

'We are the makers - IoT' Scénario d'apprentissage: biofeedback avec des capteurs de santé IoT

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

The following paper was developed and tested in a school-environment with ca. 18 students of age 13-17 in the schoolyear 2018/2019. It reflects the experience with many meanders and some failures. Since the IoT-field is complex, teaching materials must be chosen carefully. This paper is supposed to be a recommendation, as a starting point.

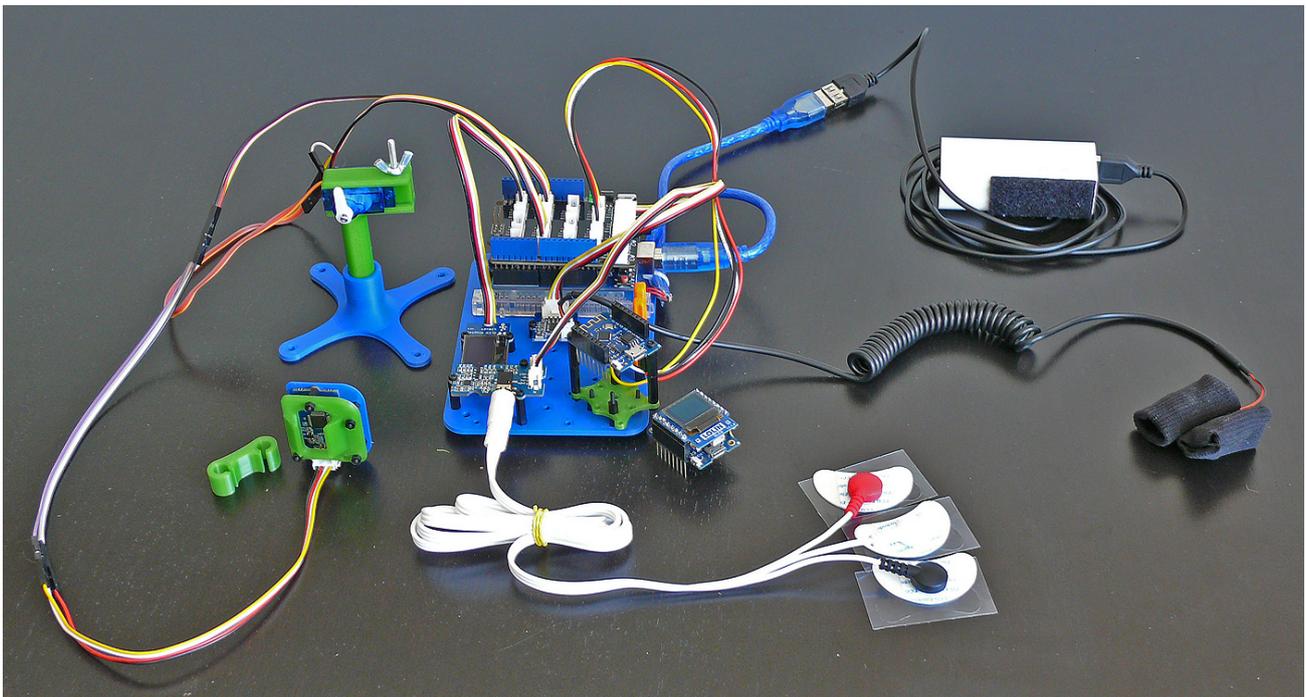
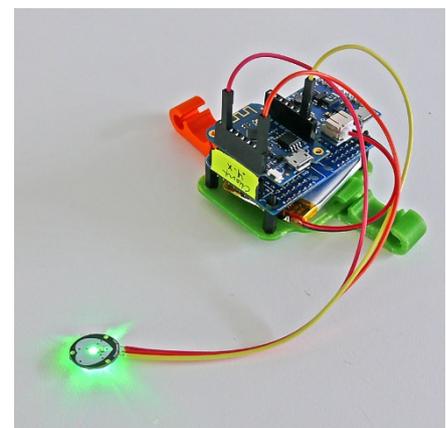


Figure 1: Prototype of a IoT biofeedback station

What about a self-made lie detector or a fitness tracker with self-programmed control software? Or a pulsometer which helps me to relax and keeps track of my strain?

What about a device which gives me feedback of my muscle tone and helps me to control myself in stress situations? Controlling the output of a machine with the power of my thoughts – possible or not?



And: what about sleep tracking devices which help to optimize my sleep and inform a physician if I'm not ok? Where are possibilities and threats of modern IoT biosensor technology?

1. Title du Scénario	biofeedback avec des capteurs de santé IoT
2. Groupe cible	14 - 17 ans
3. Durée	Au minimum 5 semaines à raison de 2 leçons de 45min par semaine : soit environ 6-8 heures.
4. Besoins couverts par l'activité	<ul style="list-style-type: none"> ▪ Interaction entre les composants électroniques et les corps humains ▪ Surveillance et modification des paramètres biologiques humains ▪ Chaîne de communication des appareils IoT ▪ Principes des capteurs et acteurs ▪ Différents principes de mesure des biosignaux. ▪ EMG I: Comment fonctionne le système de conduction musculaire? ▪ EMG II: Principes des amplificateurs d'instrumentation ▪ Principes des réseaux de communication sans fil ▪ Construction et impression 3D d'assistants de mesure.
5. Résultats attendus	<ul style="list-style-type: none"> ▪ Comment fonctionne un système IoT? ▪ Quelles sont les possibilités et les limites des systèmes IoT pour la santé? ▪ Quels composants - matériels et logiciels - sont essentiels pour construire un appareil IoT? ▪ Comment fournir du biofeedback pour aider les gens?
6. Méthodologies	Dans ce scénario, les élèves vont concevoir, construire et programmer par eux-mêmes un dispositif de biosignal interactif à partir de zéro. Les élèves utiliseront également le moniteur série Arduino et le traceur série pour visualiser et tracer le biofeedback.
7. Lieu	<ul style="list-style-type: none"> ▪ Un laboratoire avec un ensemble de pièces et composants électroniques; ▪ Chaque groupe d'élèves doit avoir un ordinateur ou un ordinateur portable avec des privilèges administratifs pour installer différents logiciels ▪ Un projecteur pour l'enseignement des tutoriels et la présentation des travaux des étudiants; ▪ Chaque étudiant doit tenir un journal de laboratoire

8. Outils / Matériaux / Ressources

Imprimantes 3D

Environ 2-3 imprimantes 3D sont nécessaires car les étudiants imprimeront leur Stations IoT de biofeedback. Bien sûr, il est possible pour les étudiants de construire les pièces des machines par eux-mêmes

Composants imprimés en 3D:

Comme point de départ, toutes les pièces nécessaires sont fournies au format .stl et en tant que fichiers Autodesk Fusion 360.

