computer so personable,
it can practically shake hands.

And so easy to use, most people
already know how.

They didn't call it the Q290, or
the Z8chip 9000.

They called it Macintosh.
And now we'd like to introduce
it to you.



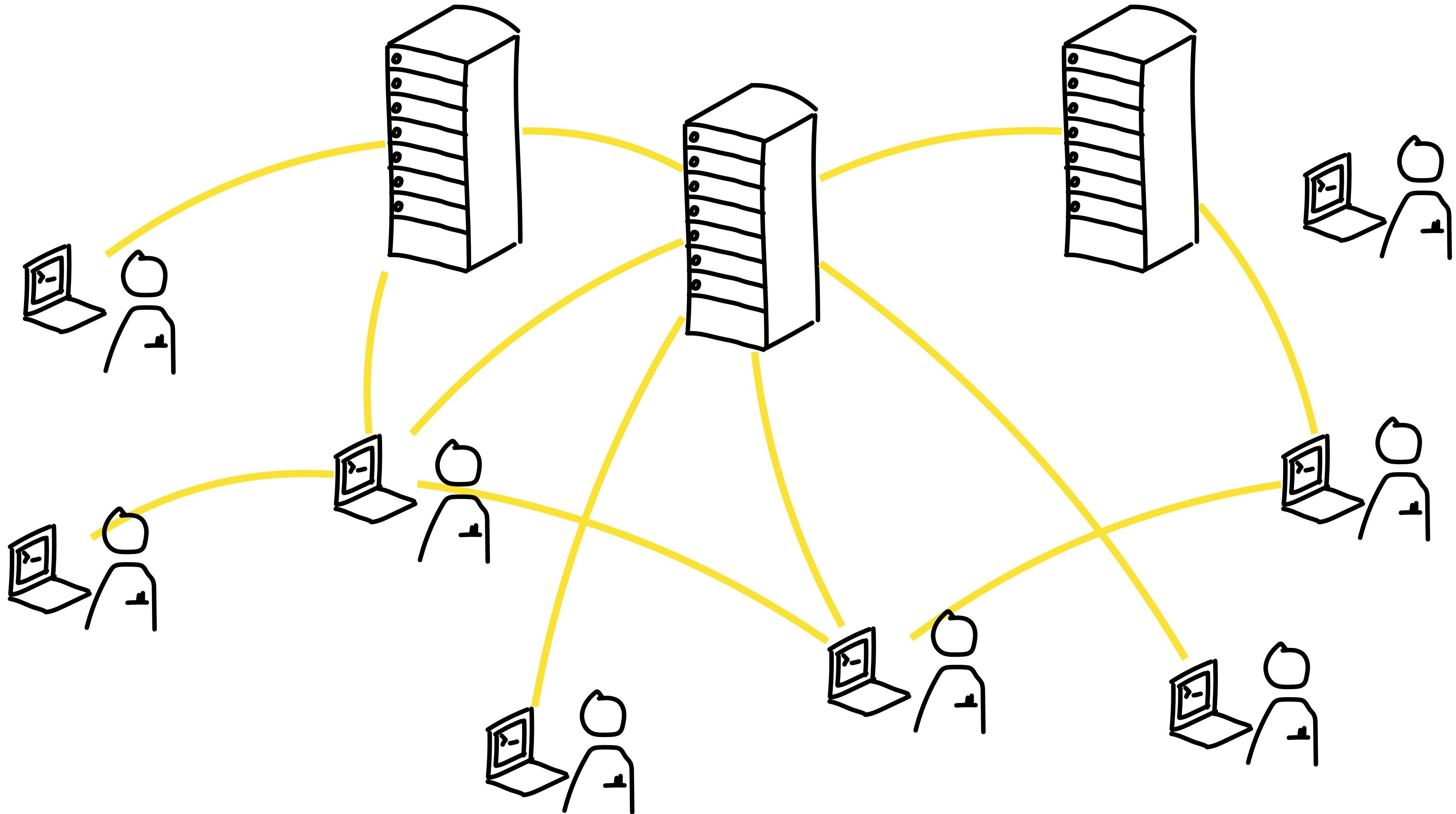


© 1998 Apple Computer, Inc. All rights reserved. The Mac logo, iMac, iPhoto, iMovie, iDVD and iLife and their designs are trademarks of Apple Computer, Inc.



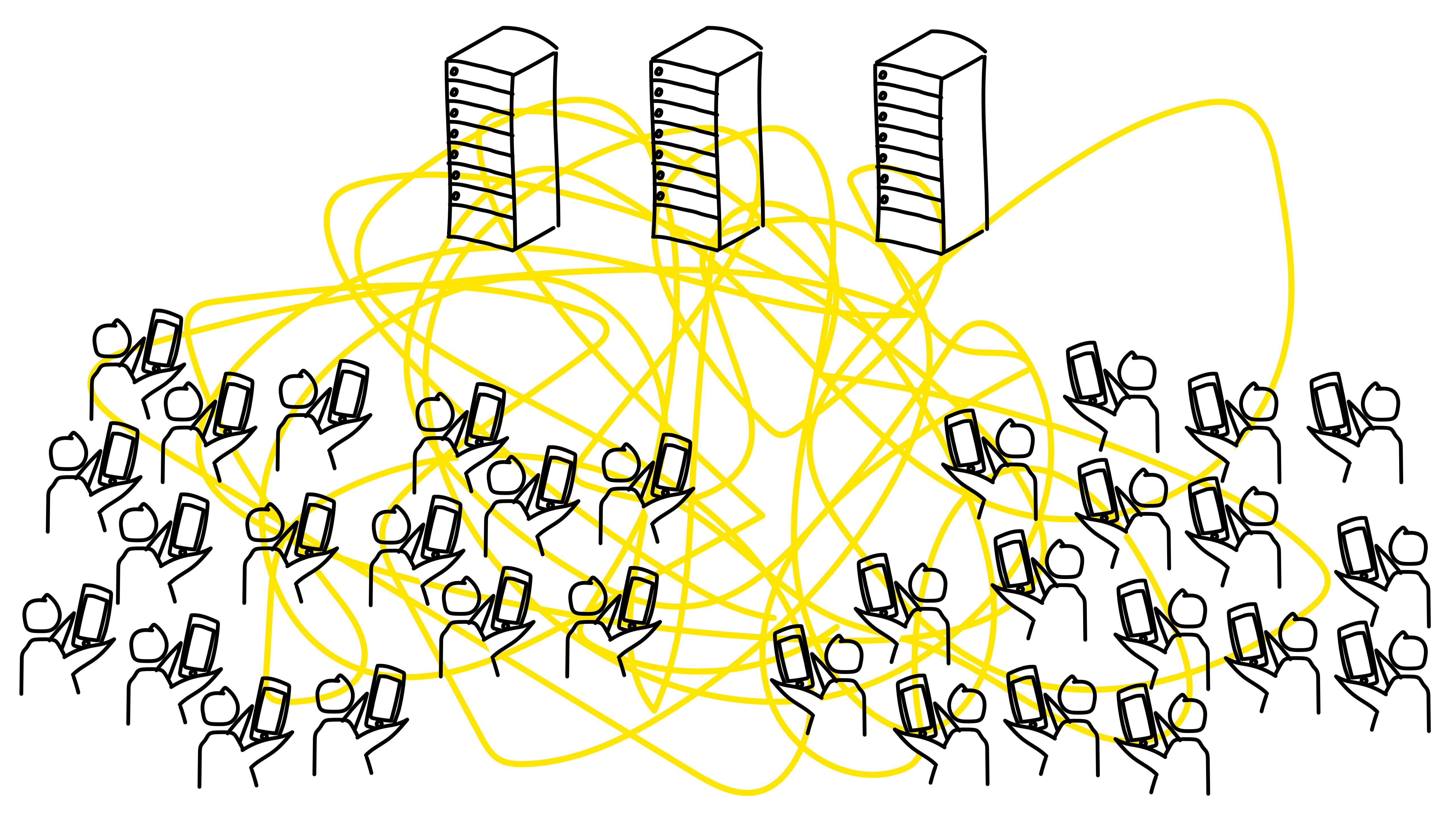
Chic. Not geek.

The new iMac.[™] A designer original. Fully accessorized with networking, modem, plenty of cool software and one-click internet access. www.apple.com.  This







