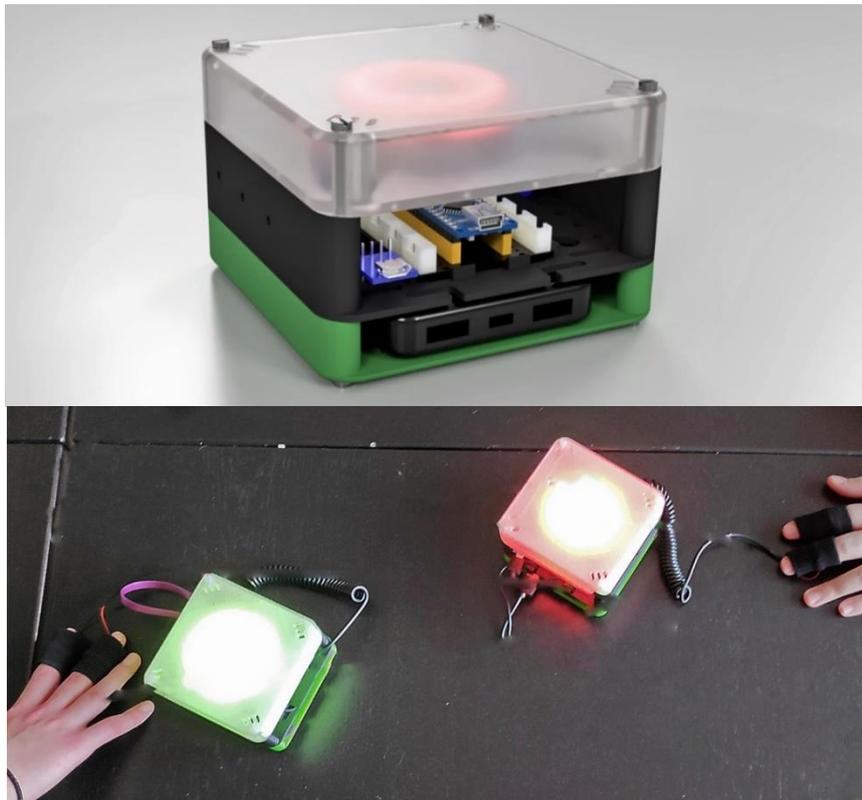


## **'We are the makers - IoT' Learn-Scenario: EDA-Cube: eine Vorstellung erhalten, wie sich Ihr Partner fühlt"**

Autor: Thomas Jörg, Johannes-Kepler-Gymnasium Weil der Stadt

*Das folgende Papier wurde als Abkömmling der Biofeedback-Station (IO2) entwickelt. Da die Biofeedback-Station sehr umfangreich zum Zwecke des Technologie-Unterrichts aufgebaut wurde, kann der EDA-Würfel hingegen für nicht-technische Zwecke genutzt werden, wie zum Beispiel als Werkzeug zur Überwachung von Dialogen: Er zeigt Emotionen seines Nutzers, soweit es die EDA-Technologie zulässt.*



*Abbildung 1: EDA-cubes in Aktion*

Was ist, wenn zwei Leute miteinander reden und höflich zueinander sein wollen. Ein großes Problem ist es immer, dass man nicht weiß, wie sich der Gesprächspartner fühlt. Ist er wütend, verängstigt, nervös, glücklich, müde? Welche Auswirkungen haben meine Worte? Ist man – in den Augen des Partners – unhöflich oder nachdenklich? Gibt es eine unmittelbare Antwort darauf, oder zumindest einen "Hinweis"?

Und: Was ist, wenn Sie eine Smartwatch tragen, die auch in der Lage ist, Ihre emotionalen Reaktionen zu messen? Möchten Sie, dass Ihre Gefühle von einer Maschine überwacht werden? Seien Sie sich der Fähigkeiten moderner Sensorik bewusst! Lassen Sie uns ein intelligentes Gerät mit einem 'Emotions'-Sensor bauen, der Ihre Körperreaktionen interpretiert!

<b>1. Title of Scenario</b>	<b>Wie man Emotionen mit Hilfe biosensorischer Messung überwacht</b>
<b>2. Zielgruppe</b>	12- 17 Jahre
<b>3. Dauer</b>	Mindestens 3 Wochen 2*45min-Unterricht pro Woche: in Summe ca. 6-8 Stunden.
<b>4. Lernbedürfnisse, die durch die Übung abgedeckt werden</b>	<p>Interaktion zwischen elektronischen Teilen und menschlichen Körpern</p> <p>Überwachung und Beeinflussung menschlicher biologischer Parameter</p> <p>Kommunikationskette von IoT-Geräten</p> <p>Grundsätze von Sensoren und Akteuren</p> <p>Was ist EDA?</p> <p>Grundsätze drahtloser Kommunikationsnetze</p> <p>Konstruktion und 3D-Druck von Messhilfen.</p>
<b>5. Erwartete Lernergebnisse</b>	<p>Wie funktioniert ein intelligentes IoT-System?</p> <p>Wo sind die Möglichkeiten und Bedrohungen von EDA-Systemen?</p> <p>Welche Komponenten – Hard&amp;Software – sind wichtig zum Bau eines IoT-Geräts?</p> <p>Bewusstsein für Biosensorik und die Fähigkeit, diese Werkzeuge zu beurteilen.</p>
<b>6. Methoden</b>	In diesem Szenario konstruieren, bauen und programmieren die Kursteilnehmer selbst ein interaktives EDA-Gerät, das die Messwerte interpretiert und visualisiert. Die Schüler können die Wi-Fi-Technologie verwenden, um die Werte zur weiteren Verarbeitung an einen Computer weiterzugeben.
<b>7. Ort/Umgebung</b>	<p>ein Labor mit einem Satz elektronischer Teile und Komponenten.</p> <p>Jede Gruppe von Kursteilnehmern benötigt einen Computer oder Laptop mit Administratorrechten für die Installation verschiedener Softwarepakete</p> <p>Ein Projektor für den Unterricht von Tutorials und die Präsentation von Studentearbeiten;</p>

