

KOMMUNIKATIONSPROTOKOLLE

How to make things talk to each other

HID Keyboard, Mouse & Co.

- + An jedem Rechner verfügbar
- + In vielen Frameworks und Prototyping Programmen nutzbar
- + Intuitiv
- Keyboard: nur digitale buttons
- Mouse: schwer für Prototyping modifizierbar/nutzbar



HID Keyboard, Mouse & Co.

Mögliche Anwendungslösungen:

Input:

Physischer-Keyboard

Arduino Keyboard Emulation

MakeyMakey

LeapMotion Touchless (win)

QR-Code Reader / RFID Reader

...

Output:

Figma

ProtoPie

Adobe XD

Javascript

Windows Touch

...

Serial USB Devices, Arduino

Bitweiser Versand von binären Daten über serielle Ports der verbundenen Geräte.

- + “Standardsprache” für viele USB Devices und Computer
- + Schnelle und niederschwellige Kommunikation zu Mikrocontrollern (Arduino etc.)
- + Basis für viele Frameworks/Libraries (Firmata, Blokdots, Johnny Five JS)
- Bitweise auslesen kann dazu führen, dass Daten bei hohen Übertragungsraten verloren gehen
- Nur per Code oder 3. Software nutzbar.
- Tiefe Integration ins System führt immer wieder zu abstürzen wenn z.B. Port blockiert ist.

```

SerialCallResponse | Arduino 1.8.11
Überprüfen

SerialCallResponse

int firstSensor = 0; // first analog sensor
int secondSensor = 0; // second analog sensor
int thirdSensor = 0; // digital sensor
int inByte = 0; // incoming serial byte

void setup() {
  // start serial port at 9600 bps:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  pinMode(2, INPUT); // digital sensor is on digital pin 2
  establishContact(); // send a byte to establish contact until receiver responds
}

void loop() {
  // if we get a valid byte, read analog ins:
  if (Serial.available() > 0) {
    // get incoming byte:
    inByte = Serial.read();
    // read first analog input, divide by 4 to make the range 0-255:
    firstSensor = analogRead(A0) / 4;
    // delay 10ms to let the ADC recover:
    delay(10);
    // read second analog input, divide by 4 to make the range 0-255:
    secondSensor = analogRead(A1) / 4;
    // read switch, map it to 0 or 255L
    thirdSensor = map(digitalRead(2), 0, 1, 0, 255);
    // send sensor values:
    Serial.write(firstSensor);
    Serial.write(secondSensor);
    Serial.write(thirdSensor);
  }
}

void establishContact() {
  while (Serial.available() <= 0) {
    Serial.print('A'); // send a capital A
    delay(300);
  }
}

1 Arduino Duemilanove or Diecimila, ATmega328P auf /dev/cu.usbserial-A907TJ4I

```

OSC Open Sound Control

Nicht nur für Musik und Sound...

Kommunikationsprotokoll auf Basis von TCP/UDP

- + Viele Integrationen und Libraries verfügbar.
- + Einfach verständliches und potentes Protokoll für Netzwerk
- + Versch. Datentypen möglich
- + Komplexere Daten transportierbar

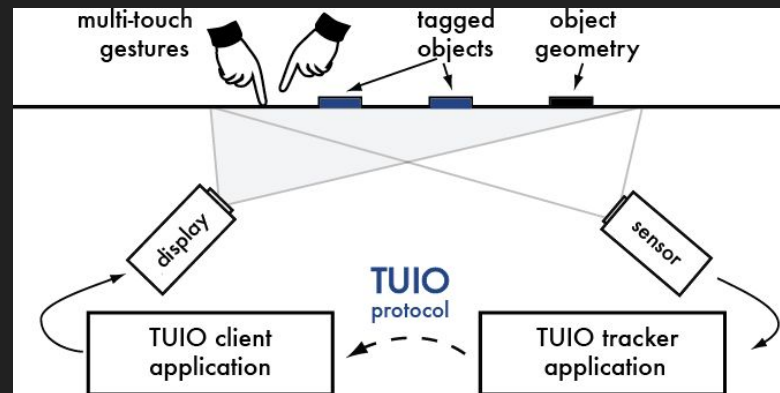
- In Anwendung eher für individuelle Programmierung nutzbar (Processing, Nodejs, MaxMSP, PureData ...)

Gute Test-App
<https://hexler.net/touchosc>

TUIO Touch und Tokenerkennung

Auf OSC aufsetzendes Protokoll für die Kommunikation von Touches und Token

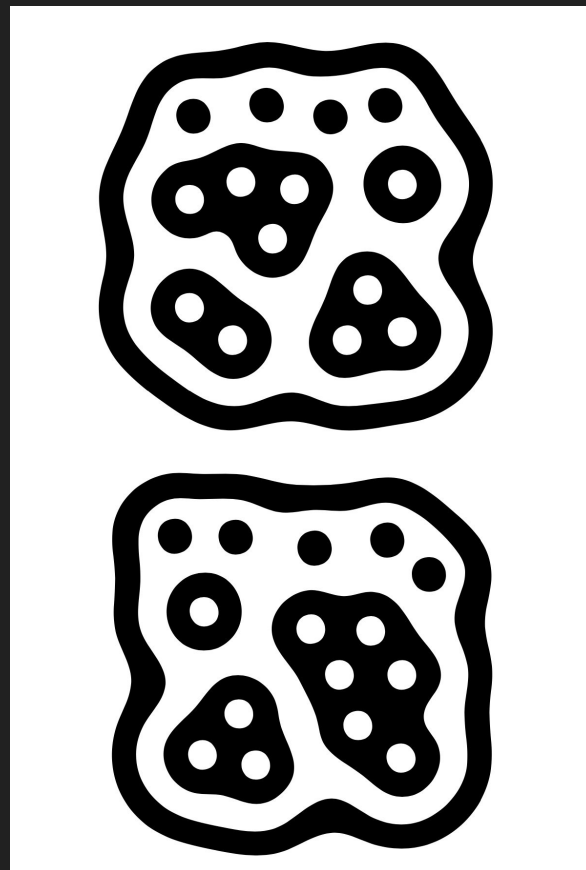
- + Ermöglicht Tracking von Touch und Token in fertige Software auszulagern.
- + Standardisiertes Handling von Interface Events
- + Viele verfügbare Integrationen
- In Anwendung eher für individuelle Programmierung nutzbar (Processing, Nodejs, MaxMSP, PureData ...)



TUIO Reactivision

Auf OSC aufsetzendes Protokoll für die Kommunikation von Touches und Token

- + Fertiges performantes Markertracking
- + Eingebautes Finger-Tracking
- + Sehr flexibel zum Prototyping nutzbar
- In Anwendung eher für individuelle Programmierung nutzbar (Processing, Nodejs, MaxMSP, PureData ...)



WebSocket Servers & Clients

Standardkommunikation für Web-Infrastruktur zwischen Server (Backend) und Client (Browser)

- + Standardisiert und in alle gängigen Browser eingebaut
- + Gezielte Kommunikation zu einzelnen Clients und/oder Broadcasting an alle
- + Lokal oder über's Internet nutzbar
- + Teilweise auch Integrationen in Software (ProtopieConnect,...)
- Hauptsächlich in Javascript Anwendungen nutzbar.



Gutes Framework:
<https://socket.io/>