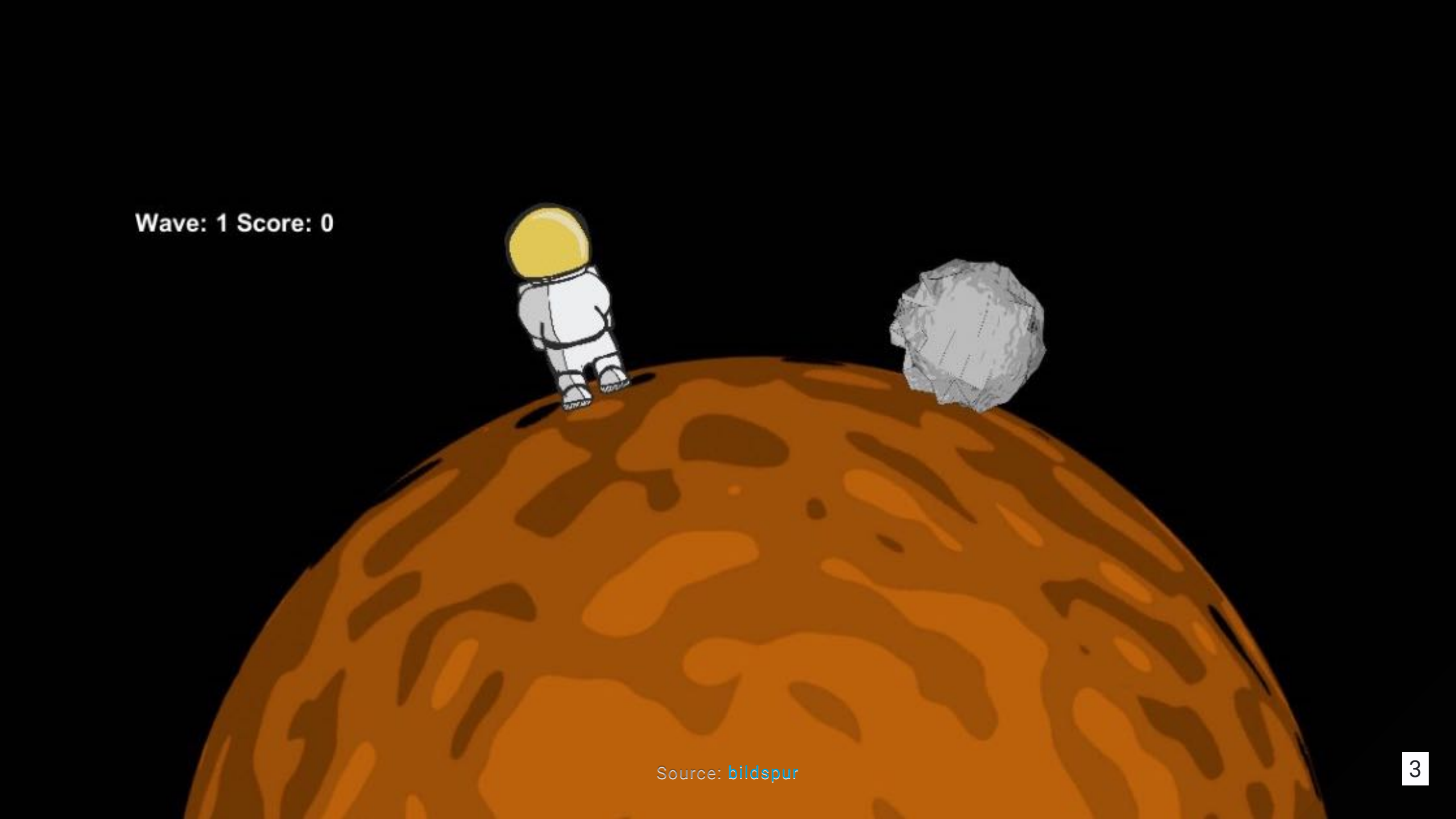
Arduino - Processing

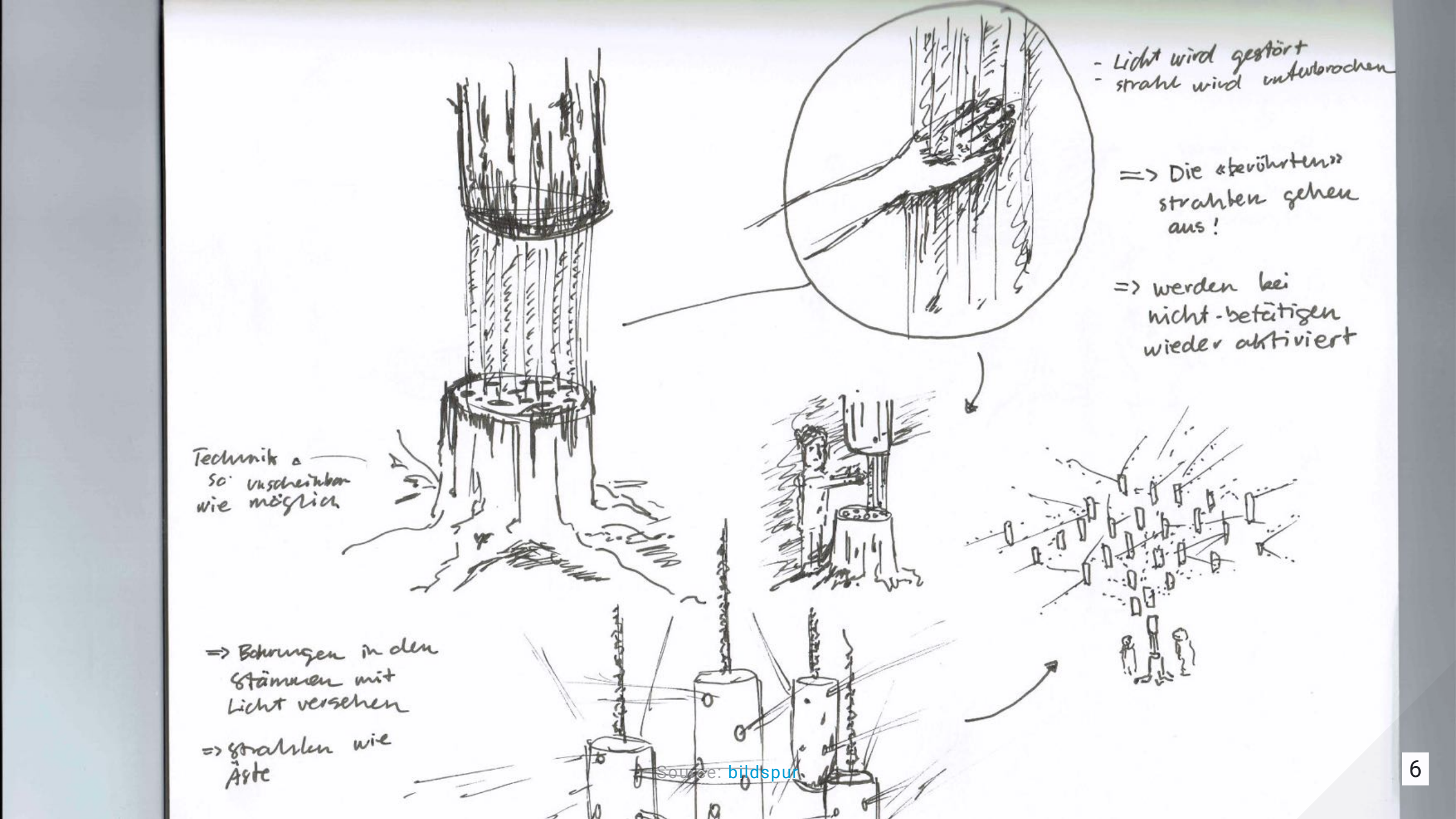
“ Ready to connect! ”

Why?