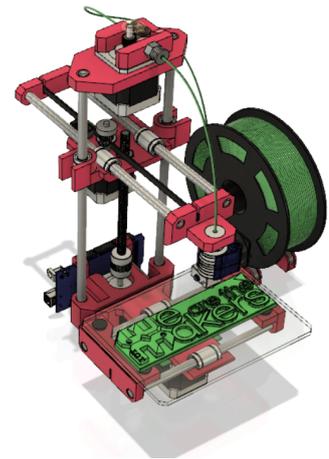


Figure 2: Imprimante 3D

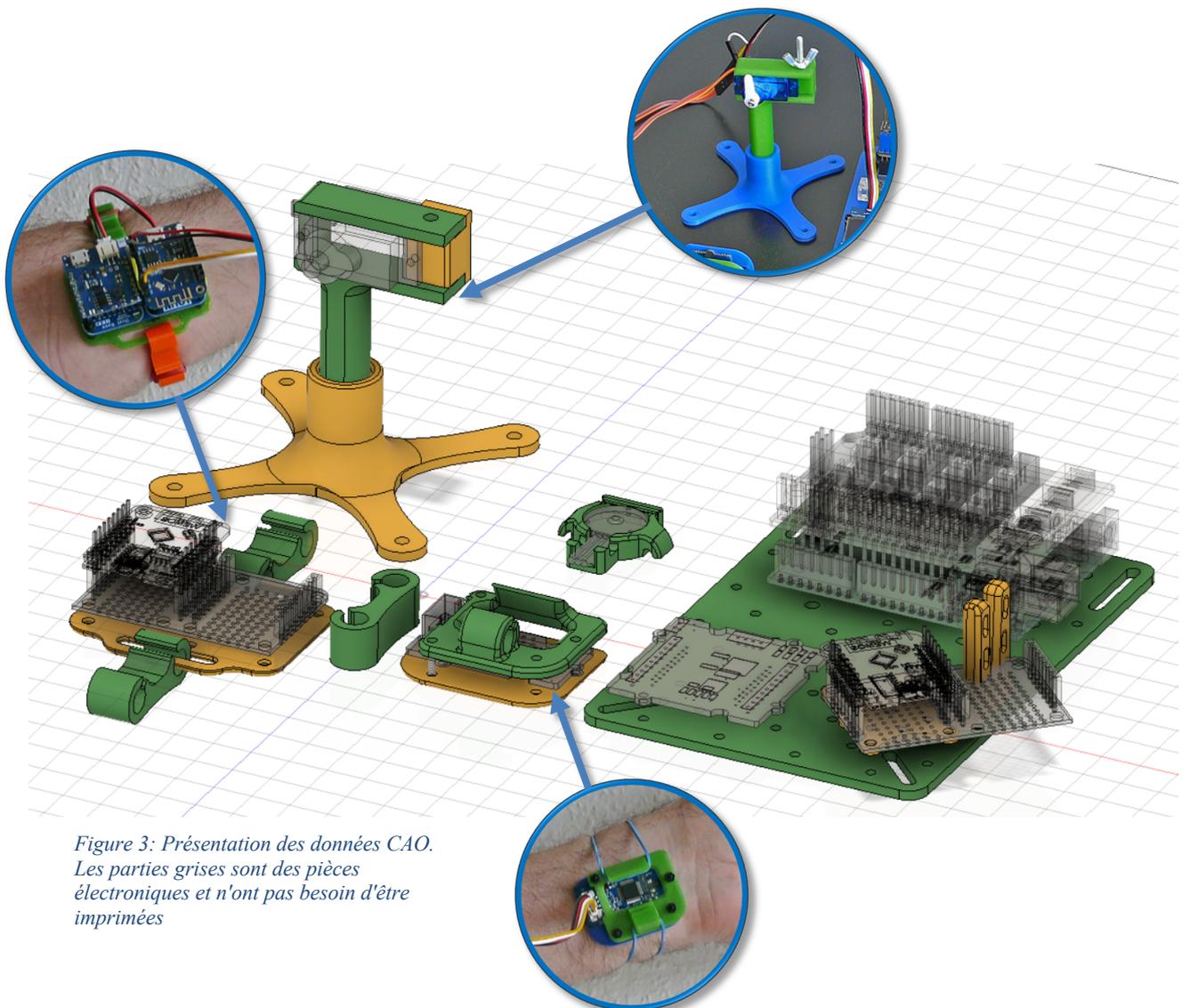


Figure 3: Présentation des données CAO.
Les parties grises sont des pièces
électroniques et n'ont pas besoin d'être
imprimées