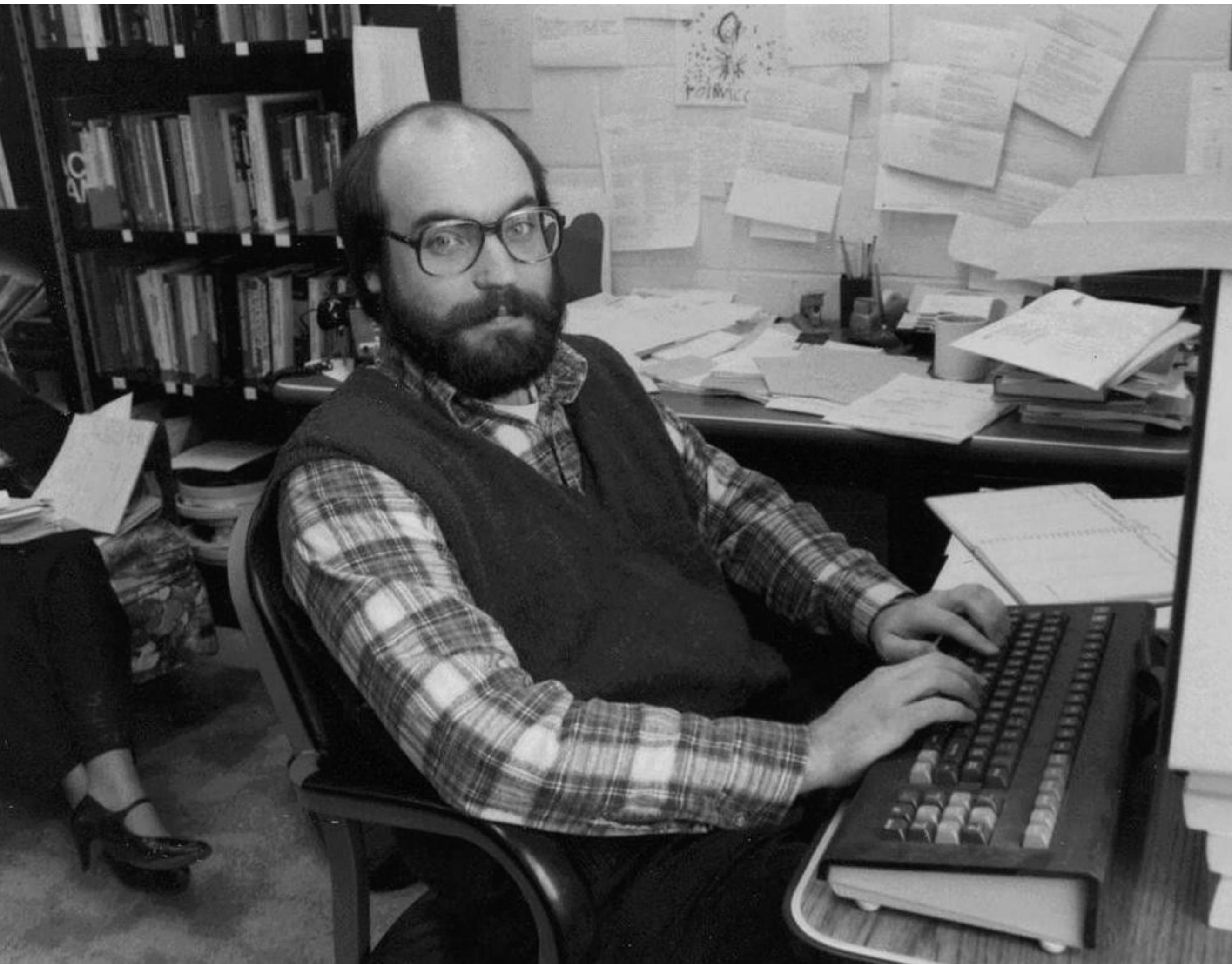




mark weiser

1955-1999

XEROX

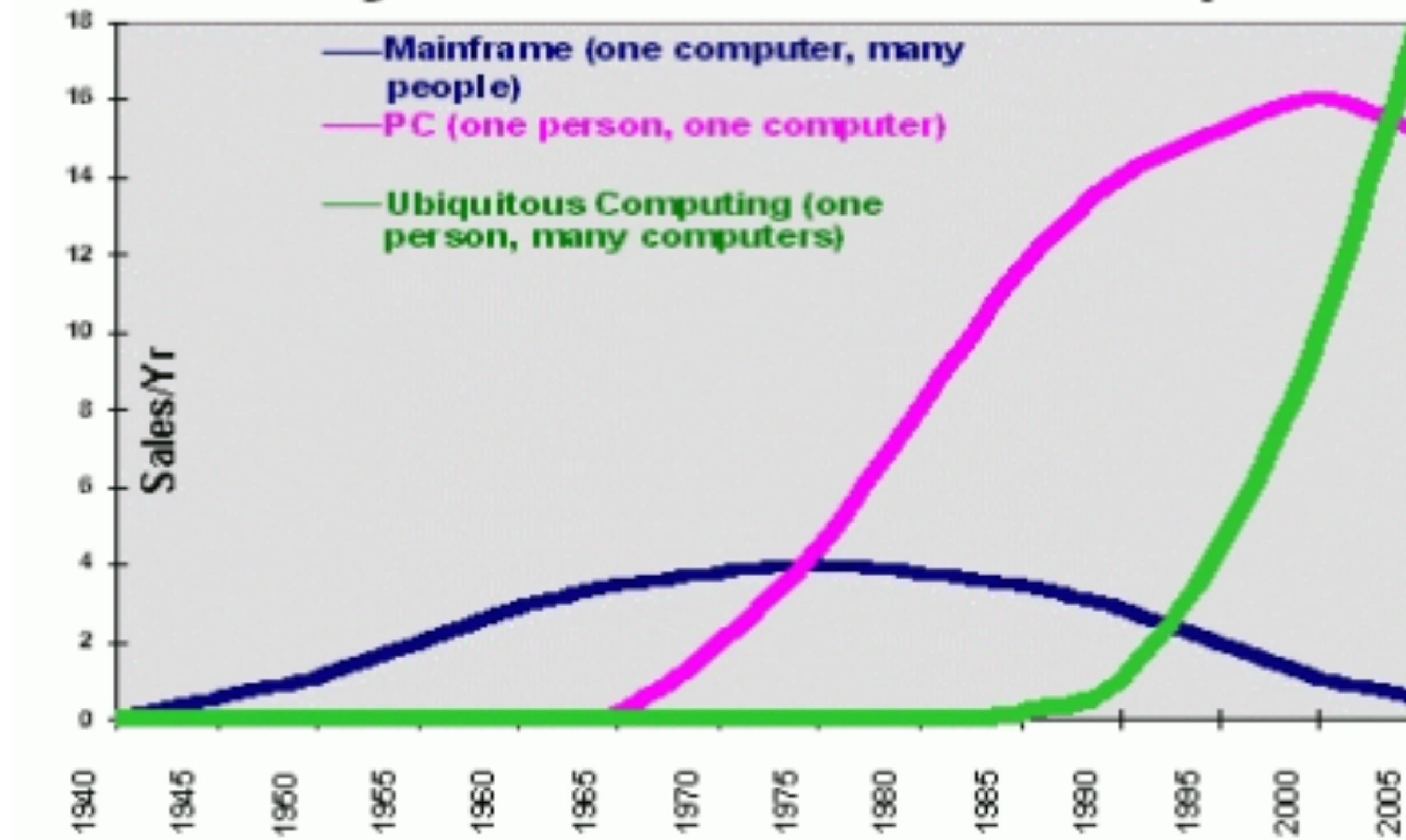


“In the 21st century the technology revolution will move into the everyday, the small and the invisible.”

Mark Weiser | The Computer for the 21st Century

Wie bereits erwähnt, hat der Nutzer in der Welt des Ubiquitous Computing nicht mehr eine universell einsetzbare Maschine, sondern die gesamte Funktionalität wird auf viele spezialisierte Geräte aufgeteilt. Dabei ist jedes Gerät genau auf eine Situation und Umgebung zugeschnitten. Natürlich ist es dabei möglich, dass sich einige Geräte in ihrem Einsatzbereich überschneiden, aber der Nutzer wird das für ihn passende auswählen.