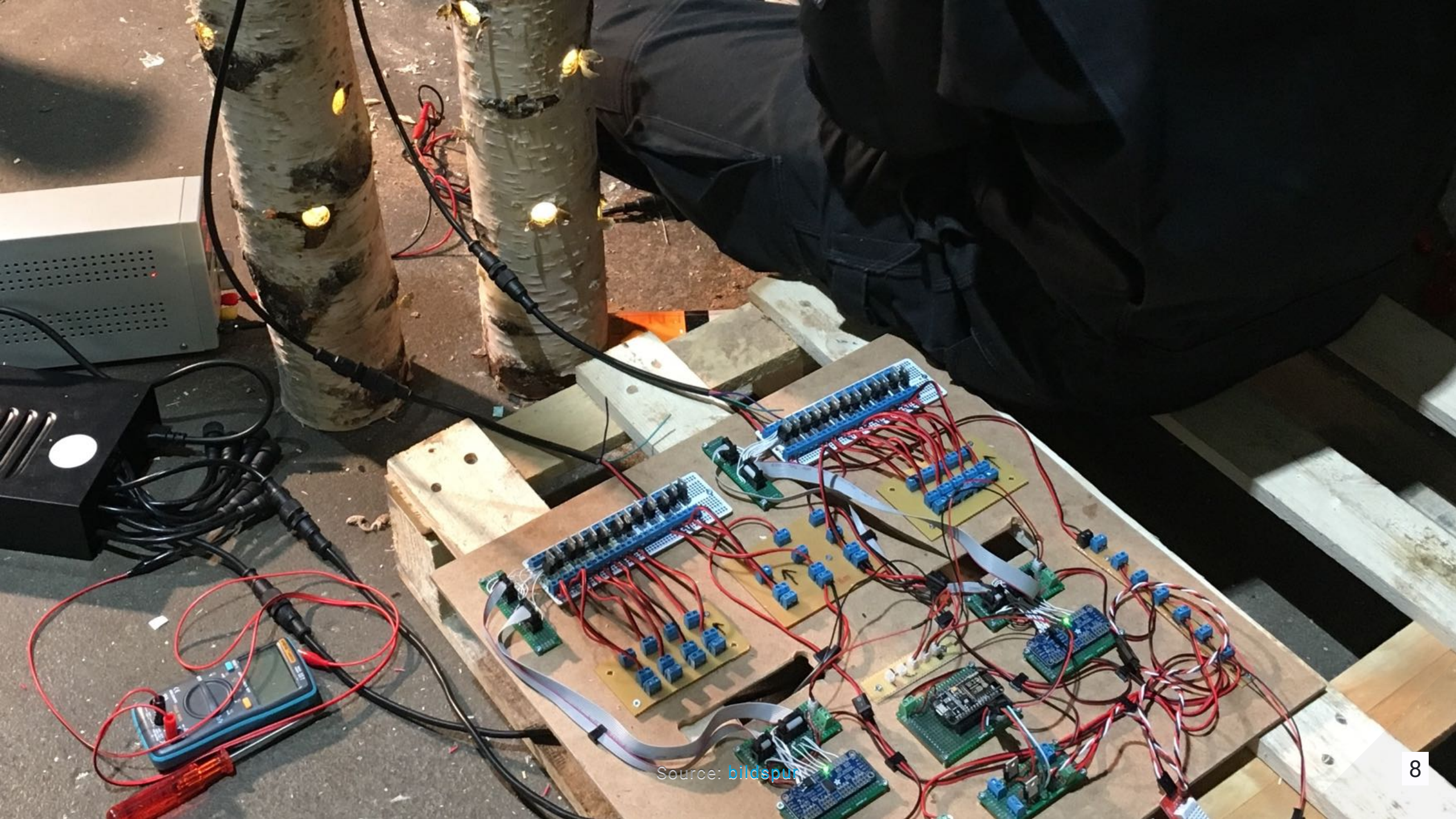


Source: bildspur

Source: bildspur

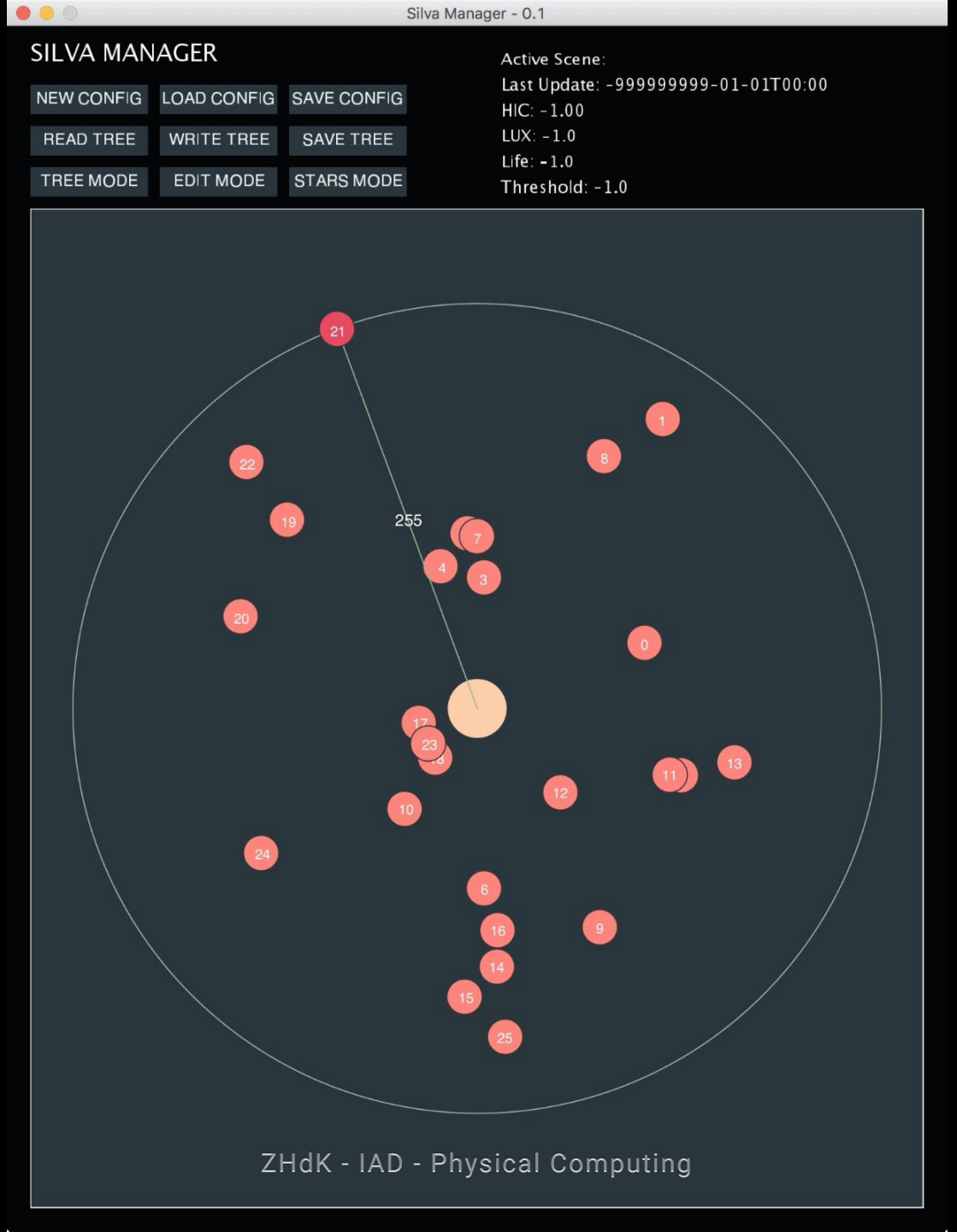


Source: bildspur



Source: bildspur

Source: bildspur

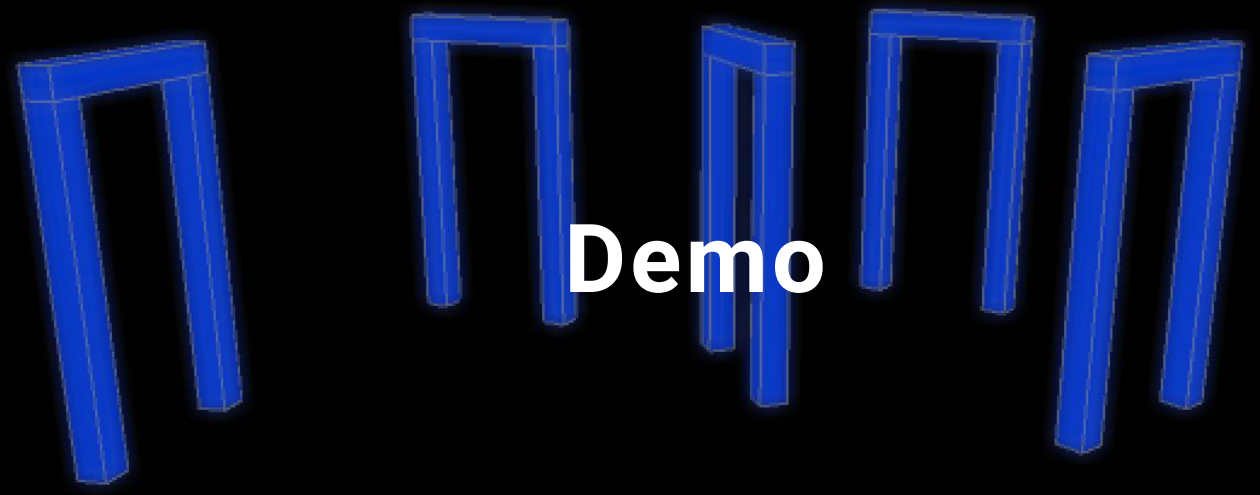

Silva Manager - 0.1
SILVA MANAGER
NEW CONFIG
LOAD CONFIG
SAVE CONFIG
READ TREE
WRITE TREE
SAVE TREE
TREE MODE
EDIT MODE
STARS MODE
Active Scene:
Last Update: -999999999-01-01T00:00
HIC: -1.00
LUX: -1.0
Life: -1.0
Threshold: -1.0
255
ZHdK - IAD - Physical Computing

Source: bildspur

MCU detected: true

attached: true

FPS: 116



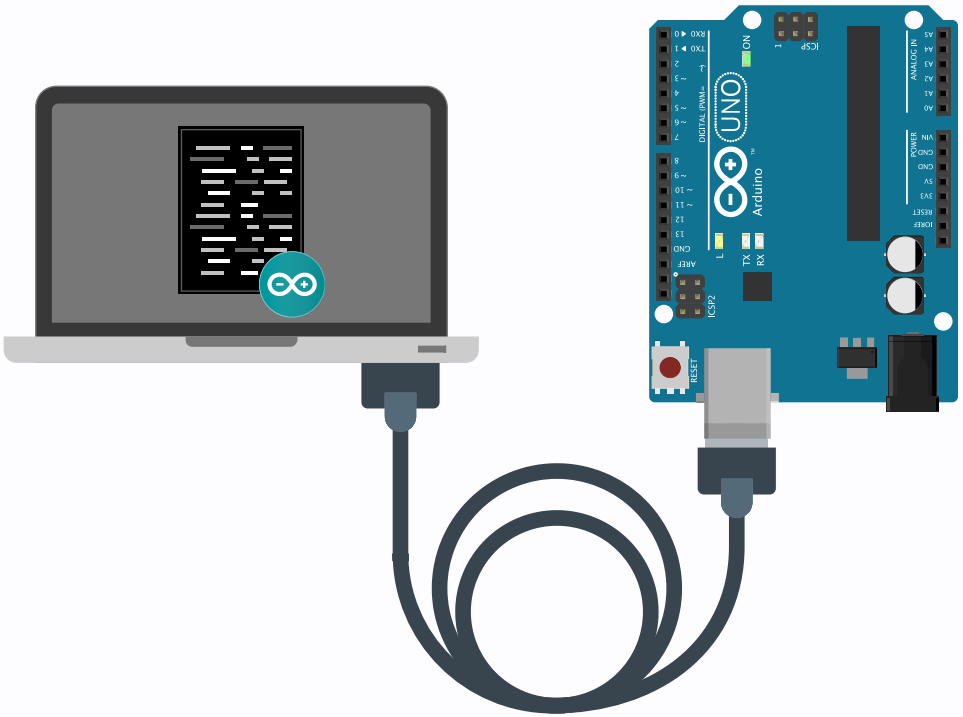
Demo

Debug "SLR;0.16 0.46 0.80;0.16 0.46 0.80;0.16 0.46 0.80;0.16 0.46 0.80;0.16 0.46 0.80;"

Source: bildspur

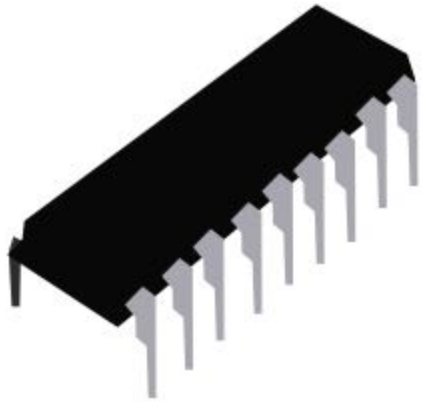
Why?