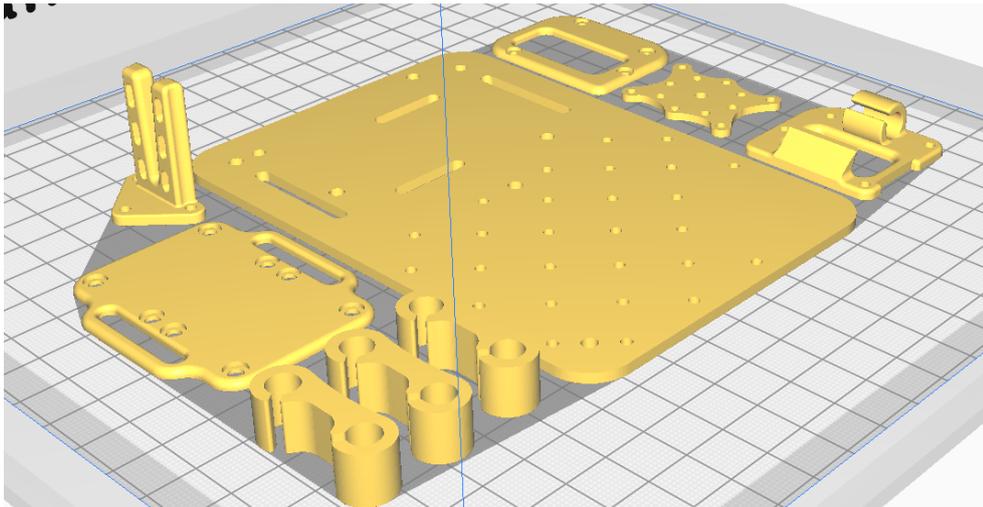
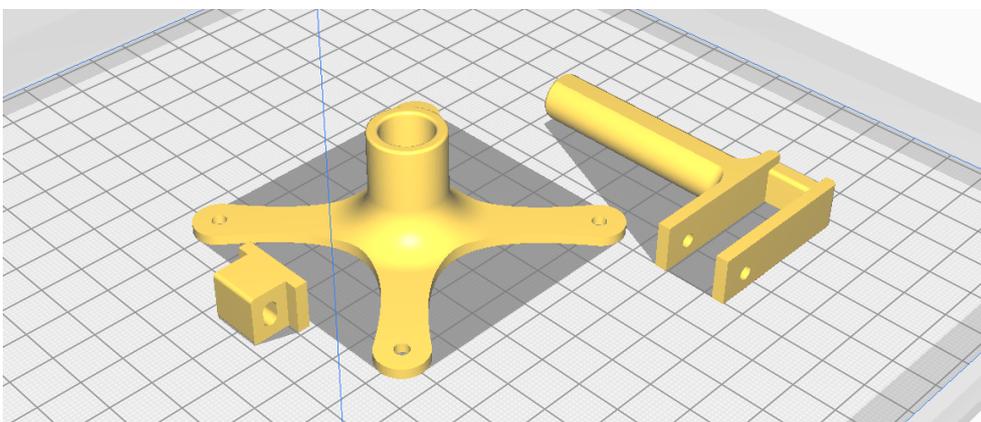


Figure 4: Ensemble de fichiers .stl prêts pour l'impression



Composants électroniques :

ATTENTION: Puisque nous faisons des expériences avec le corps humain, toutes les précautions doivent être prises! Ne connectez jamais un corps humain au système d'alimentation domestique. Le corps humain doit toujours être complètement éloigné du réseau électrique!

Cela comprend également les adaptateurs secteur qui sont branchés sur la prise murale. Ce type de circuits doit être évité. N'utilisez que des piles et accumulateurs à basse tension d'env. 3-5V.

Dans ce travail, nous recommandons le système Seeed Grove comme base pour sa facilité d'utilisation: (http://wiki.seeedstudio.com/Grove_System/) Tous les composants de base à l'exception des puces Wemos, des accumulateurs et des capteurs de fréquence cardiaque appartiennent à la norme Grove:

Composants Seeed Studio :

- 1: Shield Grove de Base pour Arduino-Uno (http://wiki.seeedstudio.com/Base_Shield_V2/)
2. Grove OLED 128x64 (http://wiki.seeedstudio.com/Grove-OLED_Display_0.96inch/)
3. Détecteur EMG Grove (http://wiki.seeedstudio.com/Grove-EMG_Detector/)
4. Fréquence cardiaque Grove Finger Clip http://wiki.seeedstudio.com/Grove-Finger-clip_Heart_Rate_Sensor/
5. GSR Grove http://wiki.seeedstudio.com/Grove-GSR_Sensor/

Capteurs et acteurs standards:

- 1: Arduino Uno (ou équivalent)
- A: **2x** Wemos LOLIN D1 mini (ou équivalent)
- B: Capteur de pouls analogique (www.pulsesensor.com)
- C: Shield de batterie Wemos (https://wiki.wemos.cc/products:d1_mini_shields:battery_shield)
- D: Micro servo moteur.

Pièces diverses:

- Ruban adhésif
- Bande de cuivre auto-adhésive
- Entretoises M3 Nylon (Hexagonales)
- Entretoises M2 Nylon (Hexagonales) pour Grove (trous de 2mm)
- Câbles Grove
- Connecteurs WAGO
- Fils de liaison
- Alimentation USB pour charger le bloc d'alimentation USB
- Boc d'alimentation USB à courant maximum
- Un fer à souder pour souder les câbles aux composants électroniques



Figure 5: Nylon Standoffs

Des ordinateurs avec les logiciels suivants préinstallés :

- Autodesk Fusion 360 (ou tout autre logiciel de modélisation 3D, par ex. Wings3D)
- Trancheur CURA,
- Une connexion Internet pour télécharger des bibliothèques
- IDE Arduino
- IDE Processing