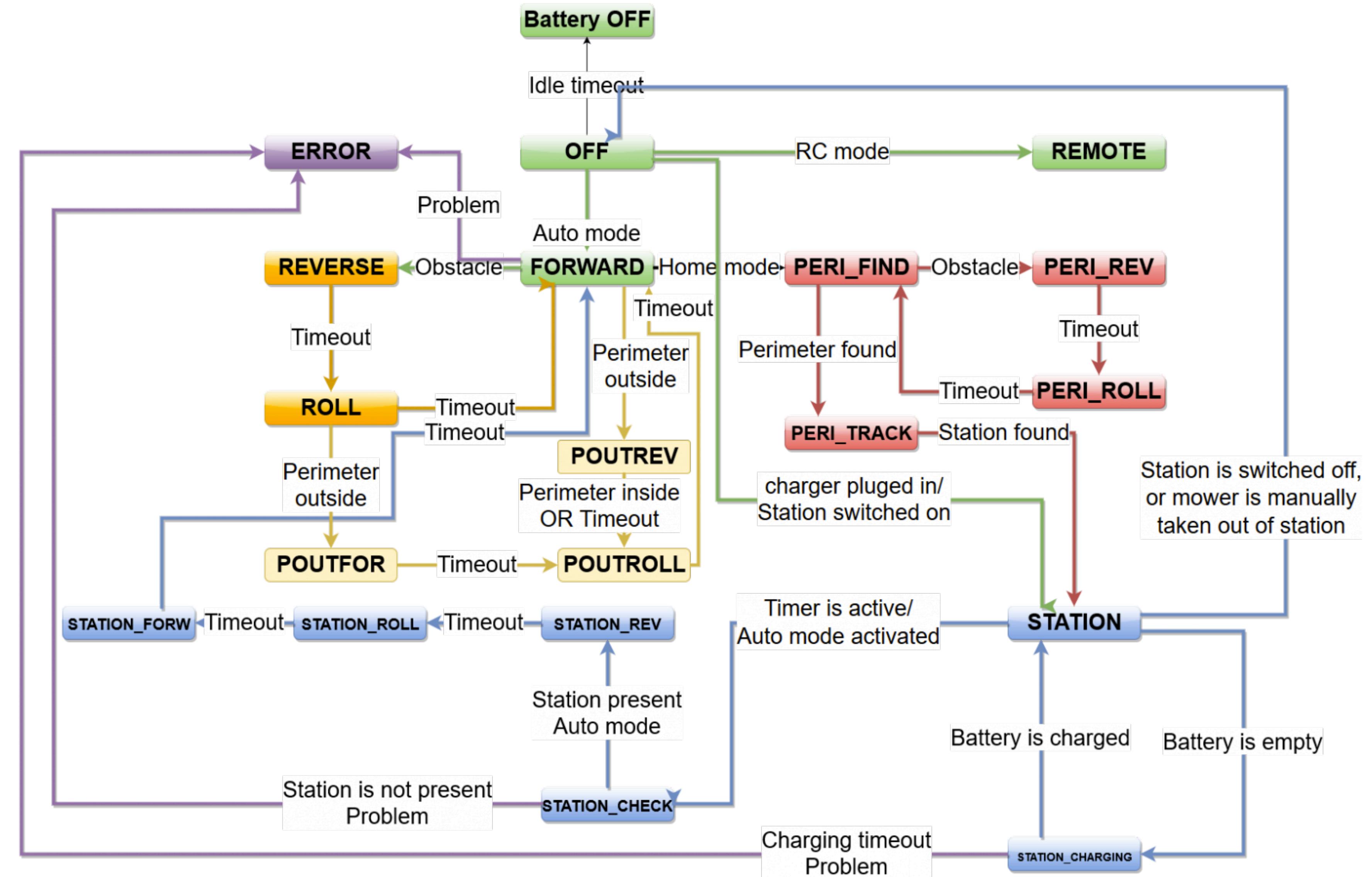
The Major Trends in Computing



interaktion







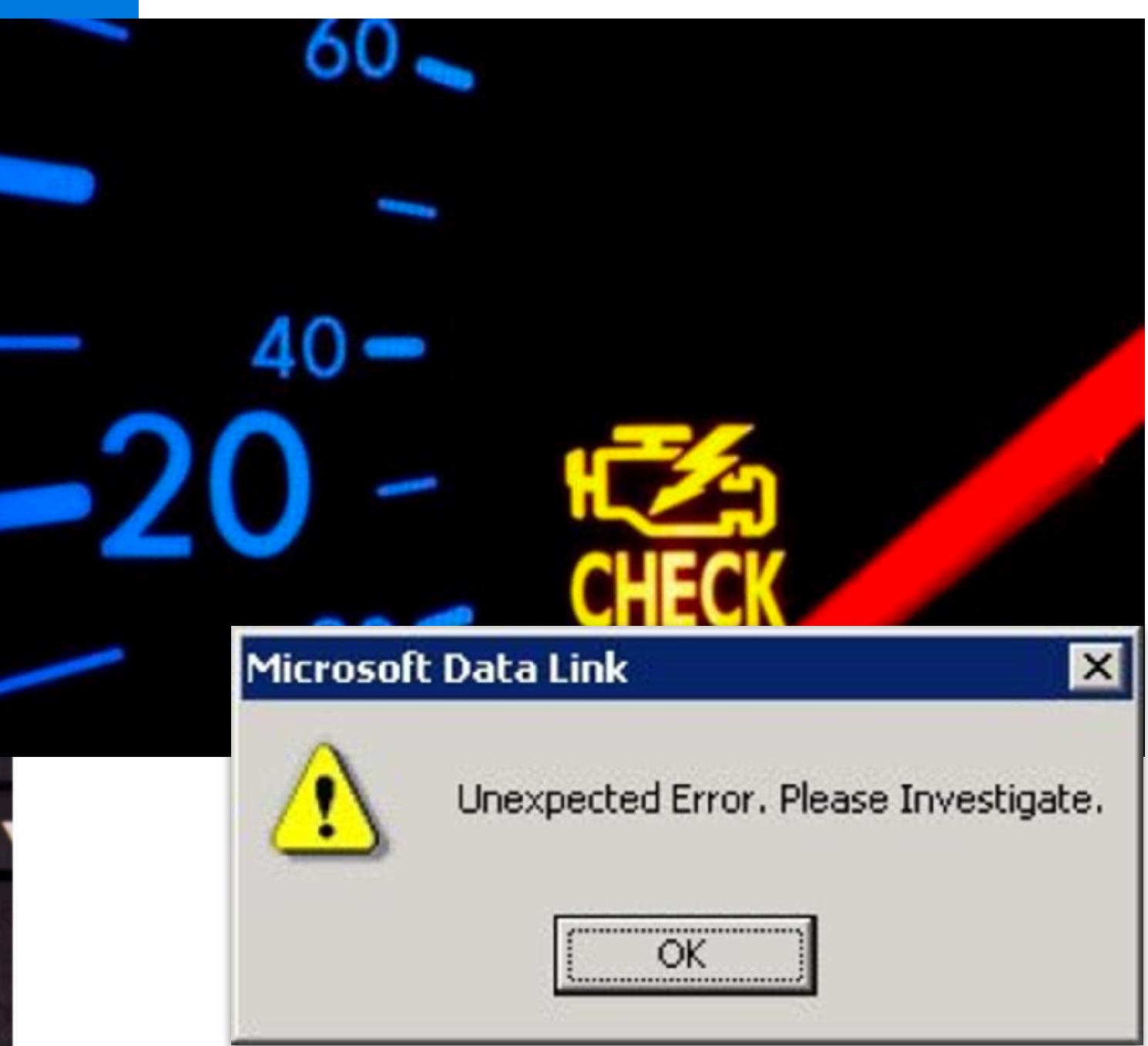
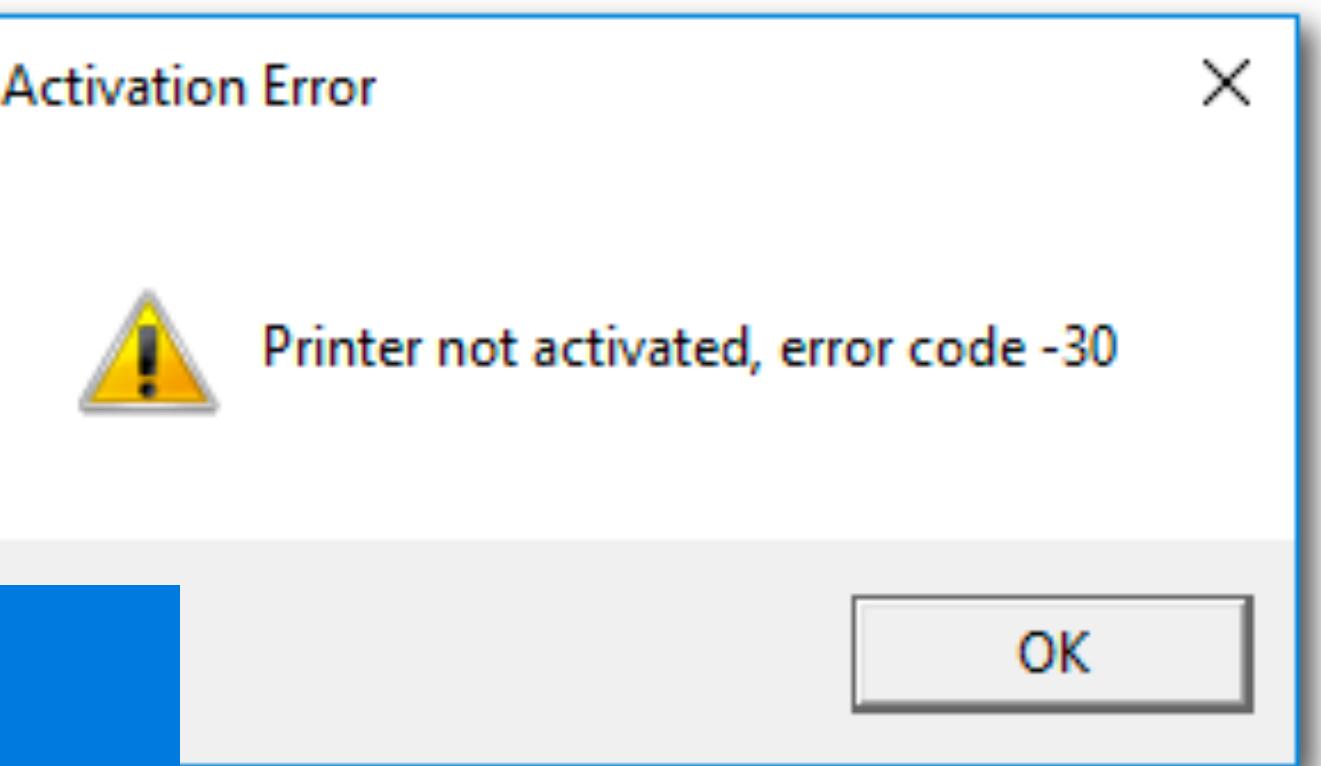
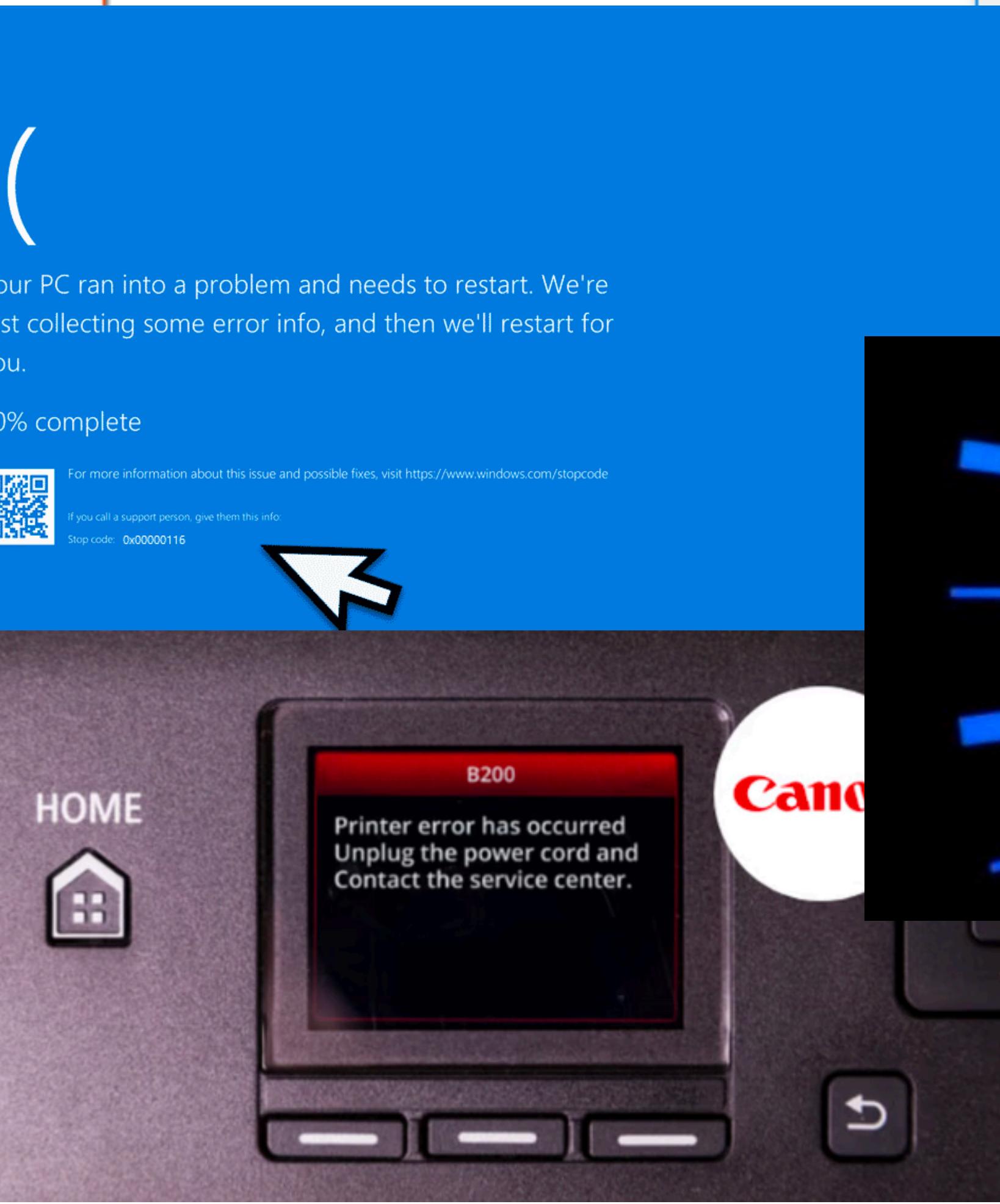
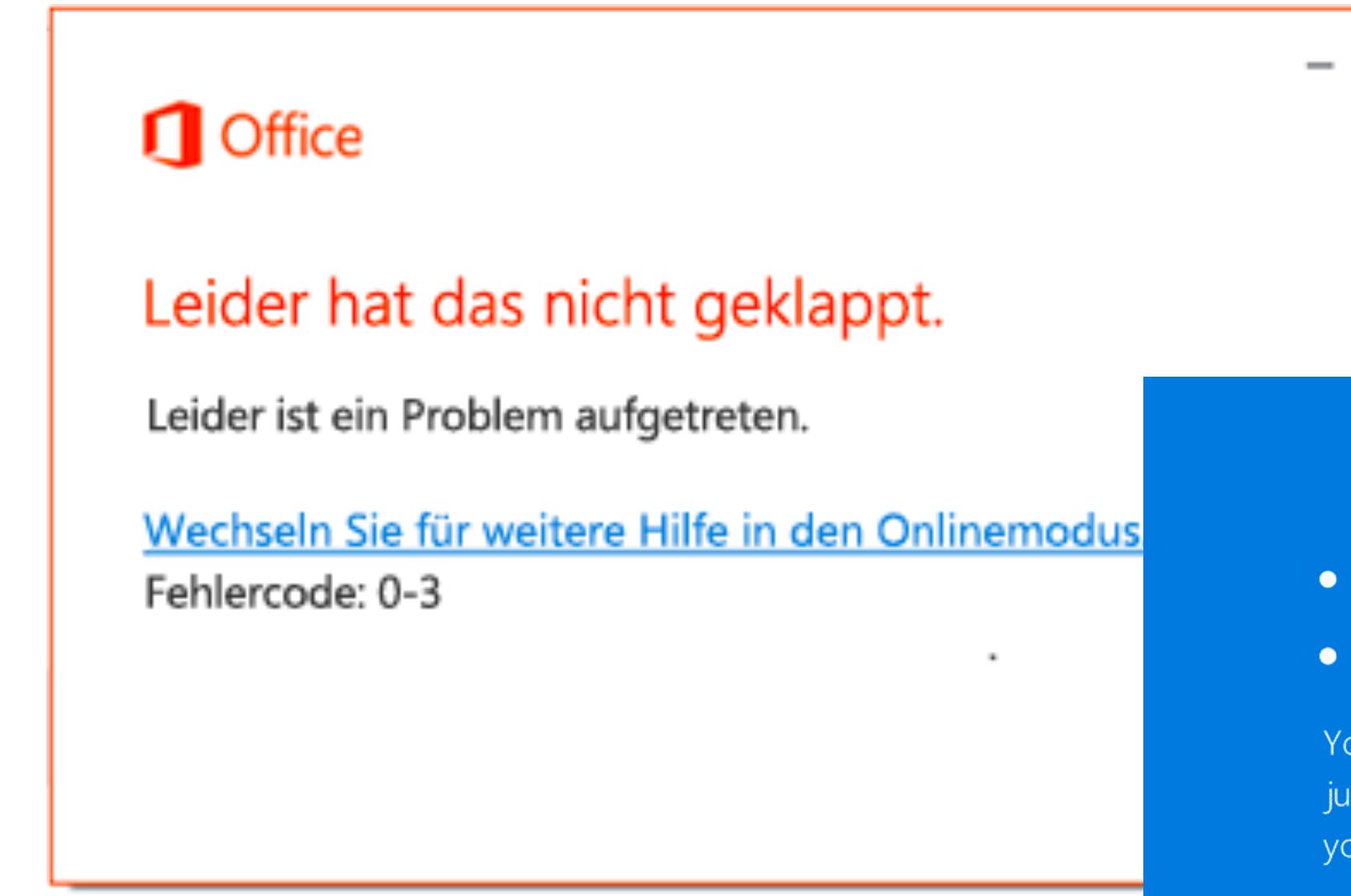












aufgabe

- Menschen greifen auf erlernte Muster und Parallelen zurück
- Wir können diese nutzen, um komplexe Themen und Funktionalitäten verständlich zu machen