- Sensor Input
- Hardware Control
- Network
- Visualisation
- Debugging / Coding



Serial

SERIAL

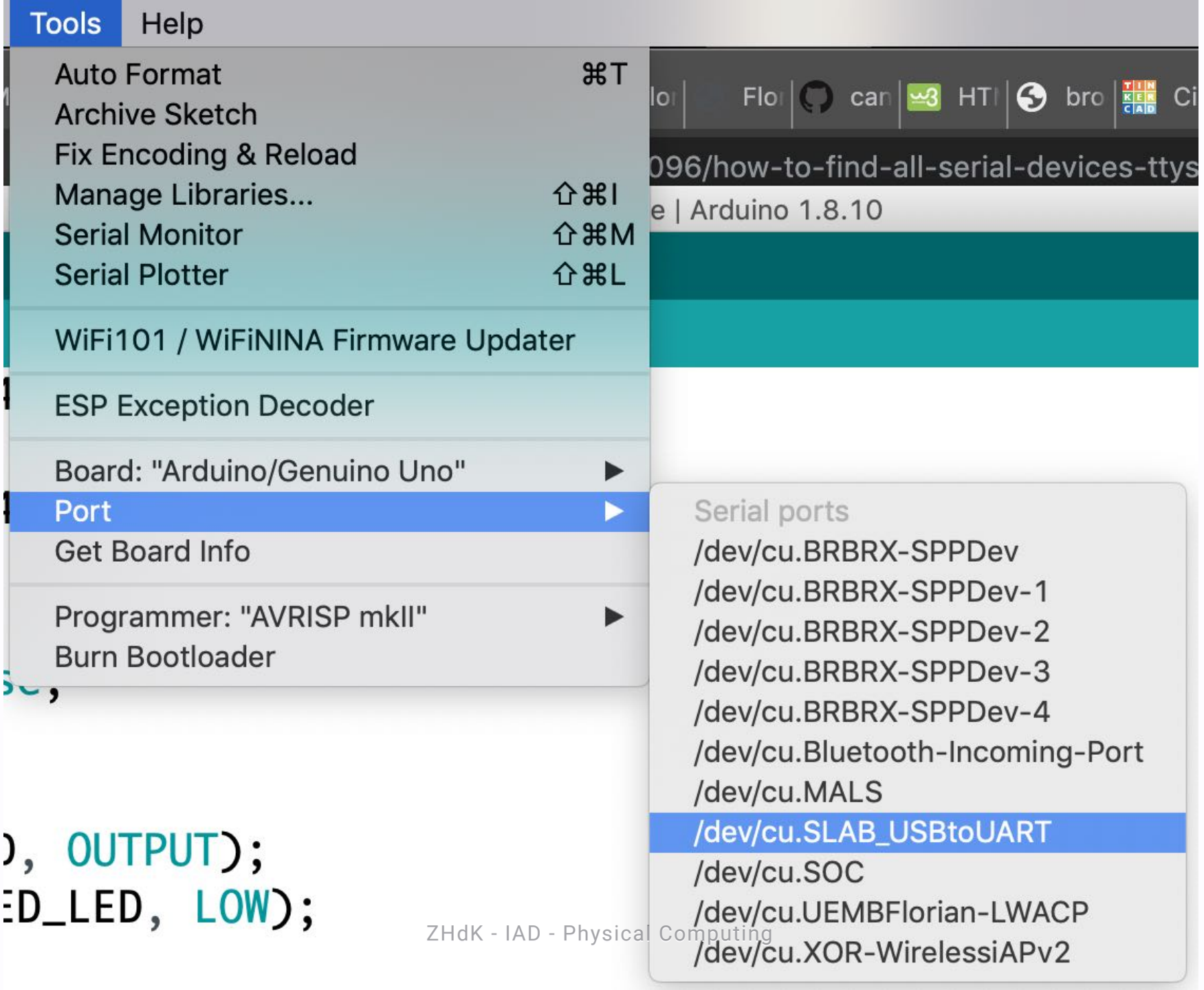


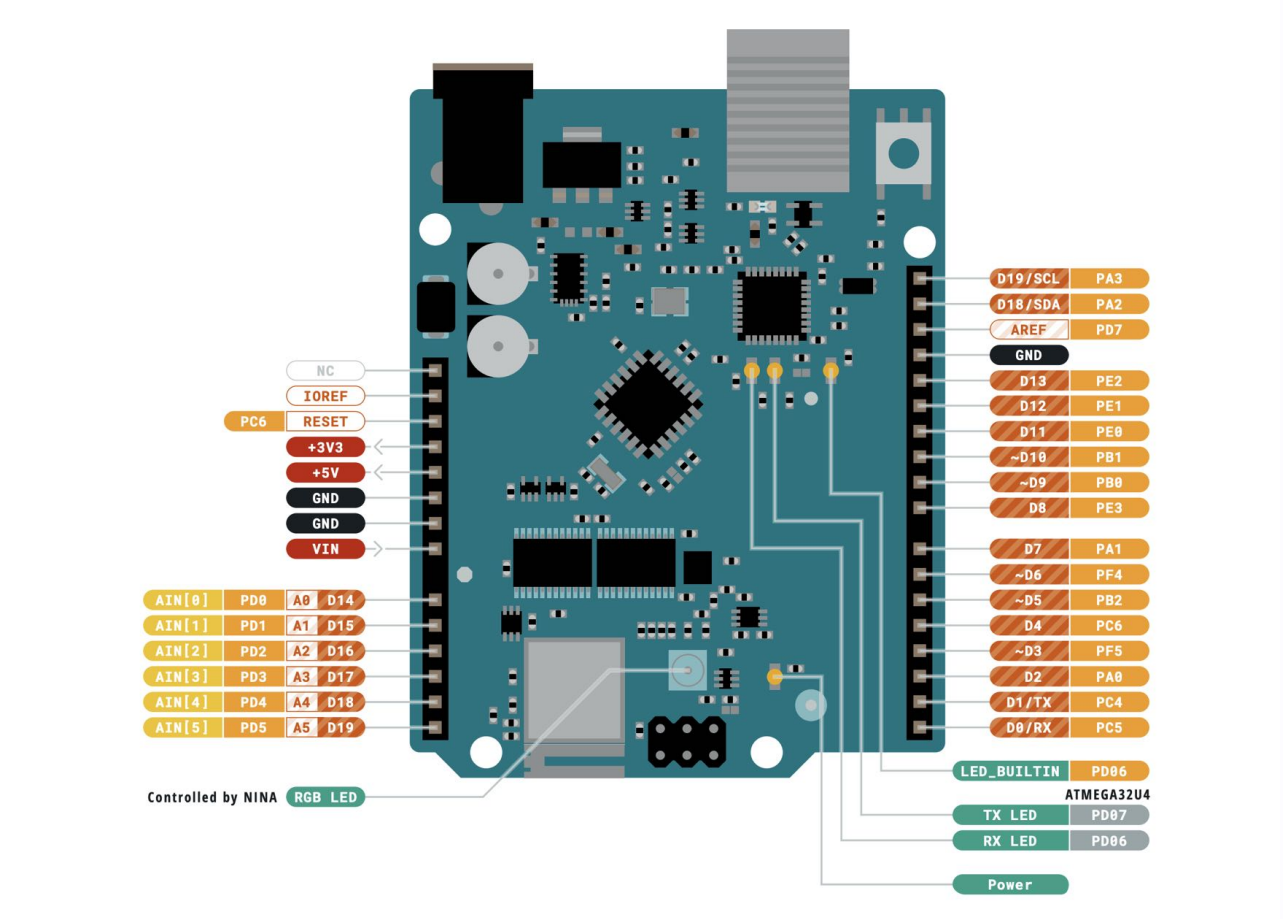
Processing

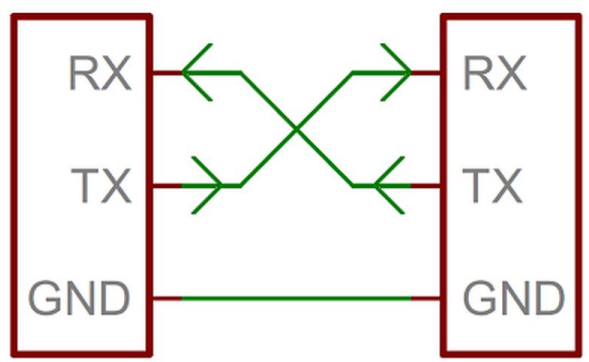
MicroController

Universal Asynchronous Receiver Transmitter

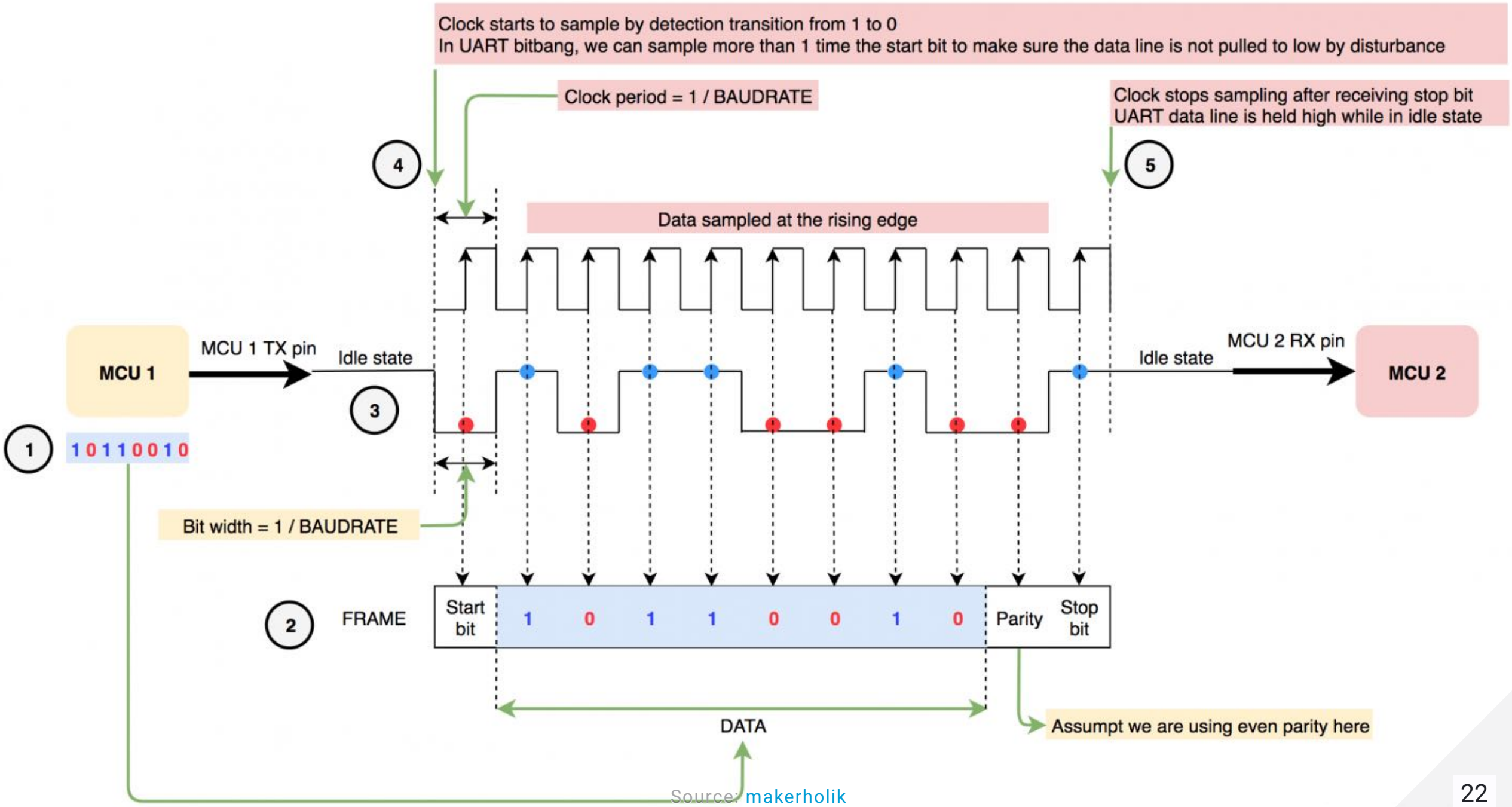
- UART (Chip)
- Baudrate (`9600`)
- Serial (RS-232)
- Port Based (tty / cu)
- `ls -la /dev/tty.*`
  - Device Manager `COM1`


Tools
Help
Auto Format ⌘T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... ⇧⌘I
Serial Monitor ⇧⌘M
Serial Plotter ⇧⌘L
WiFi101 / WiFiNINA Firmware Updater
ESP Exception Decoder
Board: "Arduino/Genuino Uno"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
/dev/cu.BRBRX-SPPDev
/dev/cu.BRBRX-SPPDev-1
/dev/cu.BRBRX-SPPDev-2
/dev/cu.BRBRX-SPPDev-3
/dev/cu.BRBRX-SPPDev-4
/dev/cu.Bluetooth-Incoming-Port
/dev/cu.MALS
/dev/cu.SLAB_USBtoUART
/dev/cu.SOC
/dev/cu.UEMBFlorian-LWACP
/dev/cu.XOR-WirelessiAPv2
096/how-to-find-all-serial-devices-ttys
e | Arduino 1.8.10
, OUTPUT);
ED_LED, LOW);