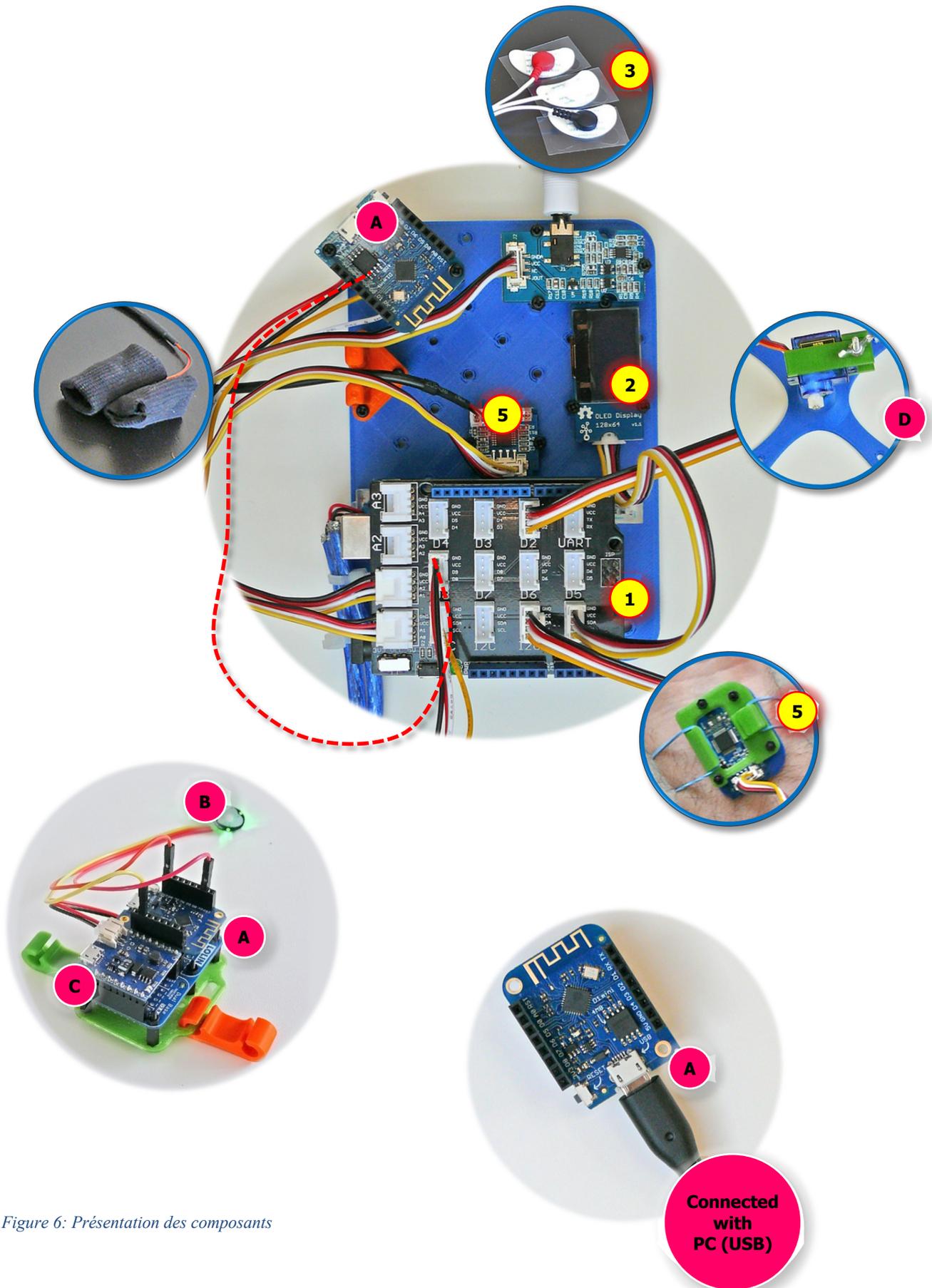


Figure 6: Présentation des composants

Bibliothèques Arduino pour les composants :

Certains composants comme le Wemos D1 Mini ou certaines des cartes Grove ont besoin de bibliothèques pour que l'IDE Arduino fonctionne correctement.

L'importation des bibliothèques est décrite ici :

<https://www.arduino.cc/en/Guide/Libraries>

OLED lib (Seeed):

https://github.com/Seeed-Studio/OLED_Display_128X64/archive/master.zip

Software I2C Master (Felix Fogg):

<https://github.com/felix-fogg/SoftI2CMaster>

Preference URL for WEMOS-Boards (ESP8266):

Pour installer les wemos, la «définition de carte» doit être installée. Elle est décrite ici :

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

1. Depuis l'IDE Arduino, ouvrez la fenêtre Préférences.
2. Entrez l'URL suivante dans le champ «Gestionnaire de carte additionnelle (Additional Board Manager)»: http://arduino.esp8266.com/stable/package_esp8266com_index.json
3. Ouvrez le gestionnaire de cartes dans le menu Outils> Carte et recherchez la plateforme esp8266.
4. Sélectionnez la version actuelle dans une liste déroulante et cliquez sur le bouton "installer".
5. Sélectionnez «(LOLIN) Wemos D1 R2 et Mini» dans le menu Outils> Carte après l'installation.

Pulsesensor:

<https://pulsesensor.com/pages/installing-our-playground-for-pulsesensor-arduino>

Le Grove GSR, le Grove Finger Clip Sensor et le détecteur EMG Grove n'ont besoin d'aucune bibliothèque, car ils peuvent être contrôlés avec de simples commandes Arduino.

Wemos D1 mini comme connexion sans fil entre les composants électroniques

- **• Les tableaux Wemos doivent être préparés par l'enseignant avant le début de la leçon et non par les élèves !**
- Les cartes Wemos-ESP8266-Wifi sont conçues comme une alternative moins coûteuse à la technologie Xbee, fiable mais aussi onéreuse.

Deux Wemos construisent une paire qui est connectée via le port Ethernet Wifi 23 (port Telnet). Le but est de **remplacer le câble de communication série**. Habituellement, un appareil électronique expérimental est connecté via un câble USB au PC. Pour obtenir une conception complètement autonome qui n'est pas connectée au réseau électrique domestique, une connexion sans fil doit être établie.

Par conséquent, la communication série habituelle (UART) est traduite en Wifi et envoyée par un Wemos, reçu par les autres Wemos et retraduit en communication série à nouveau. Pour des raisons de compatibilité, le débit en bauds est fixé à 9600 bauds, car la communication logicielle-série par un Arduino Uno est limitée à 9600 bauds.

Une mini-paire Wemos D1 se compose d'un serveur et d'un client. Le serveur doit être connecté au PC. Il doit être démarré en premier et effectue les étapes suivantes:

1. Analyse de tous les réseaux wifi disponibles,
2. Déterminer, s'il y a un canal libre inutilisé ou un réseau faible en arrière-plan,
3. Création d'un point d'accès Wifi en utilisant le premier canal gratuit, également combiné avec DHCP
4. Attendre UN (un seul!) Client qui se connecte.
5. Si le client se déconnecte, le serveur attendra que le client se reconnecte.
6. Si le serveur est réinitialisé, recommencez simplement à l'étape 1. (analyse des réseaux)

Le client doit être démarré en tant que deuxième et se connectera et se reconnectera automatiquement. Comment configurer le serveur et le client Wemos, expliqué dans «Meilleur code source du serveur»:

Voici les extraits pertinents du code source du serveur et du client qui doivent être adaptés pour configurer des paires individuelles de cartes Wemos:



Figure 7: extrait du code source du serveur



Figure 8: extrait du code source du serveur

- Les deux lignes de code soulignées doivent être exactement les mêmes pour une paire Wemos.
- Les deux lignes de code soulignées doivent être adaptées à chaque paire Wemos.

Changez l'adresse IP en
192.168.1.1 OU 192.168.2.1 OU 192.168.4.1 OU 192.168.5.1... etc.

Changez le ssid en
«Erasmus1» OU «Erasmus2» OU «Erasmus4» OU «Erasmus5»... etc.

... Compiler les scripts à l'intérieur de l'IDE Arduino et les télécharger sur les cartes Wemos appropriées.

8b Théorie des potentiels axonaux et mesure EMG

Ce texte se veut un bref aperçu et peut être considéré comme une collection de mots clés importants. Il n'est pas conçu comme un manuel!