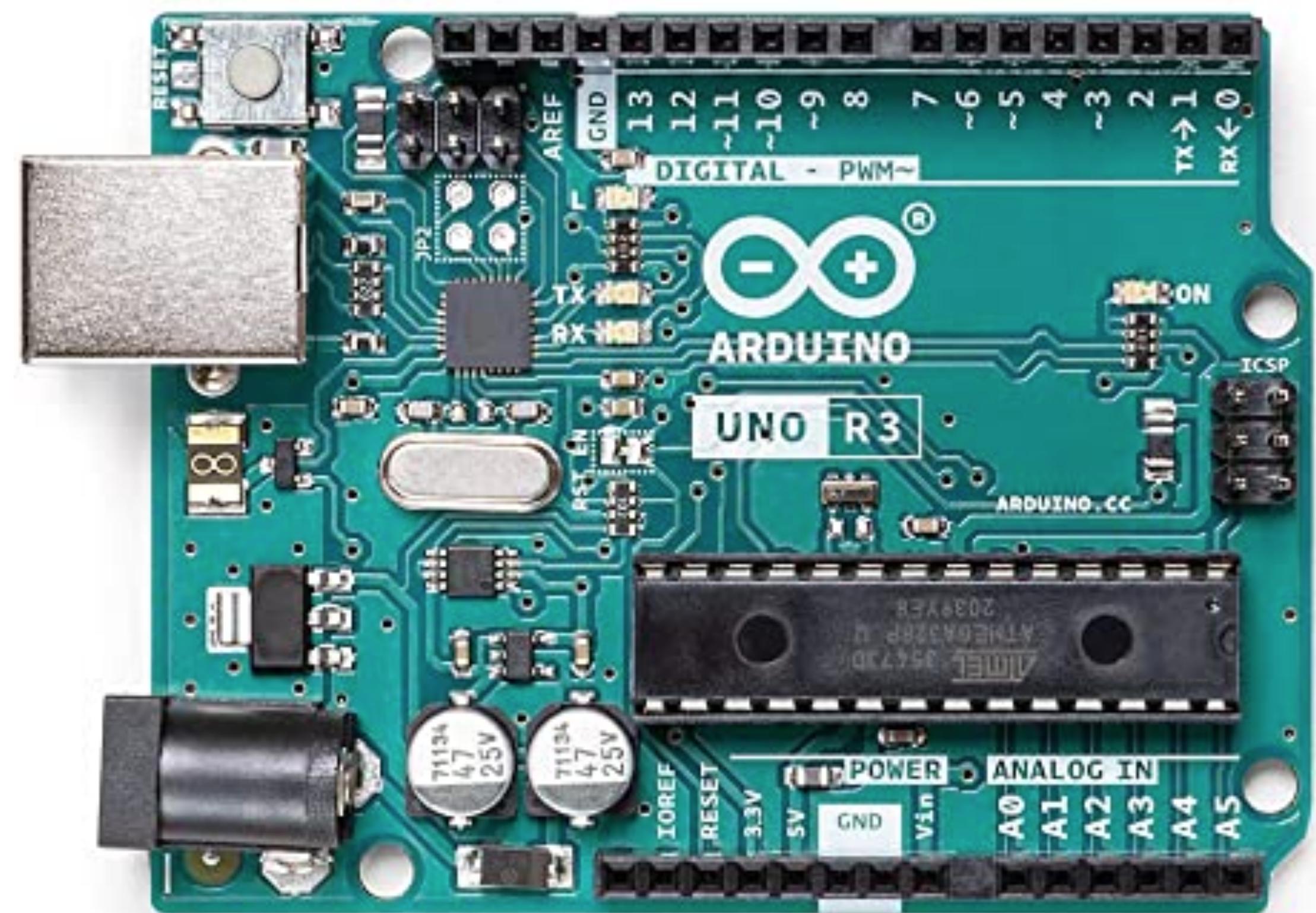
aufgabe

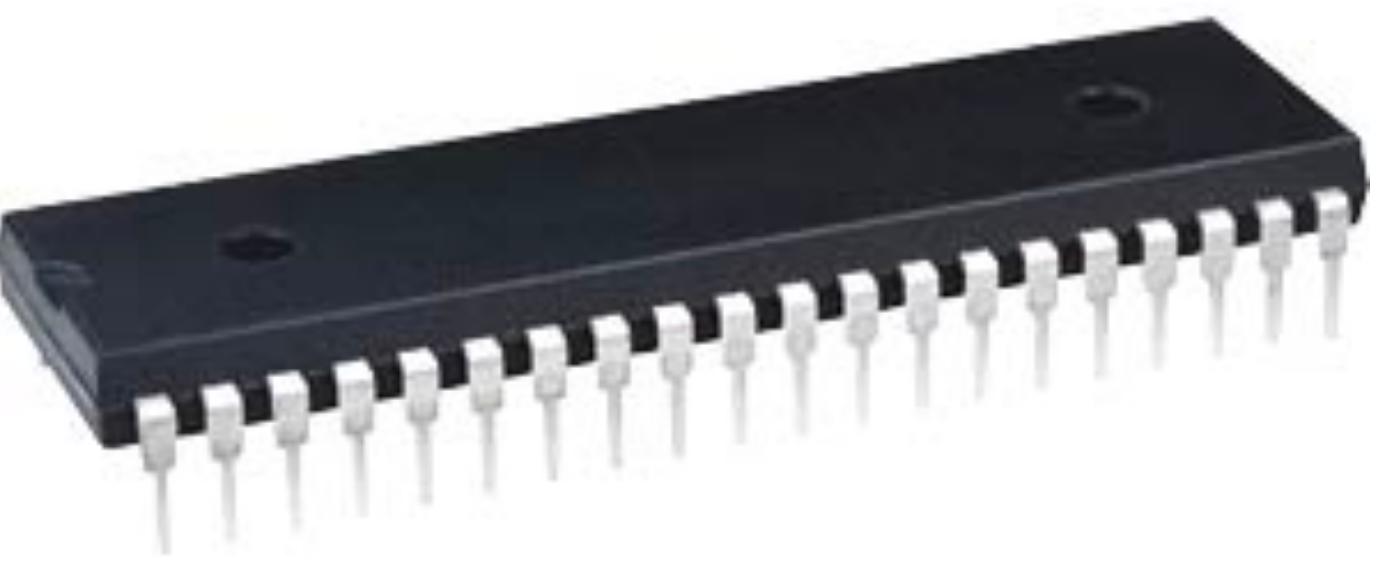
Entwickelt ein Gerät, das eine Funktion oder Aufgabe hat. Das kann ein praktischer Nutzen sein, Unterhaltung oder das Aufbauen einer Beziehung.

- Entwickelt ein Gerätekonzept mit mindest zwei Zuständen (an/ aus, offen/geschlossen, heiß/kalt,...)
- Stellt heraus, welche Parameter oder welche Mechanismen Nutzer:innen kommuniziert werden müssen
- Sammelt Beispiele aus anderen Bereichen für diese Interaktion
- Setzt eure Idee prototypisch und interaktiv in einem gestellten Karton um ohne ihn äußerlich zu verändern

tech









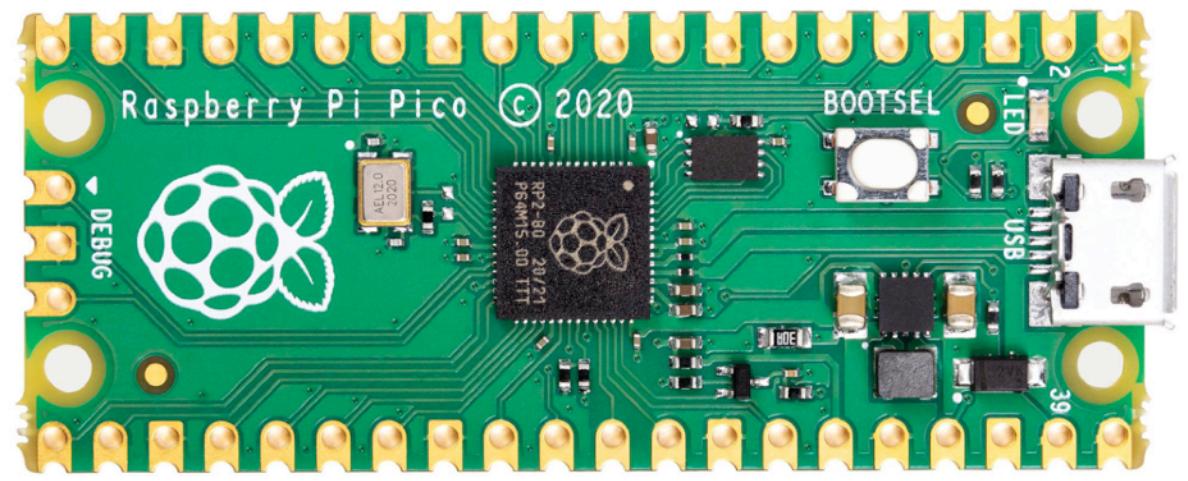
Blink | Arduino 1.8.5

This example code is in the public domain.
<http://www.arduino.cc/en/Tutorial/Blink>

```
/*
 * the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level)
    delay(1000);                      // wait for a second
    digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW
    delay(1000);                      // wait for a second
}
```