## 8. Werkzeuge/ Materialien/ Ressourcen

### 3D-Drucker

Etwa 2-3 3D-Drucker sind notwendig, da die EDA-Kuben gedruckt werden müssen. Natürlich ist es den Schülern möglich, Maschinenteile selbst zu konstruieren

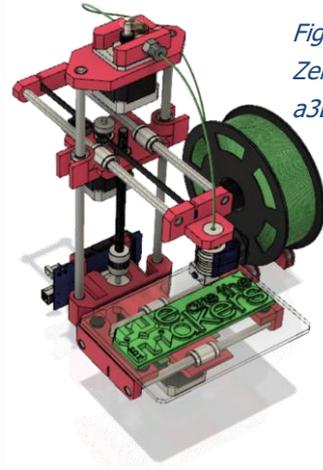


Figure 2:  
Zeichnung von  
a3D-Druckers

### 3D-gedruckte Komponenten:

Als Ausgangspunkt werden alle notwendigen Teile im stl-Format und als Autodesk Fusion 360-Dateien bereitgestellt.

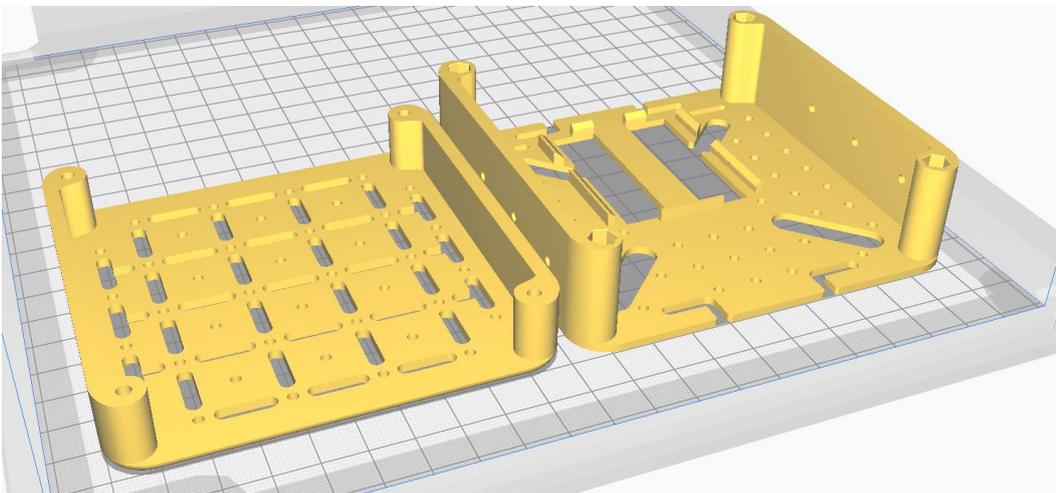


Abbildung 3: STL-Daten von regulären PLA-Teilen

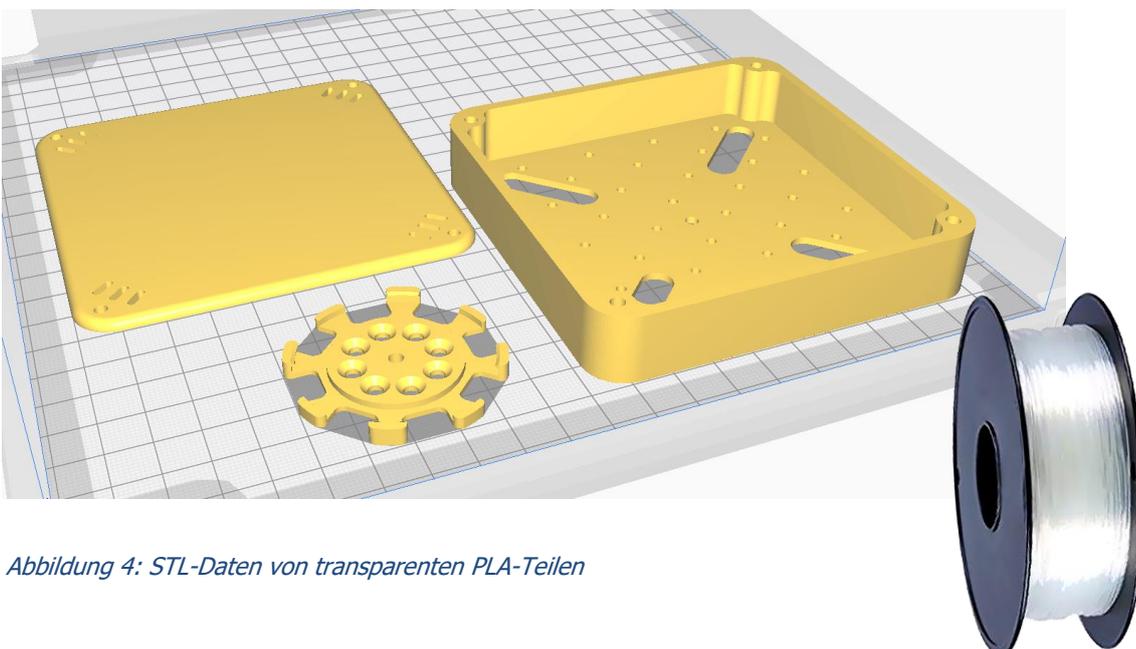


Abbildung 4: STL-Daten von transparenten PLA-Teilen