L'article suivant est basé sur le «EMG Fibel V1.1.pdf», «et les entrées Wikipedia des mots-clés» Action Potential »,« Design of an EMG Detector »,« EMG »dans le« recueil électronique »:

- <http://www.velamed.com/wp-content/uploads/EMG-FIBEL-V1.1.pdf>
- https://en.wikipedia.org/wiki/Action_potential
- <https://iem.kug.ac.at/sid/sonic-interaction-design/forschung/hardware-software-prototyping/design-and-evaluation-of-an-emg-based-recording-and-detection-system.html>
- <https://www.elektronik-kompodium.de/public/schaerer/emg1.htm>

Les muscles se contractent car ils reçoivent des signaux électriques des nerfs: Ce que l'on appelle «potentiel d'action» est un changement potentiel électrique des terminaisons nerveuses intramusculaires. Le changement se produit entre -80 mV («potentiel de repos») et + 30 mV («dépolariation»), ce qui fait au total un changement de potentiel théoriquement mesurable de 110 mV. Étant donné que les terminaisons nerveuses se trouvent à l'intérieur des fibres musculaires, nous ne pouvons (à l'école) mesurer que sur la peau. En conséquence, une grande partie de la force du signal des potentiels d'action disparaît en raison de la résistance électrique de la peau et du tissu conjonctif. Un changement typiquement mesurable du potentiel électrique est ca. 30mV.

Les microcontrôleurs capables de calculer ces signaux utilisent des convertisseurs analogiques-numériques. Ces ADC ont généralement une plage de tension d'entrée de 0 à 3,3 V (type Wemos) ou de 0 à 5 V (Type Arduino Uno). La résolution des deux ADC est de 10 bits, ce qui signifie que les microcontrôleurs peuvent diviser une plage de mesure complète de 3,3 V en 1024 étapes discrètes: $3,3 \text{ V} / 1024 = 3,2 \text{ mV}$. Si nous pouvions mesurer les changements du potentiel cutané avec un microcontrôleur autonome, nous ne pourrions atteindre qu'une plage entre 0 et 10 de 1024 valeurs discrètes possibles. C'est beaucoup moins. De plus, le potentiel cutané du corps humain change avec l'influence des champs électriques externes, ce qui conduit à des dérives potentielles 1000 fois plus importantes que la force du signal.

Par conséquent, nous avons besoin d'une composante électronique en amont qui a) amplifie notre signal de 0,03 V à 3,3 V et b) est capable de compenser la dérive du champ électrique. Ce que l'on nomme amplificateur d'instrumentation est un circuit à trois entrées: une entrée pour «+», une pour «-» et une pour référence. Alors que le but des électrodes «+» - et «-» est de mesurer la différence de potentiel de 0,03 V, l'électrode de référence prendra soin de l'influence des champs électriques externes pour compenser la dérive.

À l'intérieur de cet instrument de précision électronique, le signal est amplifié et c'est tout ce qu'il fait ! Nous pouvons maintenant mesurer et numériser le potentiel de sortie avec le microcontrôleur.

Les électrodes doivent être placées avec soin, car elles doivent mesurer les potentiels nerveux, et plus la distance au nerf se est petite, meilleur est le signal: un signal nerveux parcourt le muscle à une vitesse d'environ. 5m / s. Si nous mesurons au milieu des biceps, les deux électrodes +/- - devrait être placé ca. 10 cm les uns des autres, tandis que l'électrode de référence doit être placée à un endroit éloigné de celui-ci, par ex. sur la main. Comme mesure attendue, il devrait y avoir un temps de transit du signal d'une électrode à l'autre ("onde de dépolarisation"):

$$\frac{\text{distance des électrodes en cm}}{\frac{500 \text{ cm}}{\text{seconde}}} = 20 \text{ millisecondes}$$

Dans une mesure de peau EMG, nous verrons une superposition de nombreux signaux provenant de nombreuses fibres musculaires différentes et donc une forme d'onde de 20 ms sera difficilement reconnaissable.

Pour contracter un muscle à force variable, la valeur du potentiel d'action ne change pas. Il n'y a qu'un changement d'occurrence des changements de tension nerveuse: plus un muscle est dur à contracter, plus la «cadence de tir» des potentiels d'action des nerfs est élevée..

8c Une théorie du biofeedback

Ce texte se veut un bref aperçu et peut être considéré comme une collection de mots clés importants. Il n'est pas conçu comme un manuel!

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

La réponse d'un corps humain au stress ou aux influences externes se produit la plupart du temps automatiquement et inconsciemment. Par exemple, si un homme ment ou a peur, sa peau commence à transpirer. Cette sueur peut être mesurée comme un changement de conductivité électrique, car la sueur contient des électrolytes. Si l'ordinateur de mesure visualise ce changement, l'humain peut corréliser son état émotionnel avec le signal mesuré et peut essayer d'influencer sa réaction et apprendre à contrôler ses émotions. Les émotions cachées précédentes sont devenues maintenant conscientes à l'esprit de cette personne.

Il existe de nombreux exemples et expériences que les étudiants peuvent essayer par eux-mêmes:

- Influencer la fréquence cardiaque avec le changement de fréquence respiratoire, surveillée par des capteurs de pouls
- Influencer les réactions de peur avec changement de l'activité électrodermique, surveillé par des capteurs GSR

(Une réaction de peur pourrait être: une image d'une araignée, une vidéo youtube d'une montagne russe)

- Un polygraphe (détecteur de mensonge) est - entre autres - basé sur le changement d'activité électrodermique et peut être mesuré avec des capteurs GSR

- Coactivité des muscles: le typage informatique dans des conditions de stress conduit à la contraction du muscle trapezius dans le cou humain. Cela peut être mesuré avec EMG.

9. Plan de leçon: description étape par étape de l'activité / du contenu

Leçons 1 et 2 (90min):

Les étudiants seront initiés à l'IoT par des exemples: robots aspirateurs avec télécommandes d'application, stations météorologiques basées sur Internet, agriculture intelligente et enfin et surtout des applications de santé. Les élèves devraient examiner le fonctionnement de ces appareils et les composants nécessaires : un système basé sur un microcontrôleur contrôle et coordonne les capteurs et les acteurs connectés. De plus, il communique et se coordonne avec d'autres systèmes de type similaire souvent via des réseaux de communication sans fil. Pièces nécessaires : capteurs, acteurs, appareils de communication. Les possibilités et les risques doivent être discutés ainsi que les limites: où l'IoT a-t-il un sens et où n'en a-t-il pas?