32 Arduino/Genuino Uno on COM1



Thonny - /Users/max/dev/pi-pico/code/multi-servo.py @ 16 : 1

Files

This computer
/ Users / max / dev / pi-pico / code

- library.py
- multi-servo.py
- sensor.py
- servo-fade.py
- servo.py
- ultra-sound.py

multi-servo.py

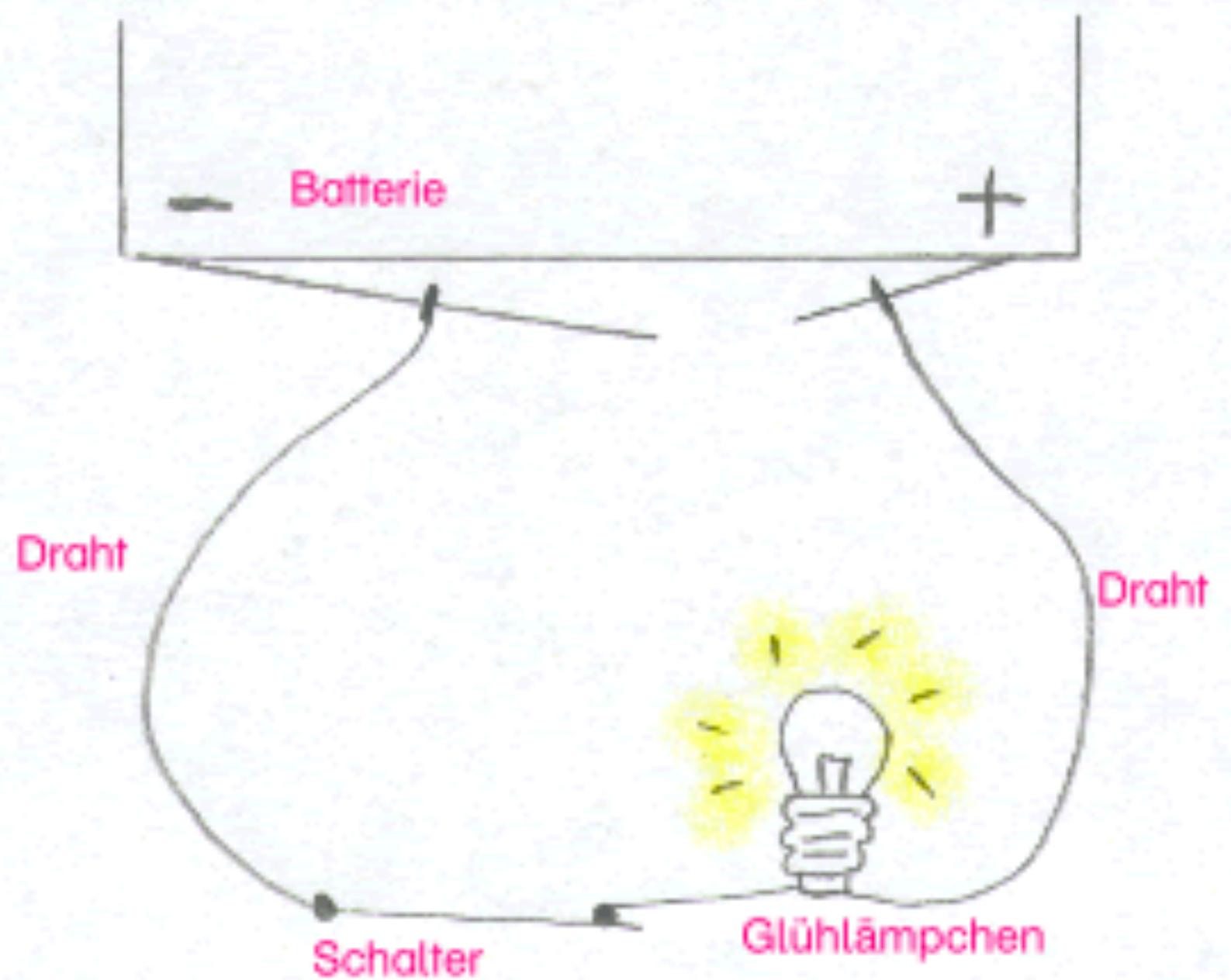
```
1 import time
2 from library import Servo
3
4 servo1 = Servo(pin=28) #To be changed according to the pin used
5 servo2 = Servo(pin=27) #To be changed according to the pin used
6
7 while True:
8     print(0)
9     servo1.move(0) # turns the servo to 0°.
10    servo2.move(0) # turns the servo to 0°.
11    time.sleep(0.5)
12    print(1)
13    servo1.move(90) # turns the servo to 90°.
14    servo2.move(90) # turns the servo to 90°.
15    time.sleep(0.5)
```

Shell

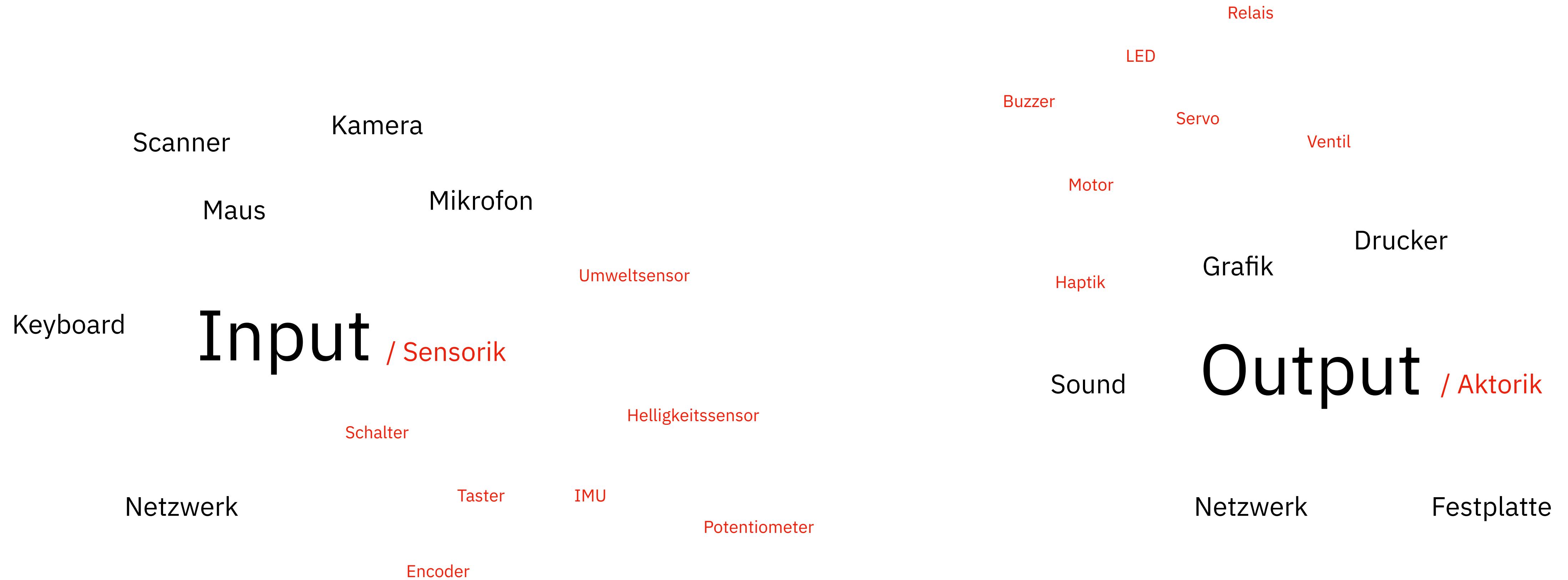
```
Traceback (most recent call last):
  File "main.py", line 6, in <module>
    KeyboardInterrupt:
MicroPython v1.22.2 on 2024-02-22; Raspberry Pi Pico
Type "help()" for more information.
>>> Connection lost -- read failed: [Errno 6] Device or resource busy
```

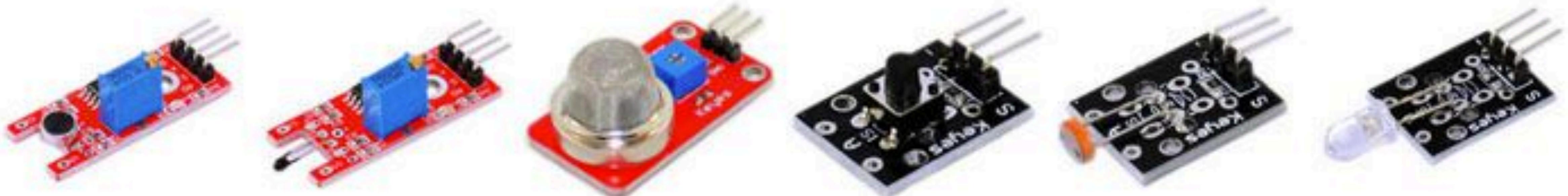
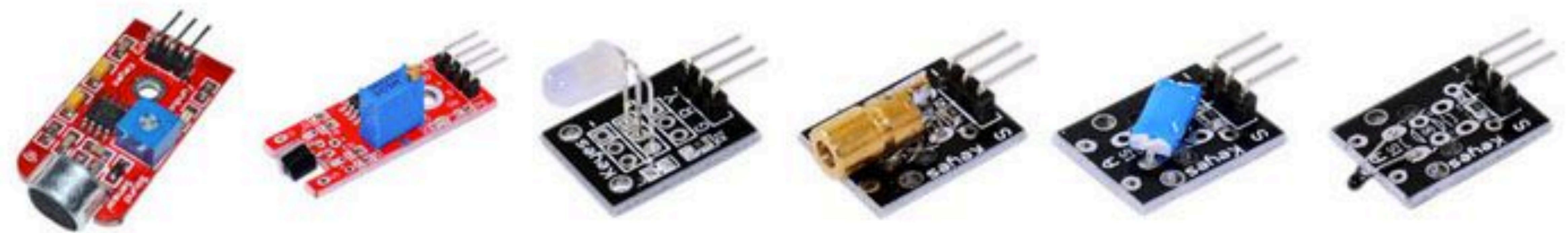
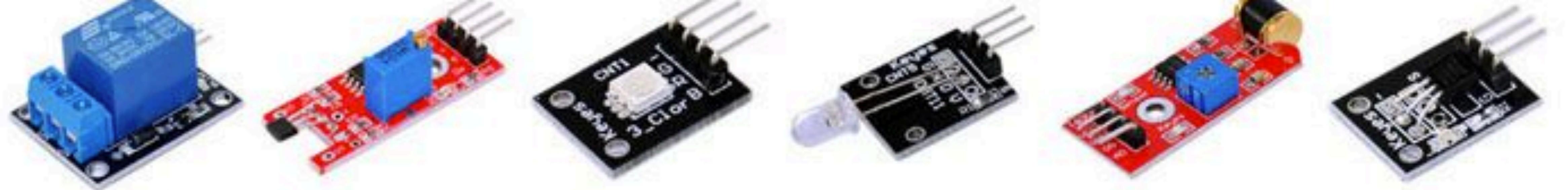
MicroPython (Raspberry Pi Pico) • Board

A screenshot of the Thonny Python IDE interface. The main window displays a Python script named 'multi-servo.py'. The script uses the 'Servo' class from a 'library' module to control two servos connected to pins 28 and 27. It prints the current servo position to the console and alternates between 0 and 90 degrees. The Thonny interface includes a file browser, a code editor with syntax highlighting, and a terminal window at the bottom showing a MicroPython error message about a connection loss due to a device being busy.



Die Digitaltechnik ermöglicht eine Verknüpfung von Ursache (Input) und Wirkung (Output) über die Möglichkeiten der direkten, diskreten Verbindung beider hinaus.