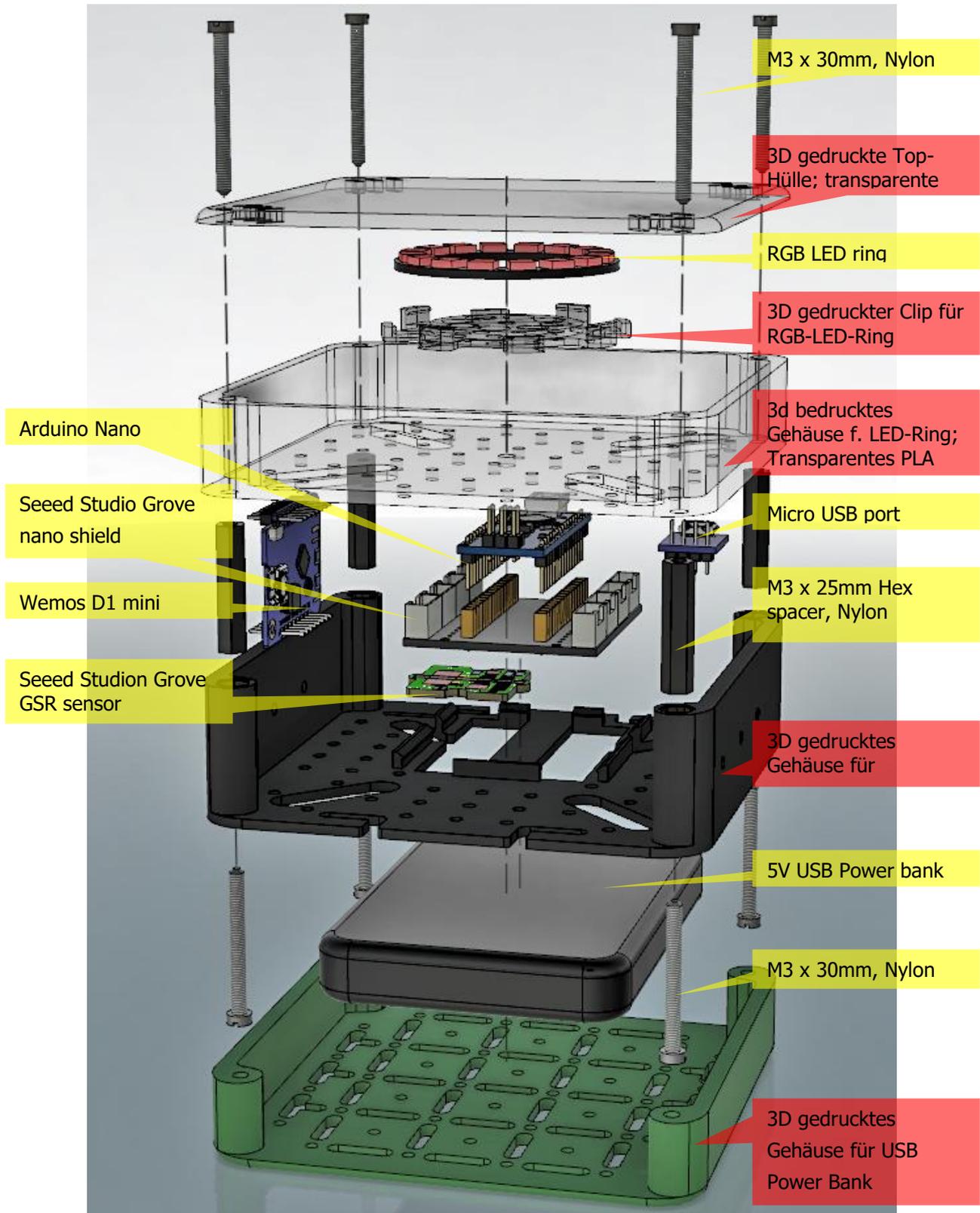


Abbildung 5: Explosionsbaugruppenzeichnung des Cubes

Electronic components:

***ACHTUNG: Da wir Experimente mit dem menschlichen Körper machen, muss jede erdenkliche Vorsichtsmaßnahme getroffen werden! Verbinden Sie niemals einen menschlichen Körper mit dem häuslichen Energiesystem. Der menschliche Körper muss immer komplett vom Stromnetz ferngehalten werden!***

Dazu gehören auch Netzadapter, die an die Steckdose angeschlossen sind. Diese Art von Schaltungen muss vermieden werden. Verwenden Sie nur Batterien und Akkumulatoren mit einer niedrigen Spannung von ca. 3-5V.