Leçons 3 et 4 (90 min)

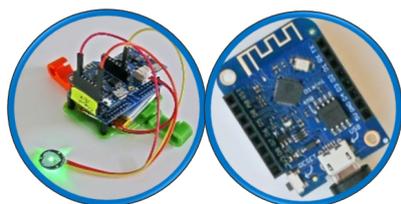


Figure 9: Pièces nécessaires

Notions de base sur les biocapteurs (capteur d'impulsions): les étudiants se voient présenter le premier capteur qui n'a pas besoin d'être isolé galvaniquement: le capteur d'impulsions. Il se compose de deux parties différentes: une simple LED verte et un phototransistor plus un circuit amplificateur directement connecté. Il doit être placé directement au-dessus d'une veine, par exemple le bout des doigts ou le bout des oreilles.

Si la veine est gonflée depuis que le cœur pompe, le sang réfléchira la lumière verte et le phototransistor reconnaîtra une valeur élevée. Si la veine se contracte depuis le cœur se contracte, le sang manquant entraînera une plus grande absorption de la lumière à l'intérieur du tissu conjonctif. Le

phototransistor mesurera une valeur plus petite. Les étudiants devraient être initiés à la programmation analogique Arduino et au calcul de valeurs analogiques: le convertisseur analogique-

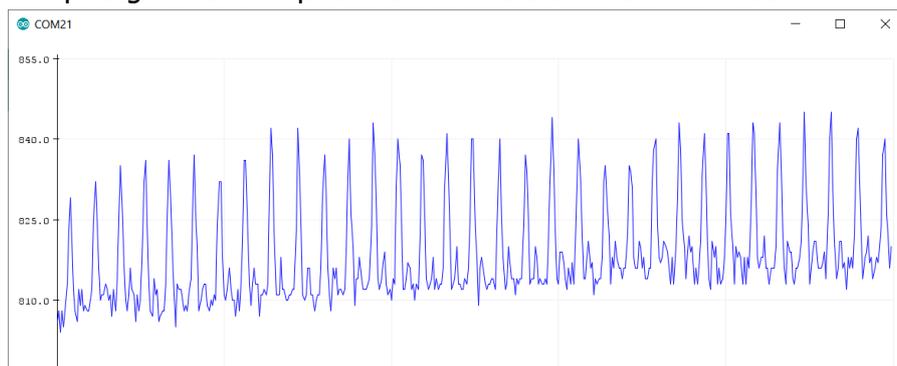
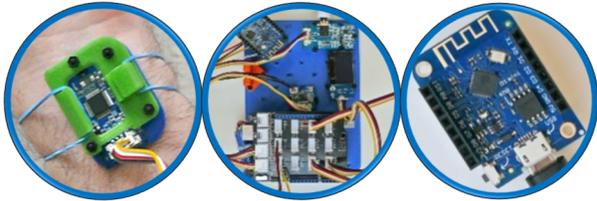


Figure 10: Exemples de données mesurées par le capteur d'impulsions

numérique (ADC) doit être présenté et comment il fonctionne. Une très bonne introduction peut être trouvée ici:

<https://www.generationrobots.com/media/DetecteurDePoulsAmplifie/PulseSensorAmpedGettingStartedGuide.pdf>

Leçons 5 et 6 (90 min)



Biofeedback de la fréquence cardiaque: les élèves apprennent que de nombreuses réactions corporelles correspondent inconsciemment à des processus émotionnels. Mais si nous comprenons ces réactions corporelles cachées, nous pouvons

commencer à les contrôler: le biofeedback est un suivi en temps réel des réponses physiques dans le but de contrôler les émotions. (<https://www.artofmanliness.com/articles/hack-your-mind-like-a-twenty-first-century-soldier-using-biofeedback-to-become-more-resilient/>).

Si le logiciel du capteur de pouls fonctionne correctement, les élèves peuvent commencer à mesurer les changements de pouls et essayer d'influencer leur fréquence cardiaque pendant la respiration: Quelle est l'influence de la fréquence respiratoire sur la fréquence cardiaque? Que se passe-t-il si nous respirons plus vite ou plus lentement? Qu'est-ce que cela signifie pour les situations de stress? Pouvons-nous prendre conscience de notre rythme cardiaque?

Utilisez cette procédure de configuration pour une expérience de biofeedback :



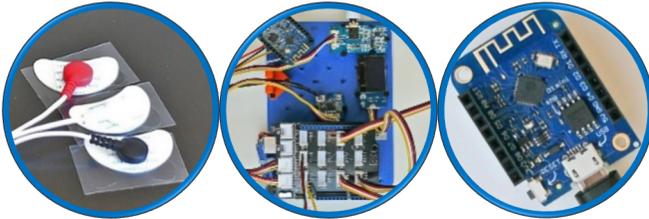
Figure 11: Application de traitement des captures d'écran: influence de la luminosité / du contraste sur la fréquence cardiaque

- a) Capteur digital de fréquence cardiaque I2C Grove attachée à un doigt de la main de l'élève
- b) Connectez la station Arduino au PC et démarrez le code source de traitement. Plus la fréquence cardiaque est rapide, plus l'image est sombre: influencez la luminosité / le contraste avec le contrôle mental de l'activité cardiaque.
- c) Procédez comme suit:
 - Inspirez lentement une respiration profonde pendant 4 secondes.

- Retenez votre respiration pendant 4 secondes.
- Expirez lentement le souffle pendant 4 secondes.
- Retenez le souffle vide pendant 4 secondes.
- Répétez jusqu'à ce que votre respiration soit sous contrôle.

([https://en.wikipedia.org/wiki/Dave_Grossman_\(author\)](https://en.wikipedia.org/wiki/Dave_Grossman_(author)))

Leçons 7 et 8



EMG Basics: Enseignement des mots-clés et des relations du potentiel actif des nerfs, de la tension musculaire et des principes de relaxation et de mesure. Les étudiants doivent comprendre comment fonctionne le code source de l'amplification du signal musculaire.

Comment placer les trois électrodes sur le bras: dans un premier temps, la personne testée doit laver le bras avec du savon et de l'eau et ensuite les positions où les électrodes seront placées doivent être nettoyées avec de l'alcool et du coton. Ensuite, les trois électrodes peuvent être placées.