Computer

- Betriebssystem
- Festplatte
- Tastatur / Maus
- USB / Video
- Kamera
- Bildschirm
- Programme und Apps



Mikrocontroller

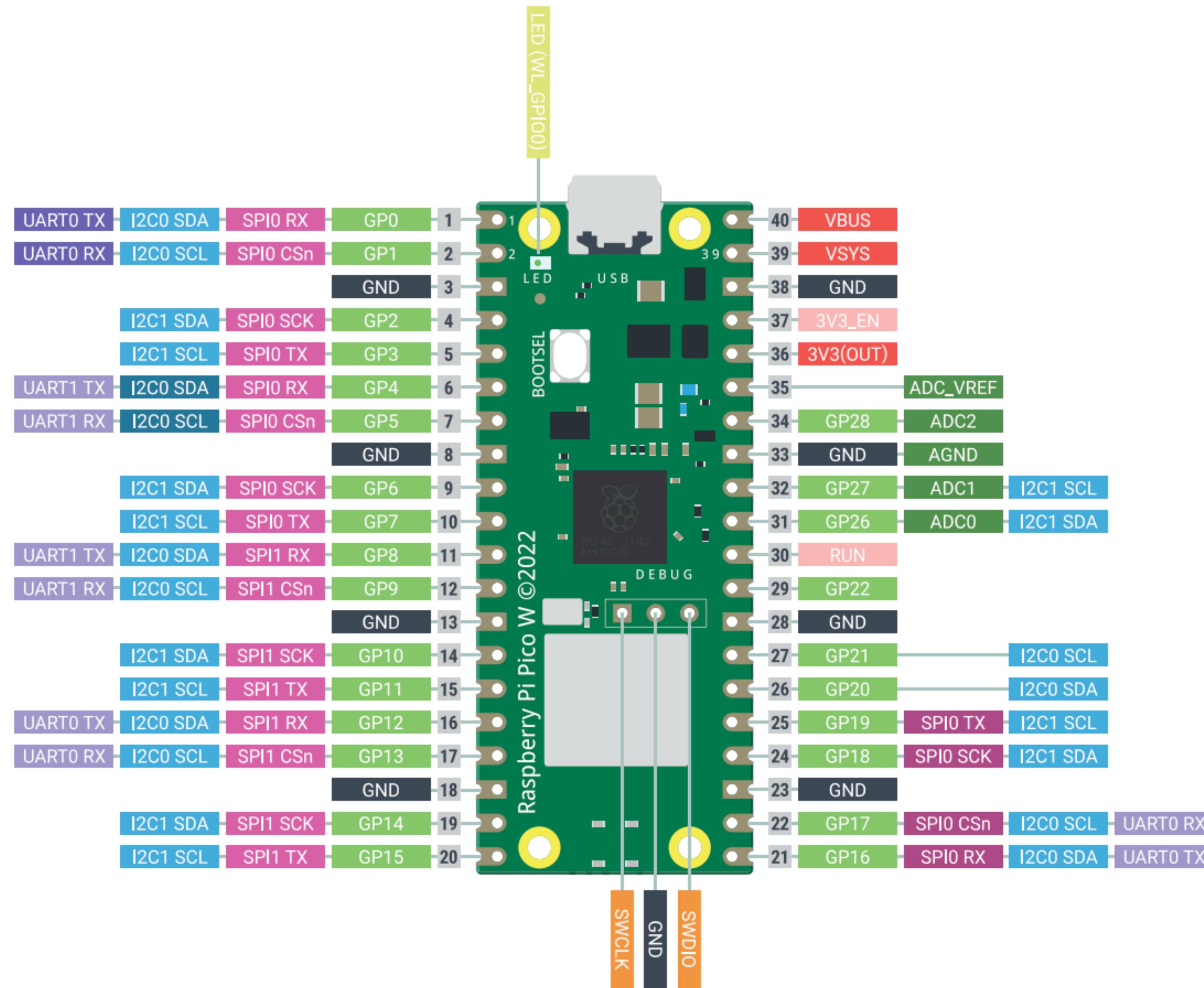
- GPIO (general purpose in / out)
- Stromversorgung
- Software