## **9. Setup components**

In dieser Arbeit empfehlen wir das Seeed Grove-System als Basis für seine Benutzerfreundlichkeit: ([http://wiki.seeedstudio.com/Grove\\_System/](http://wiki.seeedstudio.com/Grove_System/)):

Seeed Studio Components:

1x Grove Shield für Arduino Nano

<https://www.seeedstudio.com/Grove-Shield-for-Arduino-Nano-p-4112.html>

1x Grove GSR

[http://wiki.seeedstudio.com/Grove-GSR\\_Sensor/](http://wiki.seeedstudio.com/Grove-GSR_Sensor/)

4 x Seeed Studio Cables:

1x Grove - 4 pin Male Jumper to Grove 4 pin Conversion Cable

<https://www.seeedstudio.com/Grove-4-pin-Male-Jumper-to-Grove-4-pin-Conversion-Cable-5-PCs-per-Pack.html>

2x Grove - Universal 4 Pin Buckled 5cm Cable

<https://www.seeedstudio.com/Grove-Universal-4-Pin-Buckled-5cm-Cable-5-PCs-Pack.html>

1x Grove - Universal 4 Pin 20cm Unbuckled Cable

<https://www.seeedstudio.com/Grove-Universal-4-Pin-20cm-Unbuckled-Cable-5-PCs-Pack-p-749.html>

#### Microcontroller:

1x Arduino Nano (or equivalent)

<https://store.arduino.cc/arduino-nano>

1x Wemos LOLIN D1 mini (or equivalent)

[https://wiki.wemos.cc/products:d1:d1\\_mini](https://wiki.wemos.cc/products:d1:d1_mini)

#### Elektronik Teile:

1x Adafruit RGB-LED Ring

<https://www.adafruit.com/product/1463>

1x Micro USB to DIP Adapter

<https://www.google.com/search?q=Micro+USB+to+DIP+Adapter&og=Micro+USB+to+DIP+Adapter&aqs=chrome..69i57j3497j0j7&sourceid=chrome&ie=UTF-8>

#### Sonstige Teile:

- 4x M3 Nylon Standoffs 25mm (Hex spacer)
- 8x M3 Nylon Screws 30mm
- M2 Nylon Standoffs (Hex spacer) für Grove (has 2mm holes)
- Kleine USB Power Bank, max. Größe 15mm x 100mm x 60mm
- Kleines micro USB Kabel zum Verbinden der Power bank mit dem Nano
- Ein Lötkolben zum Anbringen von Kabeln an elektronischen Bauteilen



#### Computer mit der folgenden Software vorinstalliert:

- Autodesk Fusion 360 (oder jede andere 3D-Modellierungssoftware, z.B. Wings3D)
- CURA Slicer,
- Eine Internetverbindung zum Herunterladen von Bibliotheken
- Arduino IDE

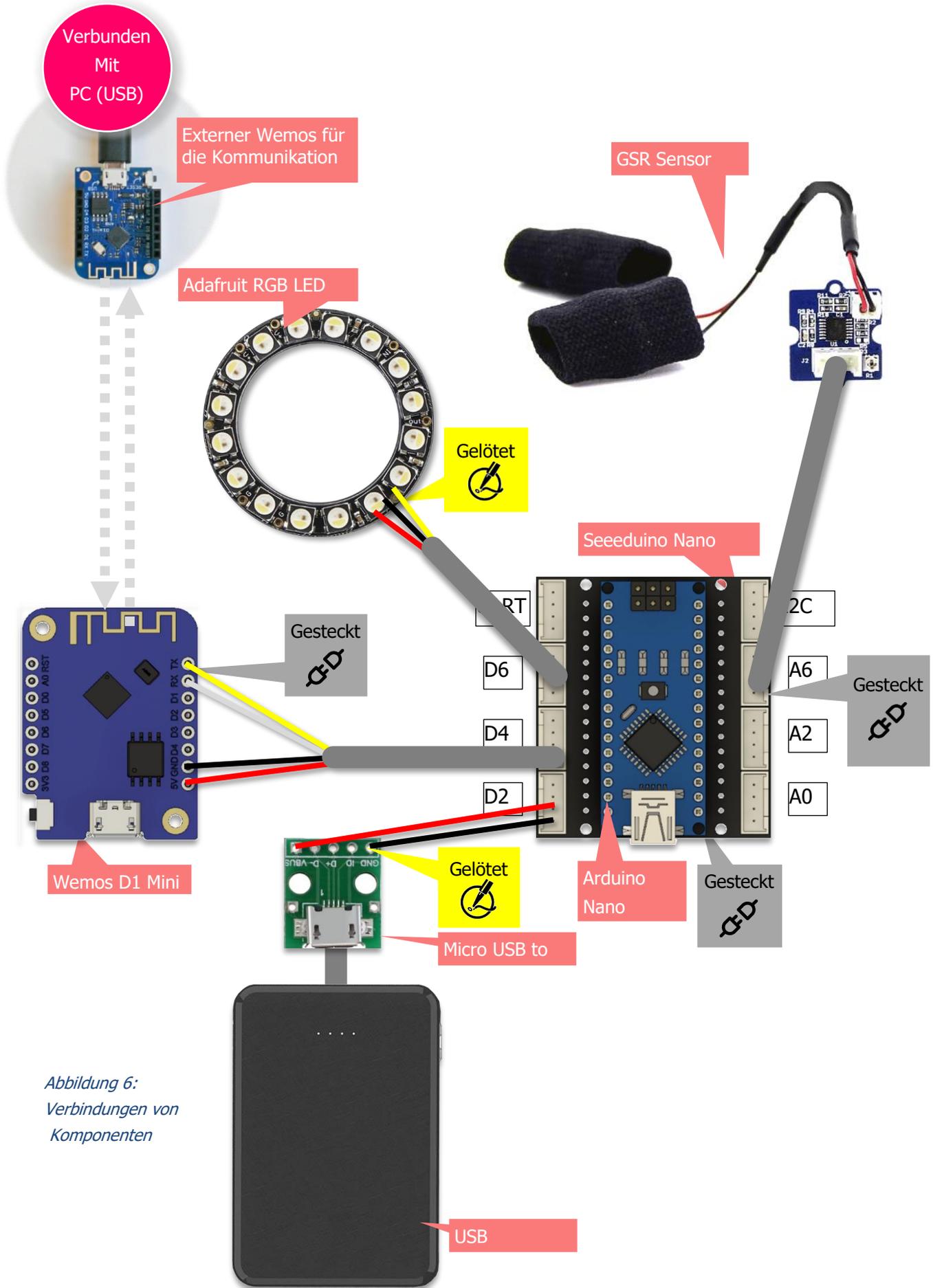


Abbildung 6:  
Verbindungen von  
Komponenten

### Arduino-Bibliotheken für Komponenten:

Die Wemos D1 Mini benötigen S-Bibliotheken, damit die Arduino IDE richtig funktioniert. Wie eine Bibliothek importiert wird hier beschrieben: <https://www.arduino.cc/en/Guide/Libraries>