Source: Ardunio

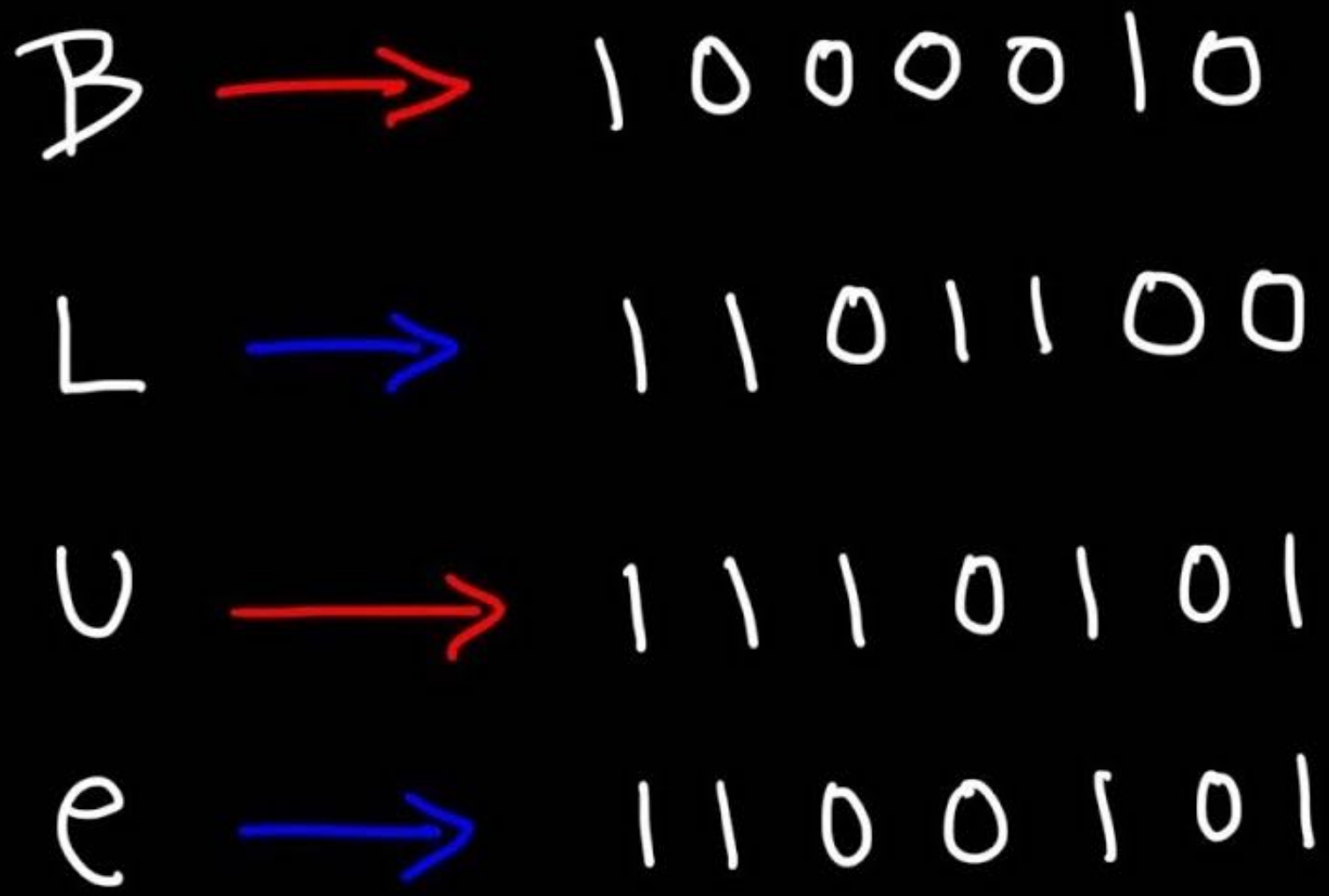


Source: makerholik

Text Econding

- How to send the text `Blue`?
- Convert into binary representation
- Use Ascii Table (UTF8 / UTF16)

Char.	ASCII	Char.	ASCII	Char.	ASCII
@	64	U	85	j	106
A	65	V	86	k	107
B	66	W	87	l	108
C	67	X	88	m	109
D	68	Y	89	n	110
E	69	Z	90	o	111
F	70	[	91	p	112
G	71	\	92	q	113
H	72	]	93	r	114
I	73	^	94	s	115
J	74	_	95	t	116
K	75	`	96	u	117
L	76	a	97	v	118
M	77	b	98	w	119
N	78	c	99	x	120
O	79	d	100	y	121
P	80	e	101	z	122
Q	81	f	102	{	123
R	82	g	103	|	124
S	83	h	104	}	125
T	84	i	105	~	126