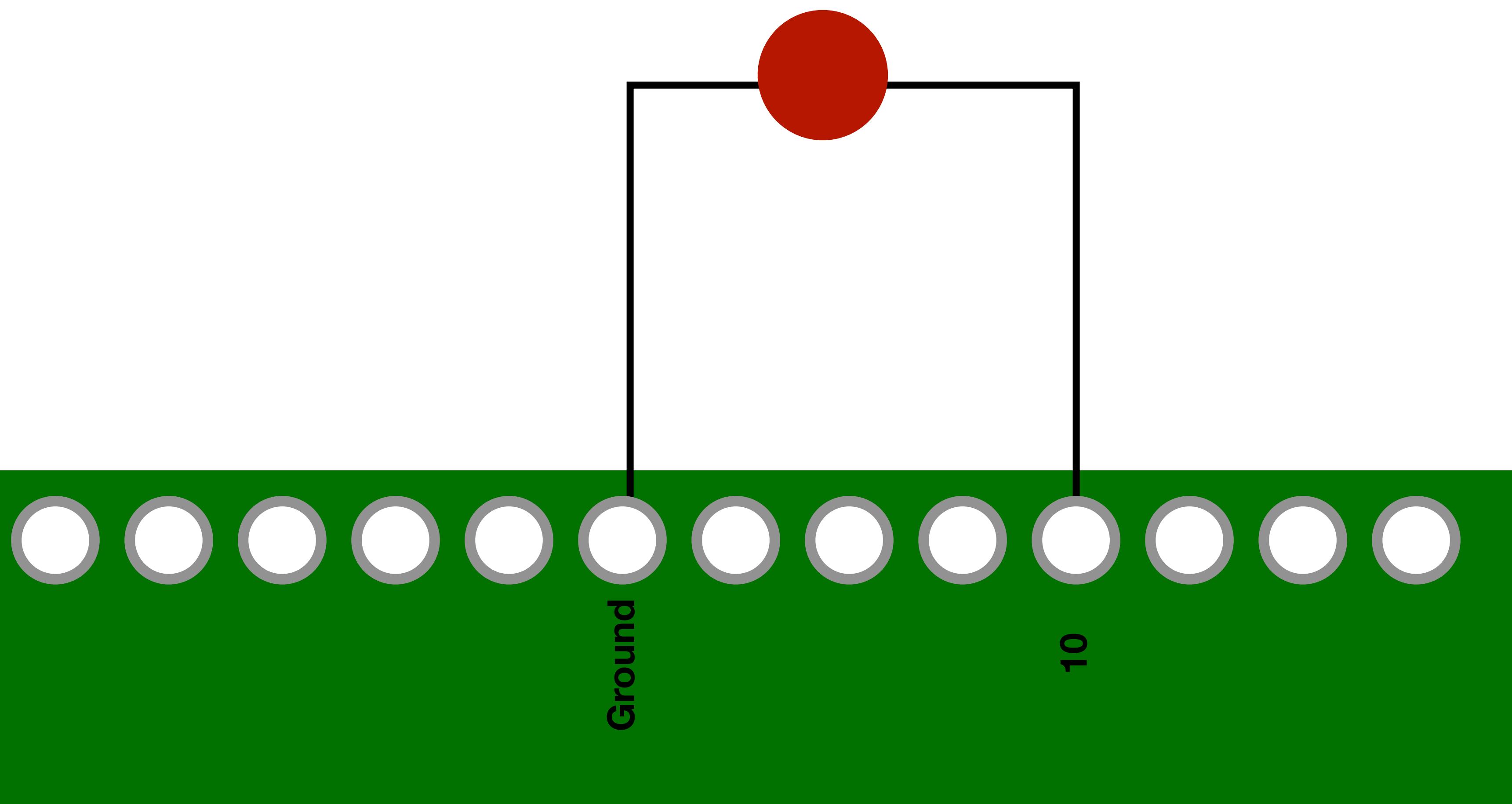


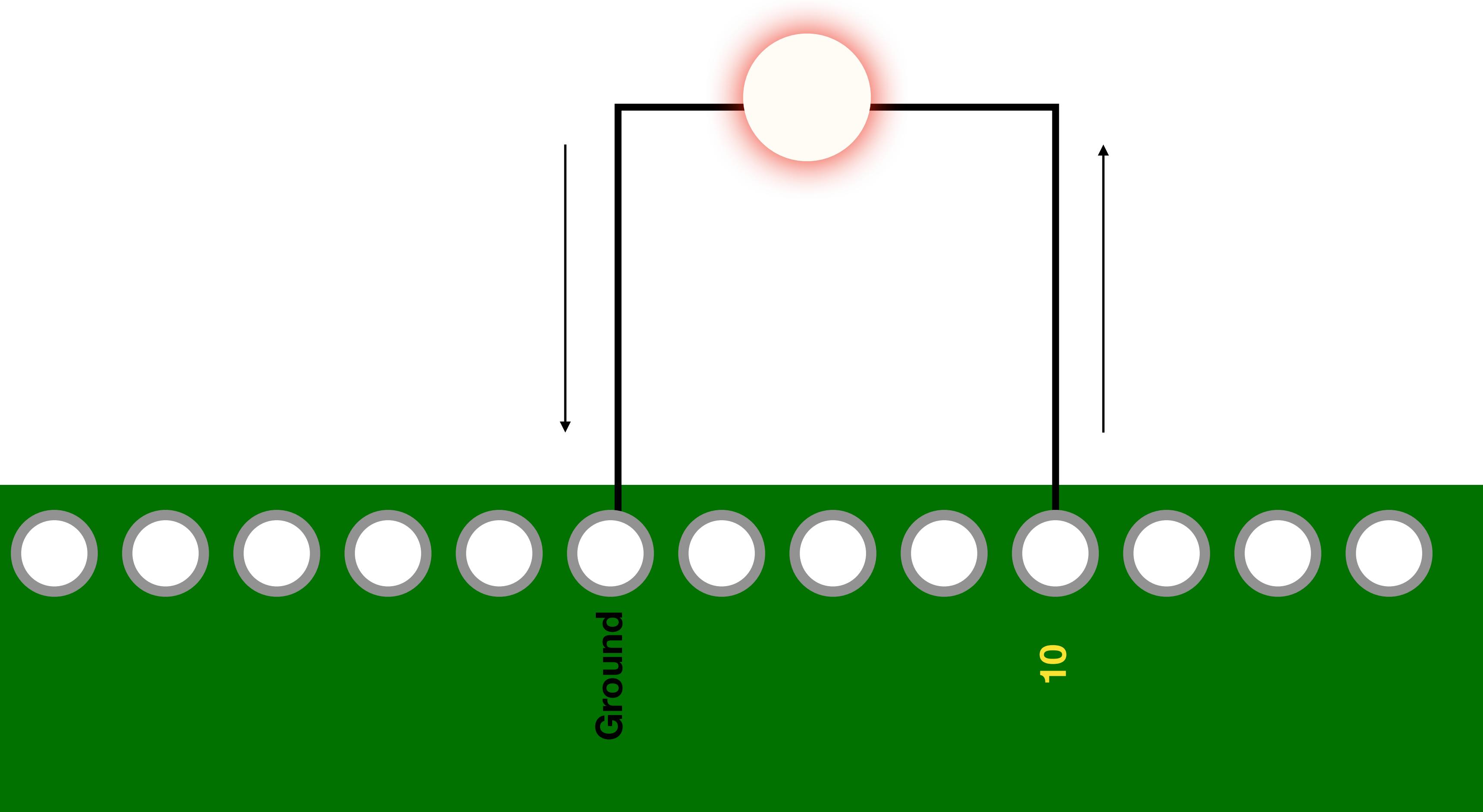
RP2040

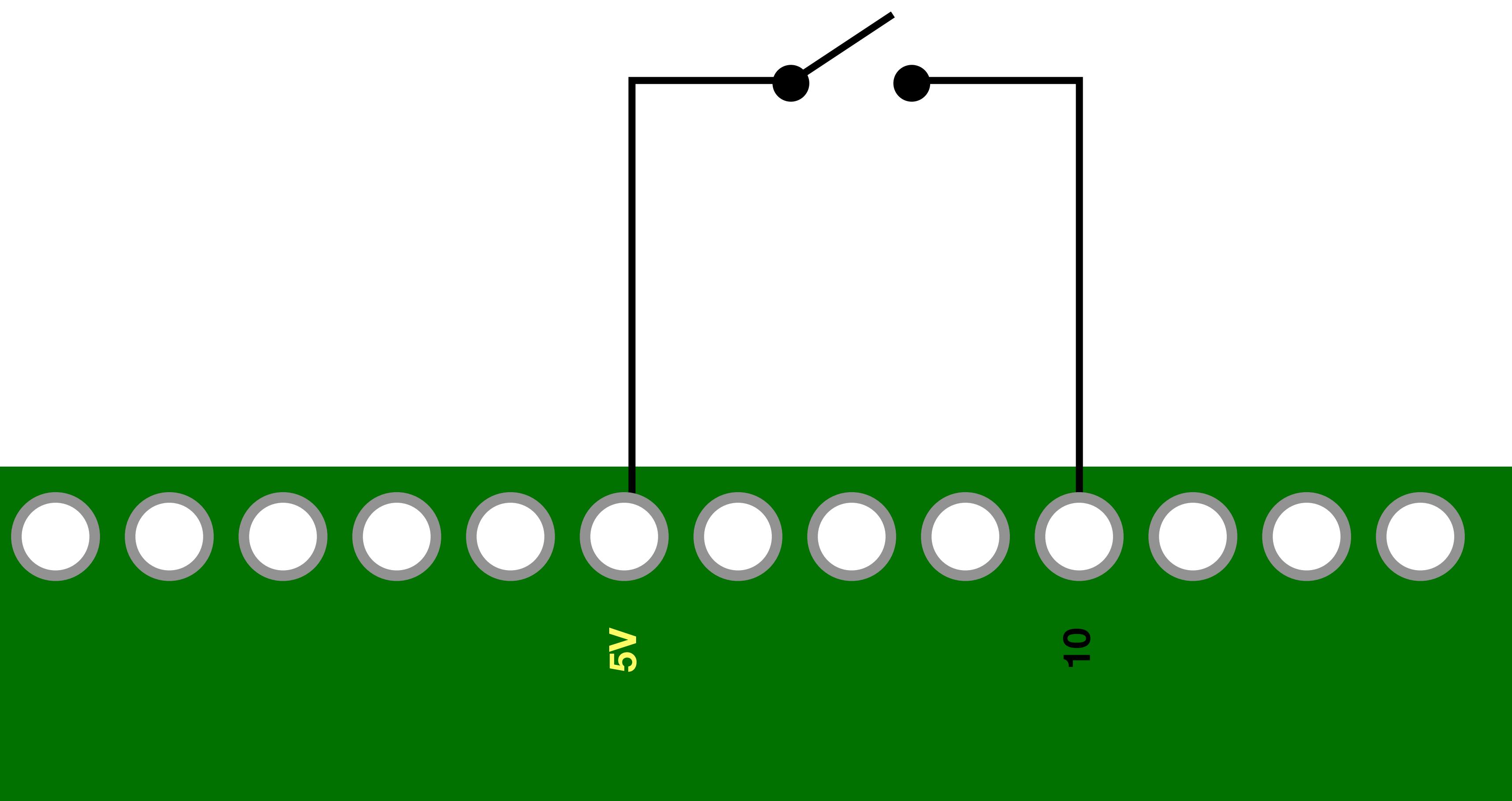


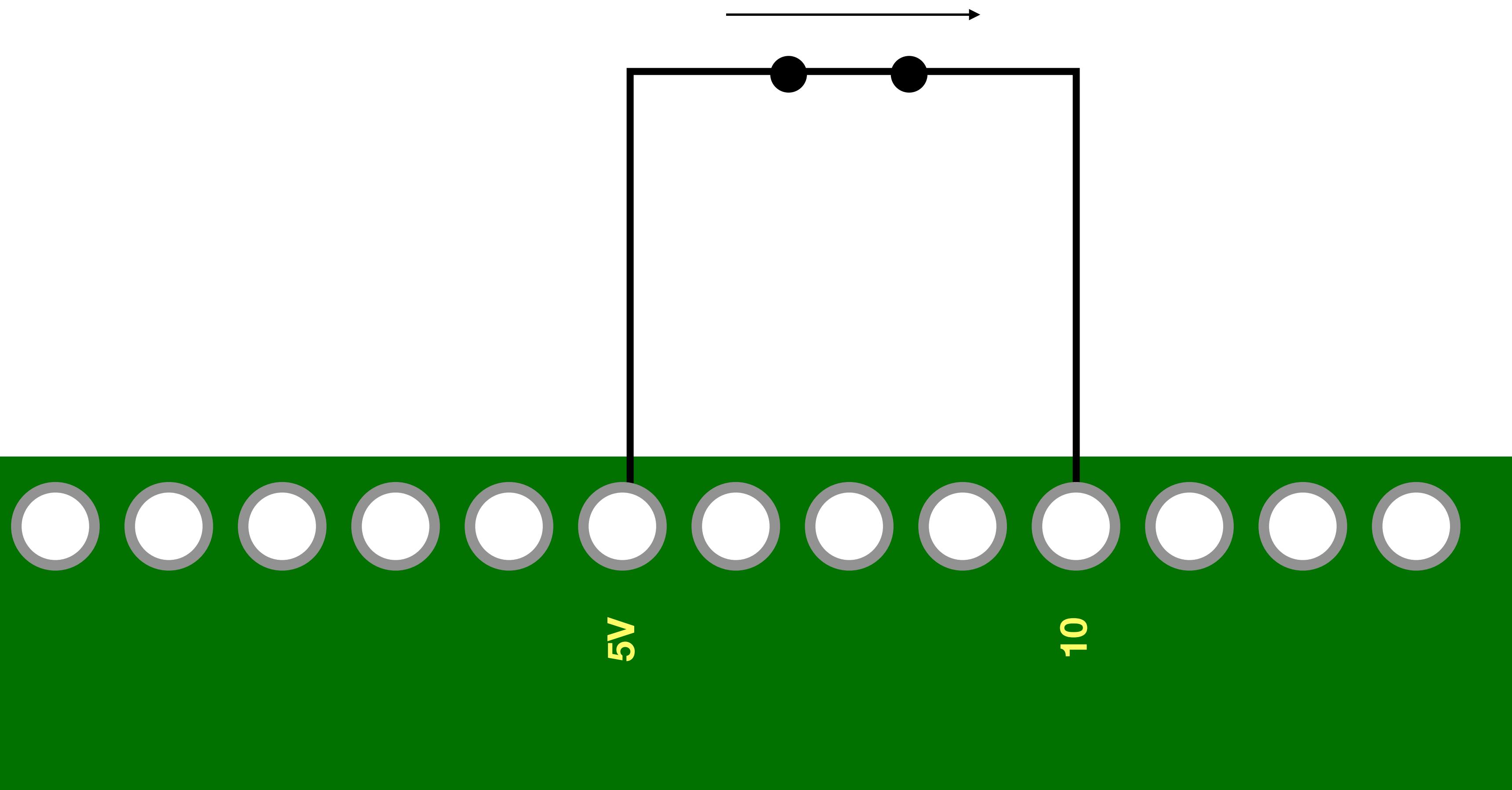
Infineon 43439

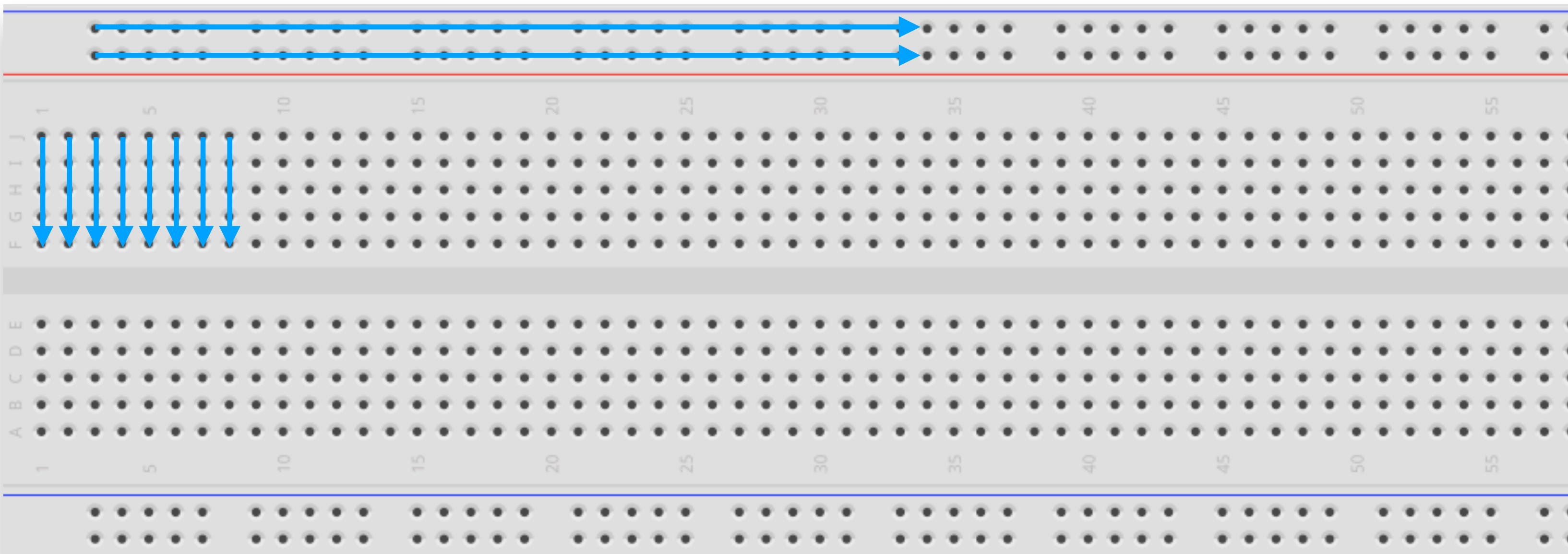


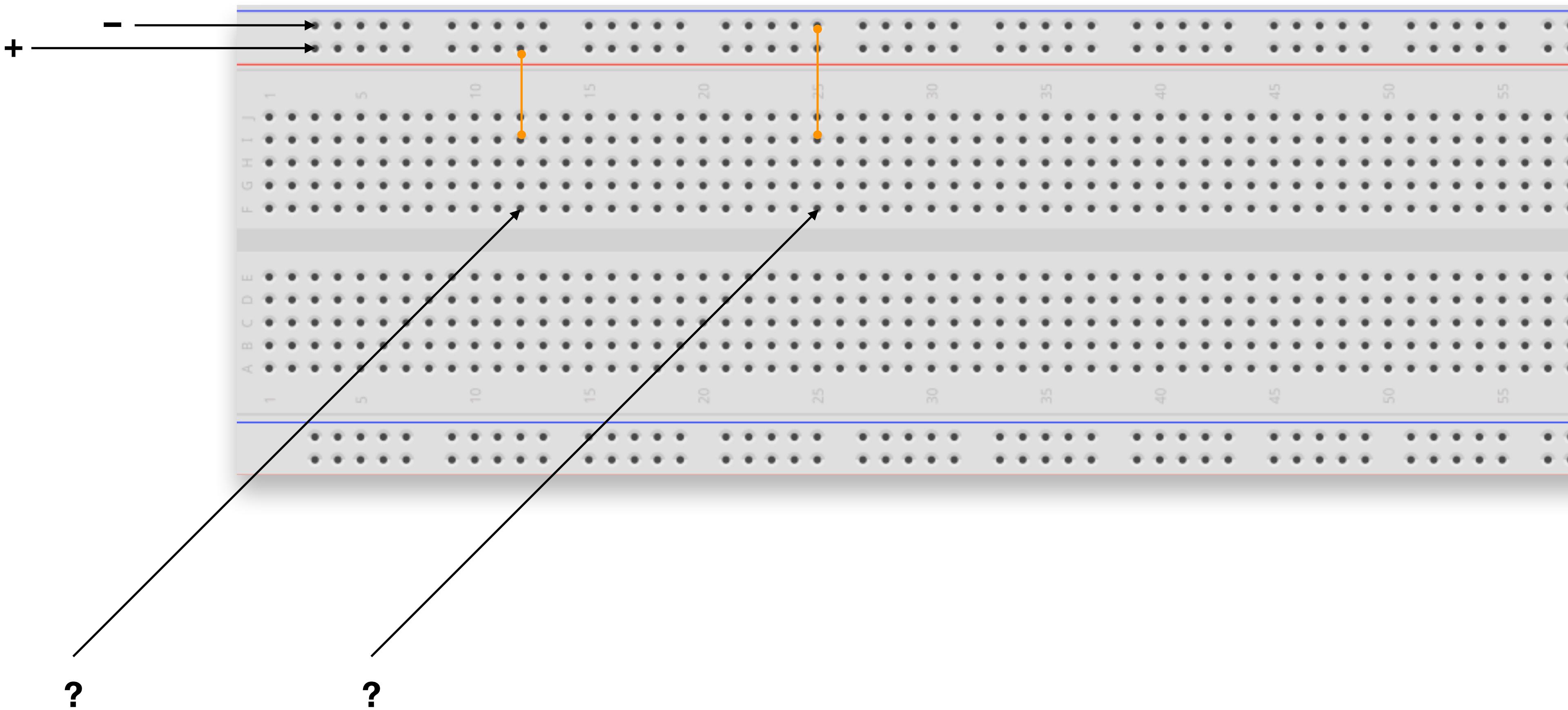


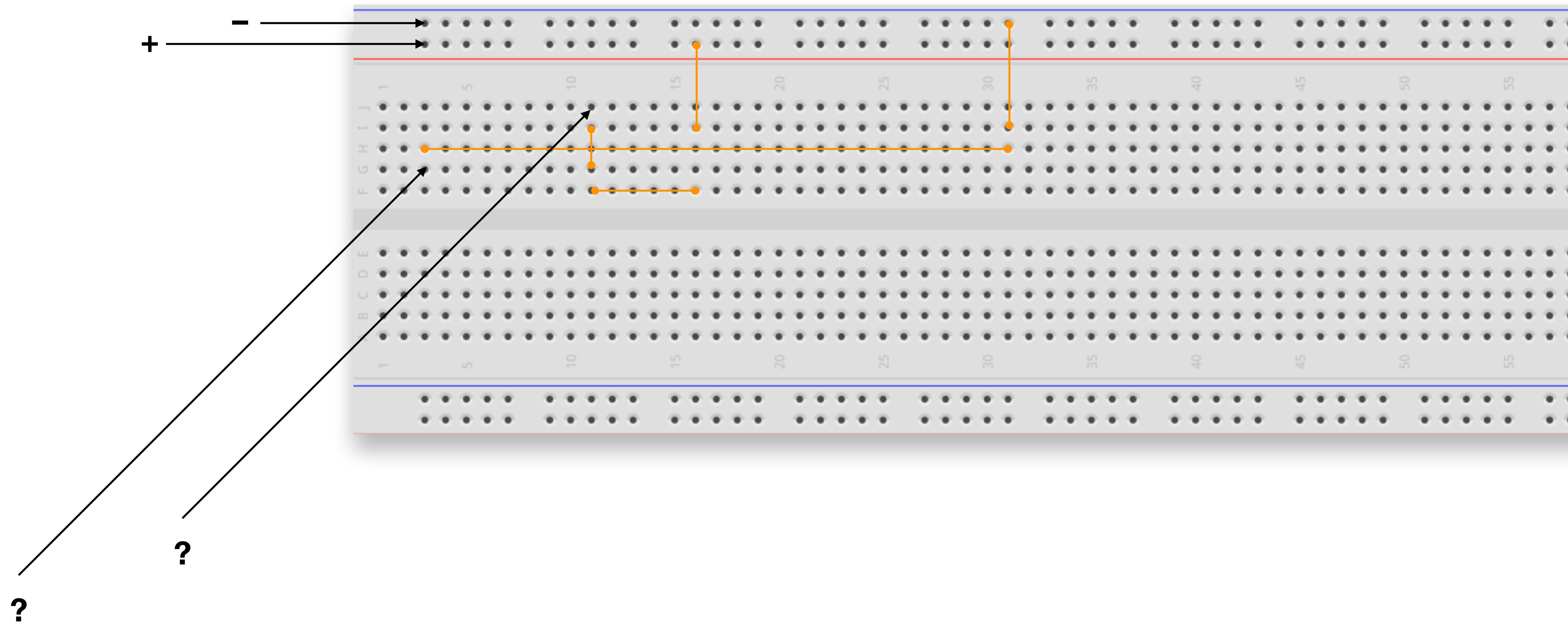










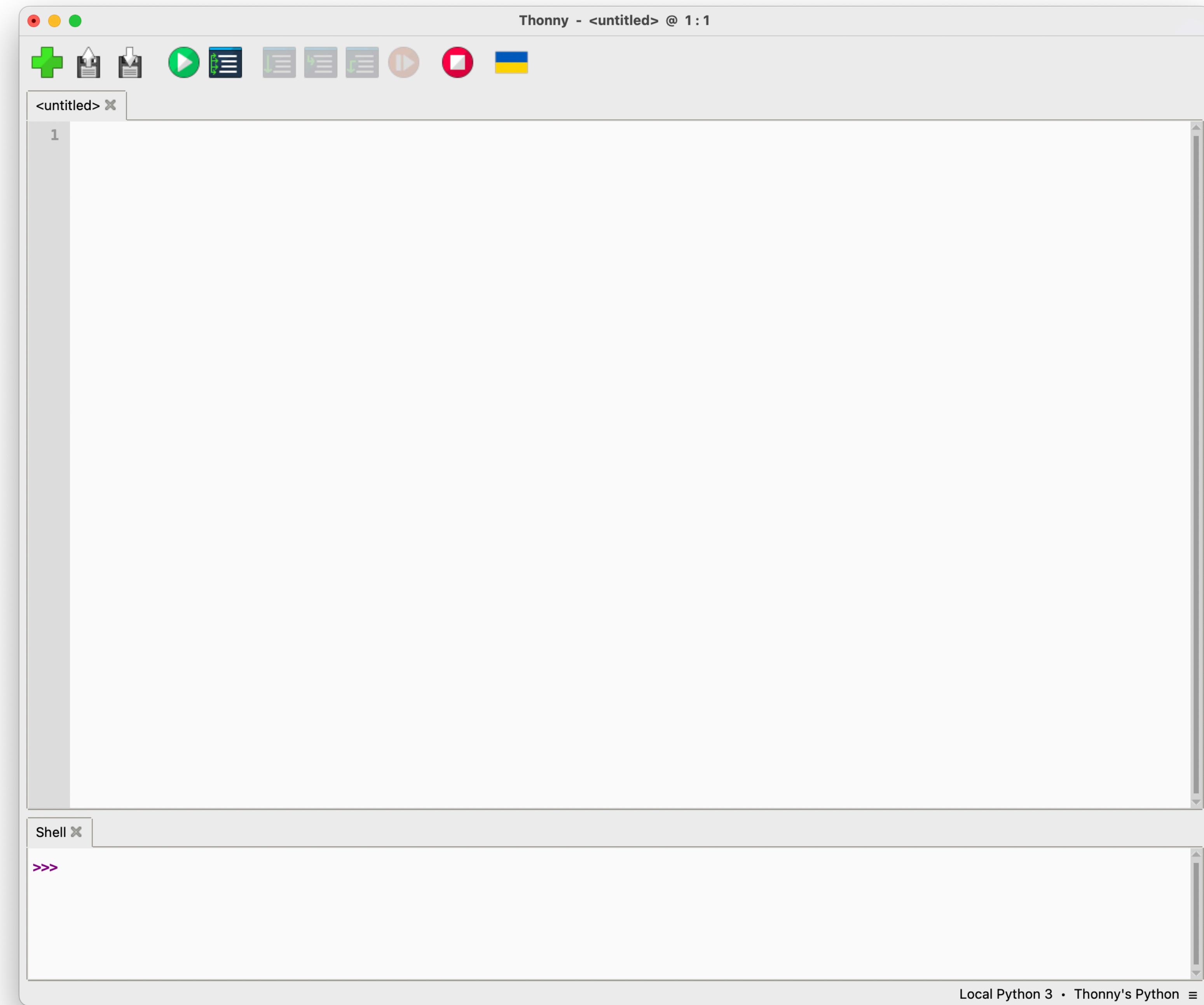


Projektdaten & Beispiele

github.com/max-hans/25-hfg-karlsruhe-arduino

Programmierumgebung

thonny.org



Thonny - <untitled> @ 1:15

The screenshot shows the Thonny Python IDE interface. At the top, there's a toolbar with various icons: a green plus sign for creating new files, a file icon, a save icon, a play button, a code editor icon, and several other icons for file operations like copy, paste, and search. Below the toolbar is a menu bar with '<untitled> * X' and 'Run current script'. The main area contains a code editor with the following content:

```
1 print("hello")
```

Below the code editor is a 'Shell' window with the title 'Shell X'. It displays the Python prompt '>>>'. A red arrow points from the bottom right corner of the shell window towards the status bar at the bottom of the screen.

Local Python 3 • Thonny's Python ≡

A screenshot of a GitHub repository page for "max-hans/pi-pico".

The repository is public and has 11 commits. The latest commit was made by "max-hans" at 4c8c0a2 · 2 hours ago. The commits include:

- rm (branch: main)
- add basics (branch: code)
- intro (branch: resources)
- gitignore (branch: .gitignore)
- add seutp and multi-servo (branch: image.png)
- rm (branch: readme.md)

The README file contains the following content:

pi pico examples

introduction

you will need thonny to run and upload your python code. get it [here](#).

A screenshot of the Thonny IDE interface is shown at the bottom, displaying the file "multi-servo.py" with some Python code.

About

No description, website, or topics provided.

Readme

Activity

0 stars

1 watching

0 forks

[Report repository](#)

Releases

No releases published

Packages

No packages published

Languages

Python 100.0%