Neopixel (Adafruit):

[https://github.com/adafruit/Adafruit\\_NeoPixel/archive/master.zip](https://github.com/adafruit/Adafruit_NeoPixel/archive/master.zip)

### **Präferenz-URL für WEMOS-Boards (ESP8266):**

Um das Wemos zu installieren, muss die sogenannte "Board-Definition" installiert werden. Es wird hier beschrieben:

<http://arduino.esp8266.com/Arduino/versions/2.0.0/doc/installing.html>

1. Im Fenster Arduino IDE öffnen der Voreinstellungen.
2. Geben Sie die folgende URL in das Feld "Zusätzlicher Board-Manager" ein:  
[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)
3. Öffnen Sie „Boards Manage“r aus Tools > Board Menü und finden Sie esp8266 Plattform.
4. Wählen Sie die aktuelle Version aus einem Dropdown-Feld aus und klicken Sie auf die Schaltfläche "Installieren".
5. Wählen Sie "(LOLIN) Wemos D1 R2 und Mini" aus Tools > Board Menü nach der Installation.

Der Grove GSR-Sensor benötigt keine Bibliotheken, da er mit einfachen analogen Arduino-Eingangsbefehlen gesteuert werden kann.

## Wemos D1 mini als drahtlose Verbindung zwischen elektronischen Komponenten

- **Wemos-Boards sollten vom Lehrer vorbereitet werden, nicht von den Schülern, bevor der Unterricht beginnt!**
- Wemos-ESP8266-Wifi-Boards sind als kostengünstigere Alternative zur zuverlässigen, aber auch teuren Xbee-Technologie gedacht.

Zwei Wemos bilden ein Paar, das über Wi-Fi-Ethernet-Port 23 (telnet) verbunden ist. Der einzige Zweck besteht darin, **das serielle Kommunikationskabel** zu ersetzen. Üblicherweise wird ein experimentelles elektronisches Gerät über ein USB-Kabel mit dem PC verbunden. Um ein völlig autonomes Design zu erreichen, das nicht mit dem häuslichen Stromnetz verbunden ist, muss eine drahtlose Verbindung hergestellt werden.

Daher wird die übliche serielle Kommunikation (UART) in Wifi übersetzt und von einem Wemos, von den anderen Wemos empfangen und wieder in serielle Kommunikation übersetzt. Aus Kompatibilitätsgründen ist die Baudrate auf 9600 Baud festgelegt, da die Software-Serial-Communication durch einen Arduino Uno auf 9600 Baud beschränkt ist.

Ein Wemos D1 Mini-Paar besteht aus einem Server und einem Client. Der Server sollte mit dem PC verbunden sein. Es sollte zuerst gestartet werden und führt die folgenden Schritte aus.:

1. Scannen aller verfügbaren WLAN-Netzwerke,
2. Bestimmen, ob sich ein ungenutzter, freier Kanal oder ein schwaches Netzwerk im Hintergrund befindet,
3. Einrichtung eines Wifi Access Points über den ersten freien Kanal, auch in Kombination mit DHCP
4. Warten auf EINEN (nur einer!) Client, der eine Verbindung herstellt.
5. Wenn die Verbindung des Clients getrennt wird, wartet der Server, bis der Client erneut eine Verbindung herstellt.
6. Wenn Server zurückgesetzt wird, beginnen Sie einfach bei 1. (Scan-Netzwerke)

Der Client sollte als zweiter gestartet werden und automatisch eine Verbindung herstellen und die Verbindung wieder herstellen.

## Konfigurieren des Wemos-Servers und -Clients, erläutert unter "Verbesserter Server-Quellcode":

Hier sind die relevanten Auszüge aus Server- und Client-Quellcode, die für die Konfiguration einzelner Wemos-Boardspaare angepasst werden müssen:

```
#include <ESP8266WiFi.h>

const char *ssid = "Erasmus";
const char *password = "12345678";

IPAddress Ip(192, 168, 3, 1);
IPAddress NMask(255, 255, 255, 0);

WiFiServer server(23);
WiFiClient serverClient;
char inChar;
```

Abbildung 7 Ausschnitt des Server-Quellcodes

```
#include <ESP8266WiFi.h>

const char* ssid = "Erasmus";
const char* password = "12345678";

IPAddress server(192, 168, 3, 1);

WiFiClient client;
char inChar;
```

Abbildung 8 Ausschnitt des Client-Quellcodes

- Beide unterstrichenen Codezeilen müssen für ein Wemos-Paar genau gleich sein.
- Beide unterstrichenen Codezeilen müssen für jedes Wemos-Paar angepasst werden.

Ändern Sie die **IP-Adresse** in

192.168.1.1 OR 192.168.2.1 OR 192.168.4.1 OR 192.168.5.1 ...etc.

Ändern Sie die **ssid** in

"Erasmus1" OR "Erasmus2" OR "Erasmus4" OR "Erasmus5" ...etc.