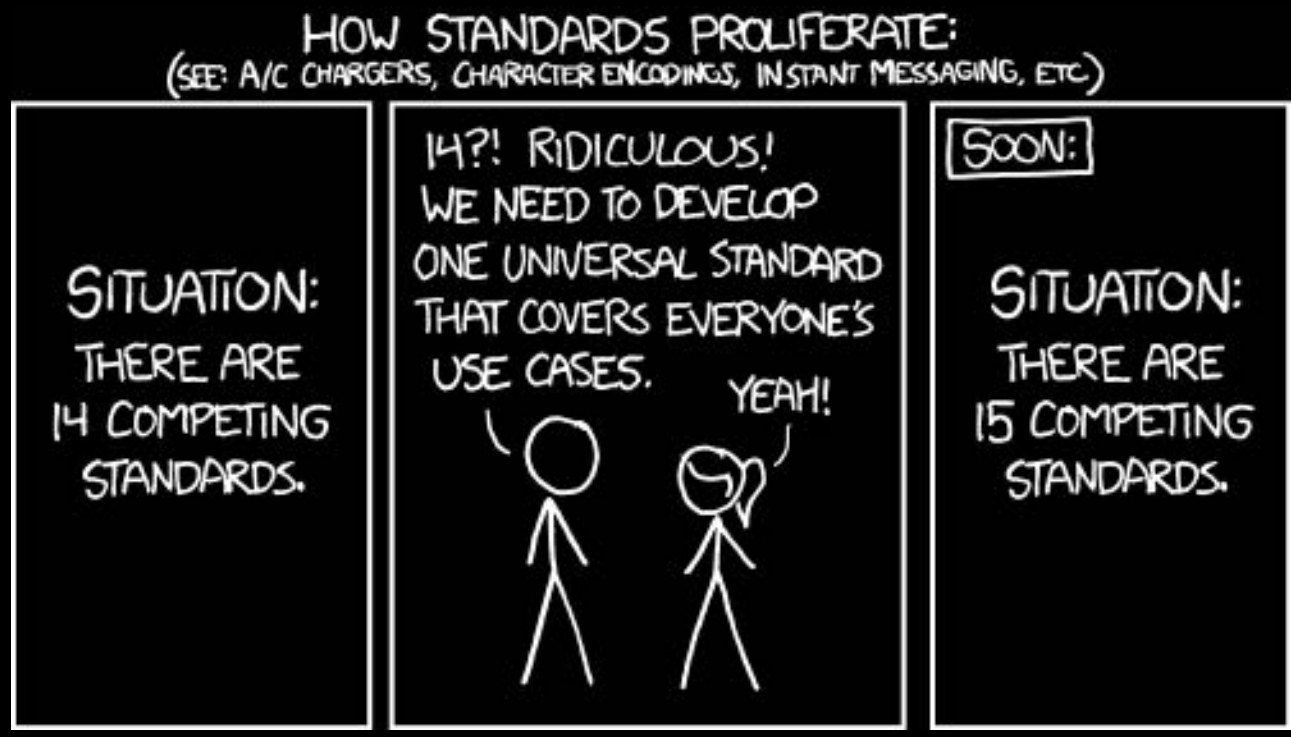


Source: The Organic Chemistry Tutor

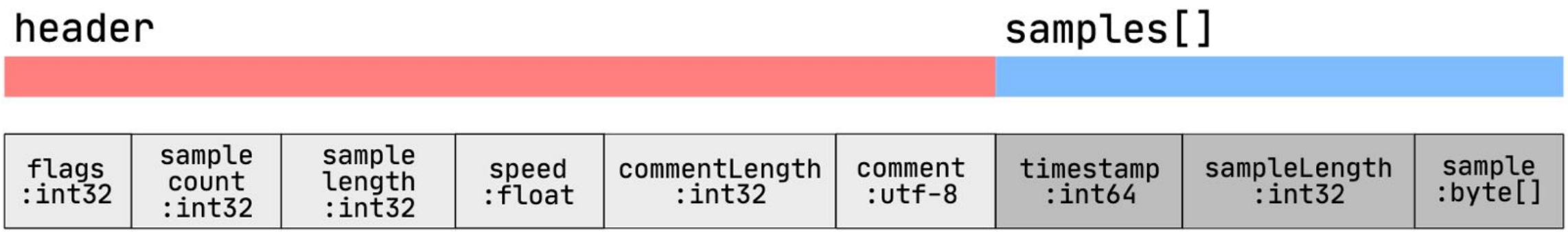
Protocol

```
hello world
```

- Delimiter
  - Command
  - Parameters
- Keywords



time resolution is currently limited to microseconds due to modern operating systems realtime limitations, but the timestamp could be also interpreted as nanosecond.



The data is in binary format with little-endian encoding. Please be aware that the header can expand and contain more fields, which currently are not defined.

Header

Only one header per file is allowed and every seqosc file has to start with the header.

- flags (`int32`)
  - compression (`bit 0`) - if on payload is compressed with delfate
- sample-count (`int32`) - how many samples are in the payload (`-1` tells the parser to iterate himself)
- sample-length (`int32`) - how long the payload is in bytes (mainly used for compression)

Source: seqosc

Datastructure Example

```
// sensor model
int time = 1000;
float temperature = 20.5;
int state = HIGH;
```

- `sensor:1000,20.5,1\n`
  - Keyword: `sensor`
  - Delimiter: `:`, `,`, `\n`
- Serialisation

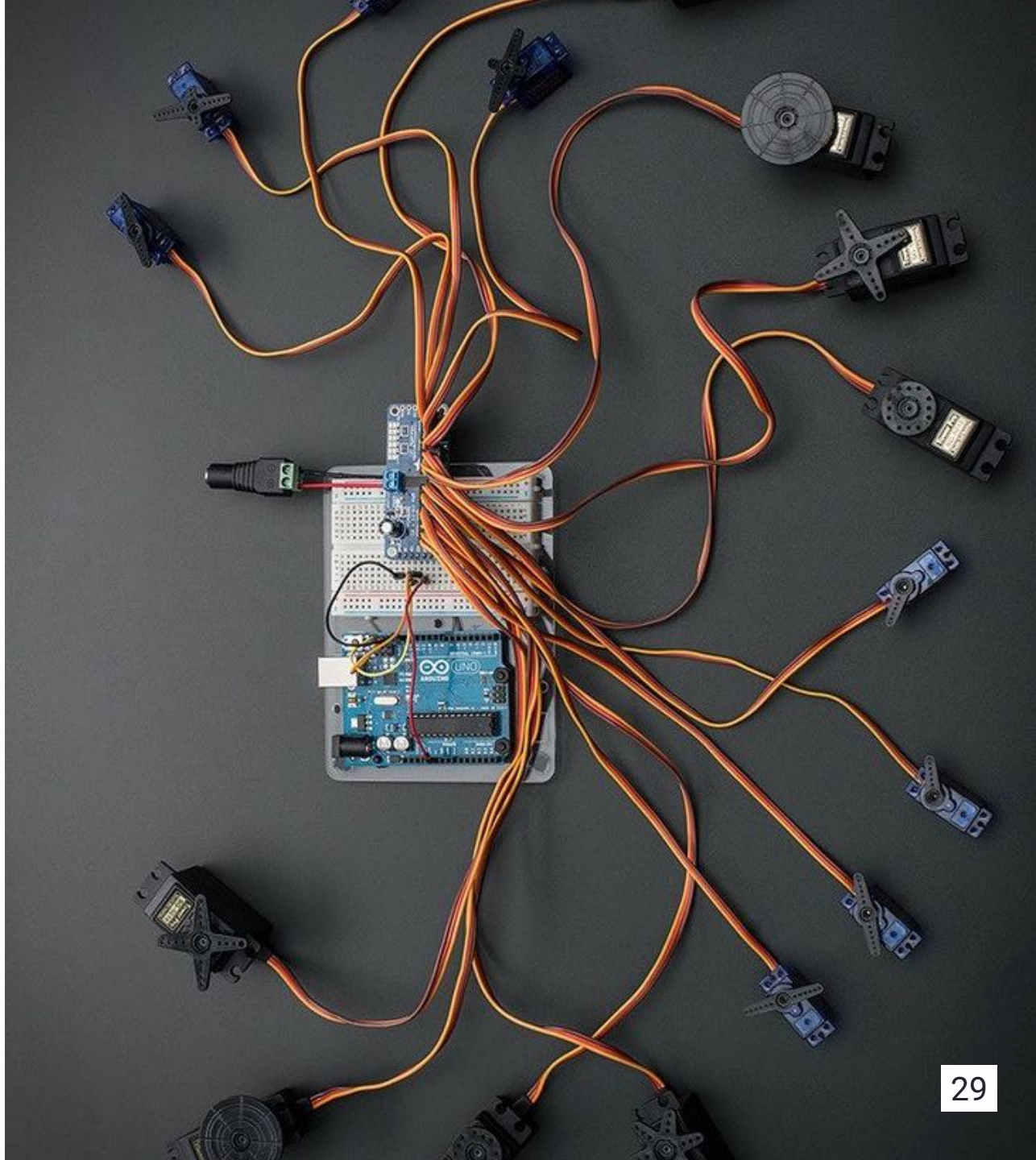
👩‍💻 Task 1: Servo Controllers (15min)

Control a **continous servo** with a **potentiometer** and send the servo state over the serial port.