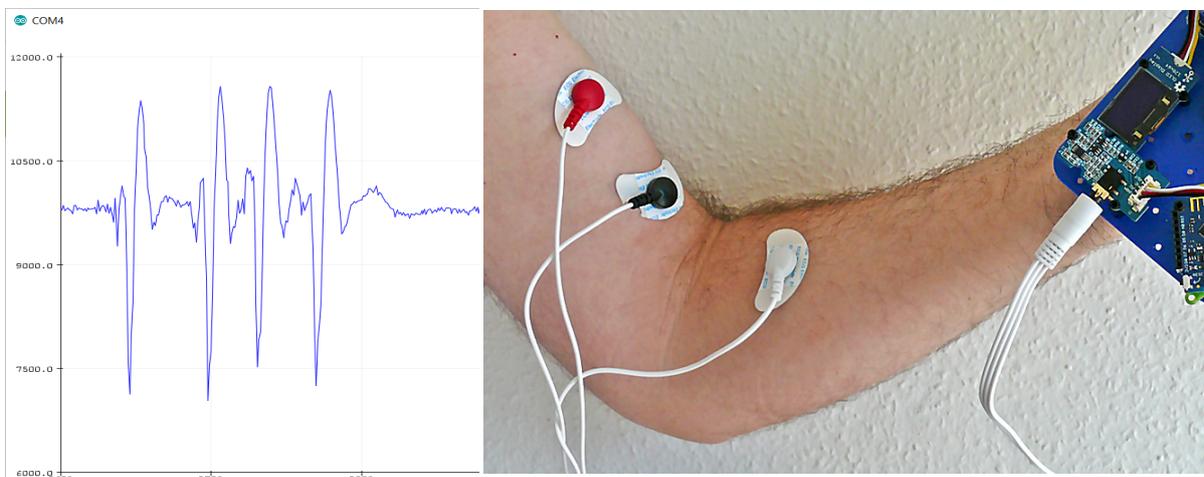
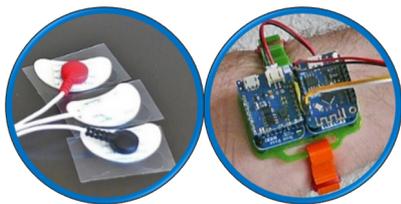


Figure 12: *Activité du biceps et placement d'électrode correspondant*

Les élèves devraient expérimenter différentes vitesses de mouvement des bras, différents poids de levage et l'effet de la relaxation. Que se passe-t-il si - avec exactement ce placement d'électrode - on ferme et ouvre la main? Que se passe-t-il si l'on retire l'électrode blanche?

Leçons 9 et 10



Expérience de biofeedback EMG:

L'expérience suivante est basée sur la thèse de doctorat

de Michael Schnoz:

<https://www.research-collection.ethz.ch/handle/20.500.11850/149225>

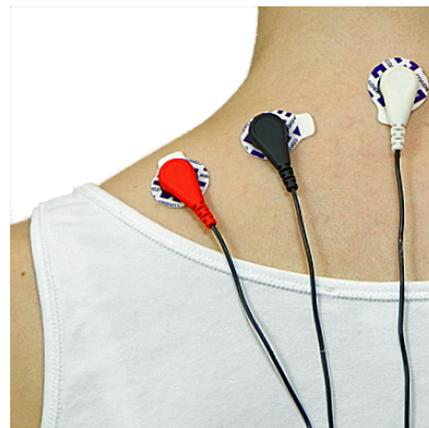


Figure 13: positionnement d'électrodes dans le cou

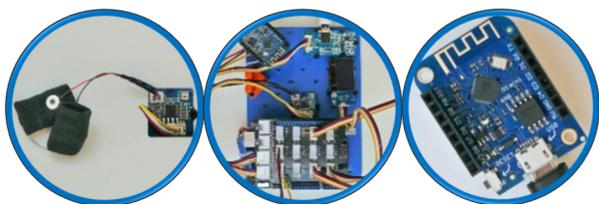
Si une personne testée tape des lettres sur l'écran d'ordinateur rapidement et est fortement concentrée, cela entraînera probablement des douleurs dans le cou. Douleurs qui peuvent être mesurées et ensuite influencées par la personne testée elle-même.

Pour mettre la personne testée dans des conditions de stress, elle pourrait utiliser le logiciel de didacticiel de dactylographie suivant :

https://portableapps.com/apps/education/tipp10_portable

Plus la personne travaille vite, plus la personne - et probablement son cou - sera tendue. La personne testée peut maintenant essayer d'influencer ce temps.

Leçons 11 et 12



Biofeedback GSR

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

Cette leçon est basée sur les réactions émotionnelles lors de la visualisation d'un «film

effrayant»: les hauts et les bas des montagnes russes peuvent avoir un effet énorme sur les sentiments des personnes testées. Comment l'influencer?

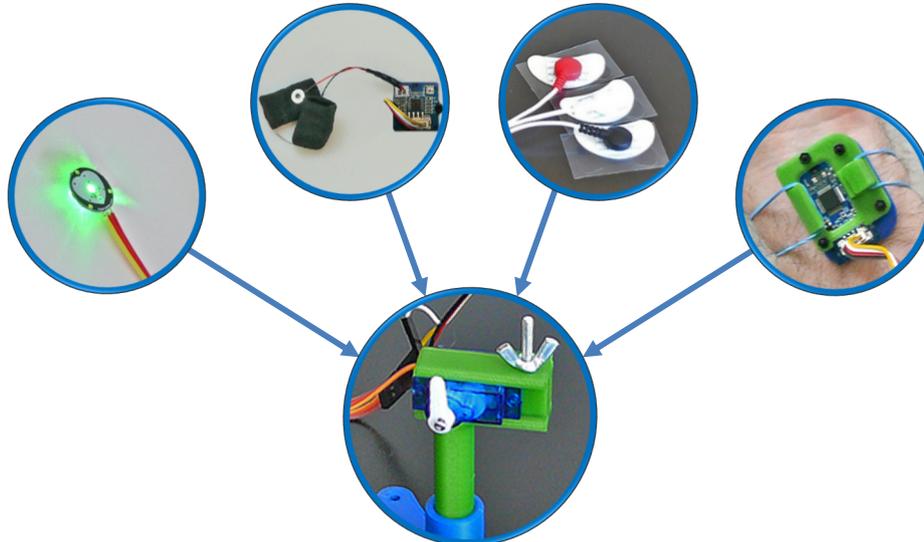
Qu'en est-il des photos d'araignées ou de serpents?