... Kompilieren Sie die Skripte innerhalb der Arduino IDE und laden Sie sie auf die entsprechenden Wemos-Boards hoch.

## 8c Einige Theorie des Biofeedbacks

**Dieser Text ist als kurzer Überblick gedacht und kann als eine Sammlung wichtiger Schlüsselwörter betrachtet werden. Es ist nicht als Lehrbuch gedacht!**

<https://en.wikipedia.org/wiki/Biofeedback>

Die Reaktion eines menschlichen Körpers auf Stress oder äußere Einflüsse geschieht meist automatisch und unbewusst. Zum Beispiel, wenn ein Mensch lügt oder in Angst ist, beginnt seine Haut zu schwitzen. Dieser Schweiß kann als Änderung der elektrischen Leitfähigkeit gemessen werden, da Schweiß Elektrolyte enthält. Wenn der Messcomputer diese Veränderung visualisiert, kann der Mensch seinen emotionalen Zustand mit dem gemessenen Signal korrelieren und versuchen, seine Reaktion zu beeinflussen und zu lernen, seine Emotionen zu kontrollieren. Die vorhergehenden verborgenen Emotionen wurden nun dieser Person bewusst.

Es gibt viele Beispiele & Experimente, die Schüler selbst ausprobieren können:

1. Beeinflussen Sie die Herzfrequenz mit Änderung der Atemfrequenz, überwacht durch Pulssensoren
2. Beeinflussen Sie Angstreaktionen mit einer Änderung der Elektroderaktivität, die von GSR-Sensoren überwacht wird (Eine Angstreaktion könnte ein Bild einer Spinne sein, oder ein YouTube-Video einer Achterbahnfahrt)
3. Ein Polygraph (Lügendetektor) basiert unter anderem auf der Veränderung der Elektroderaktivität und kann mit GSR-Sensoren gemessen werden
4. Koaktivität der Muskeln: Computertypisierung unter Stressbedingungen führt zu einer Kontraktion des Trapezmuskels im Nacken des Menschen. Dies kann mit EMG gemessen werden.

## 9. Unterrichtsplan: Schritt für Schritt Beschreibung der Aktivität/Desinhalte

### Lektion 1 & 2 (90min):



Die Schüler werden am Beispiel des IoT vorgestellt: Vakuumbotter mit App-Fernbedienungen, internetbasierte Wetterstationen, intelligente Landwirtschaft und schließlich Gesundheitsanwendungen. Die Kursteilnehmer sollten untersuchen, wie diese Geräte funktionieren und welche Komponenten benötigt werden: ein Mikrocontroller-basiertes System steuert und koordiniert angeschlossene Sensoren und Akteure. Darüber hinaus kommuniziert und

koordiniert es mit anderen Systemen ähnlicher Art oft über drahtlose Kommunikationsnetze.

Benötigte Teile: Sensoren, Schauspieler, Kommunikationsgeräte. Möglichkeiten, Bedrohungen und Einschränkungen müssen diskutiert werden: Wo macht IoT Sinn und wo nicht?

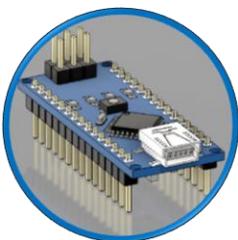
### Lektion 3 & 4 (90min):



3D-Druck und Montage des Gerätes: Die Schüler können das Gehäuse des Geräts 3D drucken, und danach sollten sie alle elektronischen Teile selbst anschließen. Die Schüler sollten ein tieferes Verständnis entwickeln, wie Teile zusammenpassen und somit ein komplettes Gerät bauen. Was ist der Zweck eines LED-Rings, wozu ist die WLAN-Verbindung geeignet? Wie wird ein Signal von seinem Ursprung bis zum Ende verarbeitet – zum Nutzer? Ausgehend von

einem Bio-Signal, das in ein analoges Signal im Inneren des Geräts umgewandelt wird, in ein digitales Signal innerhalb des ADC des Mikrocontrollers, Datenverarbeitung über die Software und Kommunikation der Ergebnisse über LED-Licht oder Übertragung von Informationen über Wi-Fi.

### Lektion 5 & 6 (90min):



Einführung von Arduino Programmierung: Arduino-DIE-Verbindung und

Kommunikations-Setup von Arduino Nano mit der IDE / dem Computer. Die grundlegende Struktur der Arduino-Plattform muss erklärt werden: Was sind Allzweck-Ein-/Ausgabe-Pins (GPIO), was digitale Logik ist und was der

Unterschied zwischen digitalem Input und Output ist. Einfache Skripte werden

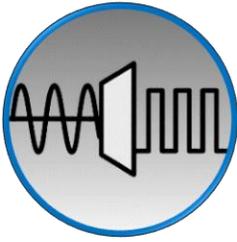
mit den grundlegenden Beispielen geändert, die mit der Arduino IDE ausgeliefert werden:

"01.Basics → Blink",

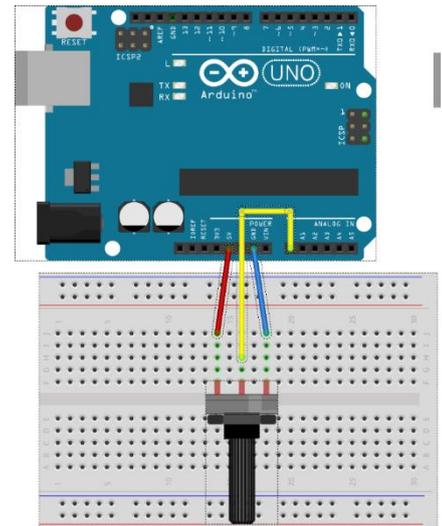
"01.Basics → DigitalReadSerial",

"04.Communications → SerialEvent",

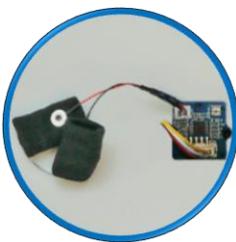
**Lektion 7 & 8 (90min):**



How is an analog signal processed by a digital machine? Theory of analog-to-digital-signal-conversion is introduced and can be taught using a simple voltage divider setup. Using the Arduino, any simple resistor-based sensor can be used to build up a simple circuit, e.g. an LDR, an thermoresistor or even a simple potentiometer.