- What is a servo?
- Protocol
- `Serial.begin(9600);`, `Serial.print("\n");`
- Increase zero range

Servo

- Accurate motion (degrees)
- Continous Servo (360°)
- `0-89` speed left side
- `90` no motion
- `91-180` speed right side



Servo Code

```
#include <Servo.h>

Servo servo;

void setup() {
  servo.attach(9);
}

void loop() {
  servo.write(45);
  delay(15);
}
```

Serial Library Processing

- Processing Readme
- `Serial.list()` - List all serial ports
- `port.readStringUntil('\n')` - Read until character
- Quick Demo

String Parsing

```
String line  = port.readStringUntil('\n');

// check if buffer is empty
if (line == null)
  return;

// remove spaces
line = line.trim();

// split by comma
String[] elements = line.split(",");
```

Datatype Conversion

```
int x = int("1234");
float f = float("0.23");
```

👨‍💻 Task 2: Parse Data (10min)

Read the **servo data** inside of a Processing sketch from the serial port and **store it** into a variable.

👩‍💻 Task 3: Virtual Servo (10min)

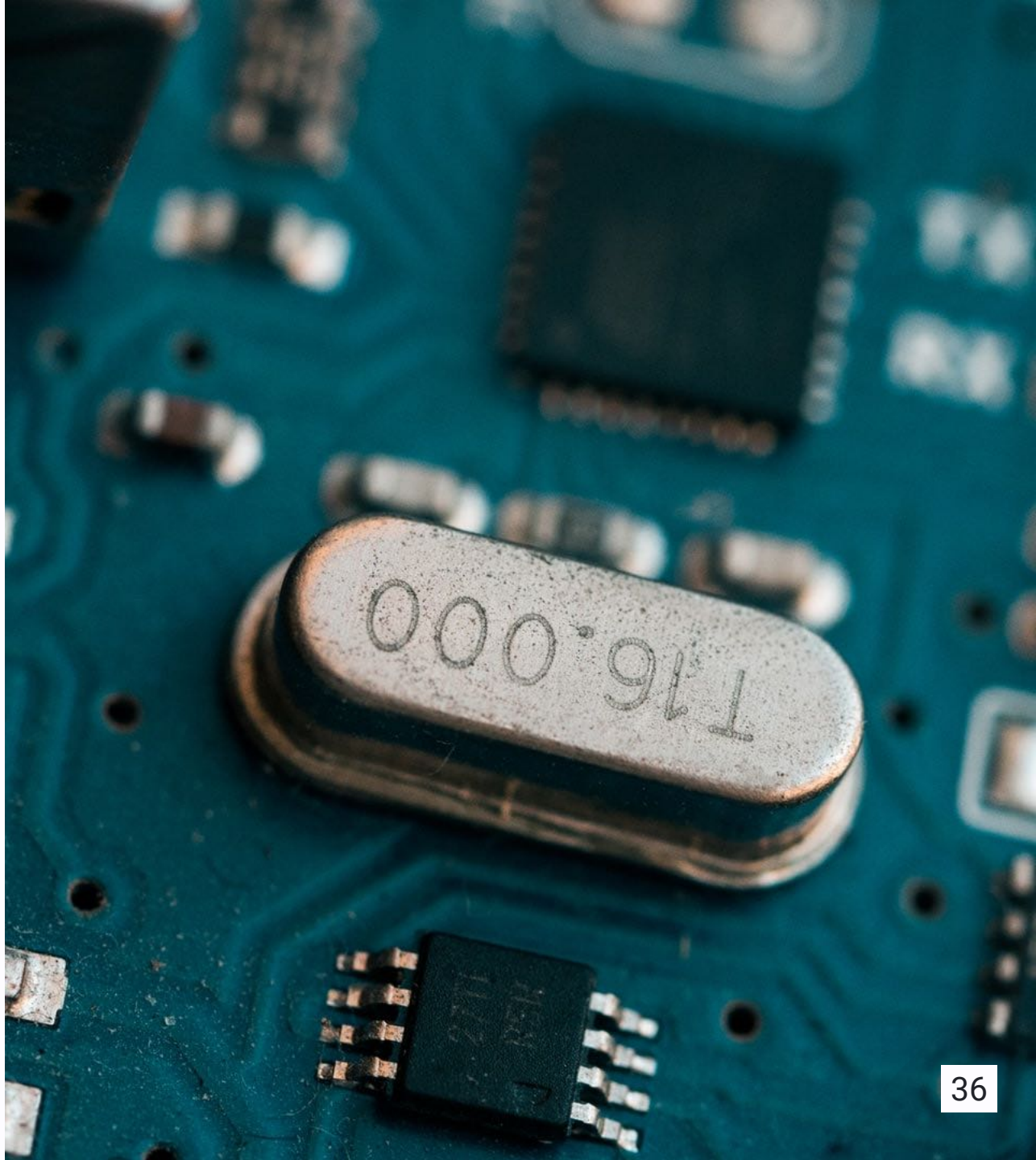
Visualize the **servo state** in Processing.

Control the Arduino

- Two-way communciation (bidirectional)
- Question / Answer
  - Who is the ~~master~~ / ~~slave~~?
  - primary / replica?
  - Timeouts / Deadlocks...?

  🙈

Source: Harrison Broadbent



```
// reading
auto line = serial.readStringUntil("\n");
line.trim();

// parsing
auto command = line.substring(0, line.indexOf(","));

// scanning
int interval;
int window;
int time;
char serviceAddress[37]; // 36 plus 1 for null

if (sscanf(line.c_str(), "%d %d %d %s",
  &interval, &window, &time, serviceAddress) < 0) {
    return -1;
}
```

How to create a GUI?

- Processing darwing
- Library
  - ControlP5


LED Forest 2 - Rod Edit
ADD
REMOVE
RODLIST
SAVE
EXIT
Rod 24
-58.666668
0.0
30.85714
24
NAME
X-AXIS
Y-AXIS
Z-AXIS
LED COUNT
INVERTED
Rod 24
Rod 1
Rod 1
Rod 1
Rod 1
Rod 1
Rod 1
Rod 1
Rod 1
Rod 1
Rod 1
Rod 1
Rod 1
Rod 24
Rod 24
Rod 1

Slider Example

```
import controlP5.*;

ControlP5 cp5;

int brightnessValue = 100;

void setup() {
  size(500, 500);

  cp5 = new ControlP5(this);

  cp5.addSlider("brightnessValue")
     .setRange(100,255)
     .setValue(120)
     .setPosition(100,200)
     .setSize(100,10)
     ;
}
```

Event or Plugin

Using the same name as the variable plugs into that variable.

Event based

```
void brightnessValue(int brightness) {
  println("Brightness: " + brightness);
}
```

👨‍💻 Task 3: Control (20min)

Create a very simple UI to control the servo's **max speed**.
Think about other things you can control (LED..).

Questions?