**Lektion 9 & 10 (90min):**

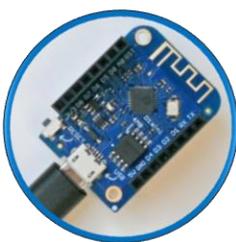


Vorstellung des GSR-Sensors: Erdbau der Signalverstärkung mit Operationsverstärkern zur Messung der Hautfeuchtigkeit mit niedrigen Spannungen. Wie kann der Widerstand der menschlichen Haut gemessen werden? Welche Spannung ist für den Anwender unbedenklich? Biologischer Hintergrund der elektrodermalen Aktivität und ihre Bedeutung/Interpretation

[https://en.wikipedia.org/wiki/Electrodermal\\_activity](https://en.wikipedia.org/wiki/Electrodermal_activity)

Die Programmierung des GSR-Sensors kann durch Die Verwendung und Anpassung der vorgefertigten Arduino-Skripte aus den vorhergehenden Lektionen erfolgen..

**Lektion 10 & 11 (90min):**



Einführung des Wemos D1 Mini: Diese Mikrocontrollerplatine kann als drahtloses Übertragungsgerät verwendet werden, das den IoT-Charakter des GSR-Cube ermöglicht. Zwei Wemos sind notwendig, um zwei verschiedene Knoten zu verbinden – ein Knoten ist der Cube selbst und der andere Knoten ist zum Beispiel der Computer eines Schülers, auf den alle verarbeiteten Signale übertragen werden.

**Da die Wlan-Technologie von Wemos sehr kompliziert ist und daher eine separate Unterrichtseinheit benötigt, müssen alle Wemos-Geräte vor dem Unterricht vorinstalliert und sorgfältig konfiguriert werden. Dies ist Aufgabe des Lehrers.**

Es ist wichtig, die Möglichkeit des Missbrauchs herauszuarbeiten: Es gibt viele Aspekte des Datenschutzes, die mit diesem Setup leicht veranschaulicht werden können. Welche Folgen hat die Übertragung eigener Signaldaten durch Körper, die zu einem Kontrollverlust der Privatsphäre führen??

## Lektion 13 & offenes Ende:



<https://www.youtube.com/watch?v=ZultgAFrxuc>

Diese Lektion basiert auf emotionalen Reaktionen beim Betrachten eines "Scary-Films": Das Auf und Ab einer Achterbahn kann einen großen Einfluss auf die Gefühle der Probanden haben. Wie kann man es beeinflussen?

Was ist mit einigen Bildern von Spinnen oder Schlangen? Oder von etwas Entzückendem / Angenehmem wie Musik? Wie wirkt sich Discomusik/Klassische Musik aus? Gibt es einen besonderen Effekt beim Hören Ihres Lieblingslieds?

## Und jetzt: Freestyle-Programmierung und Austesten des EDA-Cubes im Unterricht! 😊

Try to make some experiments with your classmates. Make a discussion round and keep track of the output of your EDA-cube. Can you recognize how your colleague feels?

### 10. Feedback

Am Ende der Lektion sollten die Schüler ein fundiertes Wissen darüber haben, wie IoT-Prinzipien in medizinischen Geräten funktionieren und wie der EDA-Kubus helfen kann, die verborgenen Eigenschaften unseres Körpers zu verstehen. Während des Unterrichts wurden wichtige Aspekte der Elektronik, der medizinischen Informatik und der Baugrundlagen vermittelt.

### 11. Assessment & Evaluation

Die Schüler führen ihr Arbeitstagebuch, das vom Lehrer überprüft werden kann. Die Schüler können auch die Ergebnisse ihrer Experimente präsentieren. Darüber hinaus muss am Ende des Unterrichts ein Standard-Test durchgeführt werden.

## Wemos Client Sourcecode

*Abbildung 9 Quellcode  
für den Wemos-Client,  
der an den EDA-Cube  
angeschlossen ist*



```
#include <ESP8266WiFi.h>
const char* ssid      = "Erasmus";
const char* password = "12345678";
IPAddress server(192, 168, 3, 1);
WiFiClient client;
char inChar;

void setup() {
  Serial.begin(9600);
  WiFi.setSleepMode(WIFI_NONE_SLEEP);
  WiFi.mode(WIFI_STA);
  WiFi.setOutputPower(10); // 10: 10mW, 14: 25mW, 17: 50mW, 20: 100mW
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {delay(5);}
  Serial.print("WiFi Channel: ");
  Serial.println(WiFi.channel());
  if (client.connect(server, 23)) {
    Serial.print("Local IP: ");
    Serial.println(WiFi.localIP());
    pinMode(LED_BUILTIN, OUTPUT);
    digitalWrite(LED_BUILTIN, LOW);
  }
}

void loop() {
  if (!client.connected()) {
    digitalWrite(LED_BUILTIN, HIGH);
    unsigned long startzeit = micros();
    client.connect(server, 23);
    Serial.println(micros() - startzeit);
  }else{
    digitalWrite(LED_BUILTIN, LOW);
  }
  if (client.available()) { //wenn WIFI verfügbar ist,
    char c = client.read(); //lesen was reinkommt
    Serial.print(c); //und auf UART schreiben.
  }
  while (Serial.available() > 0) { //solange auf UART Daten,...
    inChar = Serial.read(); //Daten von UART einlesen
    if (client.connected()) { //und wenn WIFI läuft, ...
      client.write(inChar); //auf WIFI schreiben
      delay(1);
    }
  }
}
```

## Wemos Server Sourcecode



*Abbildung 10 Dieses  
Skript sollte für den mit  
dem PC verbundenen  
Wemos kompiliert  
werden*

```
#include <ESP8266WiFi.h>
const char *ssid = "Erasmus";
const char *password = "12345678";
IPAddress Ip(192, 168, 3, 1);
IPAddress NMask(255, 255, 255, 0);
WiFiServer server(23);
WiFiClient sClient;
char inChar;

void setup() {
  Serial.begin(9600);
  unsigned int c_frei = SSID_scan();
  Serial.println("Configuring access point");
  WiFi.softAPConfig(Ip, Ip, NMask);
  WiFi.softAP(ssid, password, c_frei, false, 1);
  Serial.print("Channel: ");
  Serial.println(c_frei);
  Serial.println("Starting server");
  server.begin();
  server.setNoDelay(true);
  Serial.print("Server IP: ");
  Serial.println(WiFi.softAPIP());
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, HIGH);
}

void loop() {
  uint8_t i;
  if (server.hasClient()) {
    if (!sClient || !sClient.connected()) {
      if (sClient) sClient.stop();
      sClient = server.available();
      digitalWrite(LED_BUILTIN, LOW);
    }
  } else digitalWrite(LED_BUILTIN, HIGH);
  if (sClient.available()) {
    digitalWrite(LED_BUILTIN, LOW);
    while (sClient.available()) {
      inChar = sClient.read();
      Serial.write(inChar);
    }
  } else digitalWrite(LED_BUILTIN, HIGH);
  if (Serial.available()) {
    size_t len = Serial.available();
    uint8_t sbuf[len];
    Serial.readBytes(sbuf, len);
    if (sClient.connected()) {
      sClient.write(sbuf, len);
      Serial.write(sbuf, len);
    }
  }
}
```

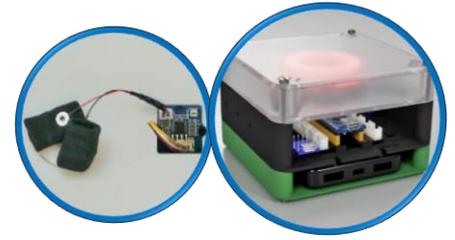
```

int SSID_scan() {
  int frei = 0;
  Serial.println("scan start");
  WiFi.disconnect();
  delay(100);
  int n = WiFi.scanNetworks();
  if (n == 0) {
    Serial.println("no networks found");
    frei = 1;
  } else {
    int belegt[n];
    int staerke[n];
    Serial.print(n);
    Serial.println(" networks found.");
    for (int i = 0; i < n; ++i) {
      belegt[i] = WiFi.channel(i);
      staerke[i] = WiFi.RSSI(i);
      delay(10);
    }
    for (int i = 0; i < 12; ++i) {
      int diff = belegt[i + 1] - belegt[i];
      if (diff > 1) {
        frei = belegt[i] + 1;
        break;
      }
    }
    if (frei != 0) {
      Serial.print("done. free channel: ");
      Serial.println(frei);
      return frei;
    } else {
      int maxnummer = 0;
      int maxstaerke = staerke[maxnummer];
      for (int j = 0; j < n; j++) {
        if (maxstaerke > staerke[j]) {
          maxnummer = j;
          maxstaerke = staerke[maxnummer];
        }
      }
      frei = belegt[maxnummer];
      Serial.print("done. weakest channel: ");
      Serial.println(frei);
      return frei;
    }
  }
}
}
}

```

## EDA-Cube Arduino Nano code

*Abbildung 11 ein funktionierendes Beispielfür den EDA-Cube während aller Lektionen*



```
#include <Adafruit_NeoPixel.h>
#define NEOPIXELPIN 6
#include <SoftwareSerial.h>
SoftwareSerial Serial_45(4, 5);

Adafruit_NeoPixel pixels(16, NEOPIXELPIN, NEO_RGBW + NEO_KHZ800);
const int GSR = A6;
long sum = 0;
int gsr_average, sensorValue, r, g, gsr_alt, delta = 0;

void setup() {
  Serial.begin(9600);
  Serial_45.begin(9600);
  pixels.begin();
  pixels.clear();
}

void loop() {
  pixels.clear();
  sum = 0;
  for (int i = 0; i < 20; i++)
  {
    sensorValue = analogRead(GSR);
    sum += sensorValue;
    delay(5);
  }
  gsr_average = sum / 10;
  delta = abs(gsr_average - gsr_alt);
  delta = constrain(delta, 0, 255);
  gsr_alt = gsr_average;
  Serial.println(gsr_average);
  Serial_45.println(gsr_average);
  r = 255 - (int) ((gsr_average - 600) / 3.125);
  g = (int) ((gsr_average - 600) / 3.125);
  for (int i = 0; i < 16; i++) {
    pixels.setPixelColor(i, pixels.Color(g, r, 0, delta));
  }
  pixels.show();
  delay(10);
}
```