Ou de quelque chose de délicieux / agréable comme la musique? Quel est l'effet de la musique disco / musique classique? Y a-t-il un effet spécial en écoutant votre chanson préférée?

Leçon 13 pour terminer (270 min):

Programmation freestyle! Et heureux biofeedback! 😊

Essayez de combiner la sortie des différents capteurs avec le servomoteur. De quelles possibilités disposez-vous pour contrôler l'angle de rotation des servos ?



<p>10. Retour d'information</p>	<p>À la fin de la leçon, les étudiants devraient avoir une connaissance approfondie du fonctionnement des principes de l'IoT dans les dispositifs médicaux et de la manière dont le biofeedback peut aider à comprendre les caractéristiques cachées de notre corps. Pendant la leçon, des aspects importants de l'électronique, de l'informatique médicale et des bases de la construction ont été enseignés. De plus, les aspects biologiques des activités musculaires ont été enseignés.</p>
<p>11. Evaluation</p>	<p>Les étudiants tiennent leur journal du travail, qui peut être examiné par l'enseignant. Les étudiants peuvent également présenter les résultats de leurs expériences. De plus, un test standard en classe doit être effectué à la fin des cours.</p>

Wemos Client Sourcecode

```
#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);
    }
  }
}
```



Figure 14: Source code for the Wemos client attached to the Arduino station

Wemos Server Sourcecode

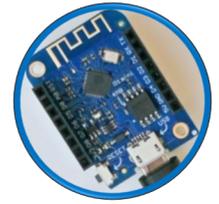


Figure 15: This script should be compiled for the Wemos connected with PC

```
#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;
        }
    }
}
}
```

EMG Sourcedode example

```
#include <SeeedOLED.h>
#include <Wire.h>
#include <SoftwareSerial.h>
SoftwareSerial Serial_89(8, 9);

int max_analog_dta      = 300; // max analog data
int min_analog_dta      = 100; // min analog data
int static_analog_dta   = 0;  // static analog data
int level = 5;

int getAnalog(int pin) {
    long sum = 0;
    for (int i = 0; i < 32; i++){
        sum += analogRead(pin);
    }
    Serial.println(sum);
    Serial_89.println(sum);
    int dta = sum >> 5;
    max_analog_dta = dta > max_analog_dta ? dta : max_analog_dta;
    min_analog_dta = min_analog_dta > dta ? dta : min_analog_dta;
    return sum >> 5;
}

void setup() {
    Wire.begin();
    Serial.begin(9600);
    Serial_89.begin(9600);
    SeeedOled.init();
    SeeedOled.clearDisplay();
    SeeedOled.setNormalDisplay();
    SeeedOled.setPageMode();
    SeeedOled.setTextXY(1, 0);
    SeeedOled.putString("EMG prototype");
    long sum = 0;

    for (int i = 0; i <= 10; i++) {
        for (int j = 0; j < 100; j++){
            sum += getAnalog(A0);
            delay(1);
        }
    }
    sum = sum / 1100;
    static_analog_dta = sum;
    Serial.print("static_analog_dta = ");
    Serial.println(static_analog_dta);
}

void loop() {
    int val = getAnalog(A0);
    int level2;
    if (val > static_analog_dta) {
        level = 5 + map(val, static_analog_dta, max_analog_dta, 0, 10);
    } else {
        level = 5 - map(val, min_analog_dta, static_analog_dta, 0, 10);
    }
    for (int i = 0; i < 10; i++) {
        SeeedOled.setTextXY(1, i);
        SeeedOled.putChar(32);
    }
    for (int i = 0; i < level - 5; i++) {
        SeeedOled.setTextXY(1, i);
        SeeedOled.putChar(124);
    }
    delay(20);
}
```

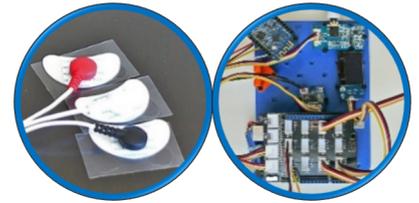


Figure 16: Script for lessons 7&8 at the Arduino Station

Pulse-Sensor on Wemos Client using AnalogRead

```
#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);
  pinMode(A0, INPUT);
  WiFi.setSleepMode(WIFI_NONE_SLEEP);
  WiFi.mode(WIFI_STA);
  WiFi.setOutputPower(10); // 10: 10mW, 14: 25mW, 17: 50mW, 20: 100
  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 Schnittstelle verfügbar ist,
    char c = client.read(); //lesen was reinkommt
    Serial.print(c);
  }
  delay(50);
  if (client.connected()) { //und wenn Server-Client läuft, ...
    unsigned int value = analogRead(A0);
    Serial.println(value);
    String einsnachdemandern = String(value);
    for (int i = 0; i < einsnachdemandern.length(); i++) {
      client.write(einsnachdemandern[i]);
    }
    client.write(10);
  }
}
```

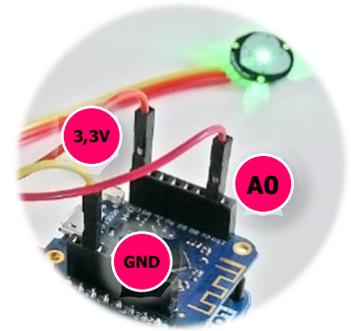


Figure 17: Wiring Wemos - Pulse Sensor

GSR Source code on Station client

```
#include <SeeedOLED.h>
#include <Wire.h>
#include <SoftwareSerial.h>
SoftwareSerial Serial_89(8, 9);

const int GSR = A1;
int sensorValue = 0;
int gsr_average = 0;

void setup() {
  Wire.begin();
  Serial.begin(9600);
  Serial_89.begin(9600);
  SeeedOled.init();
  SeeedOled.clearDisplay();
  SeeedOled.setNormalDisplay();
  SeeedOled.setPageMode();
  SeeedOled.setTextXY(1, 0);
  SeeedOled.putString("GSR prototype");
  delay(2000);
}

void loop() {
  long sum = 0;
  for (int i = 0; i < 20; i++)
  {
    sensorValue = analogRead(GSR);
    sum += sensorValue;
    delay(5);
  }
  gsr_average = sum / 10;
  Serial.println(gsr_average);
  Serial_89.println(gsr_average);
  SeeedOled.clearDisplay();
  SeeedOled.setTextXY(1, 2);
  SeeedOled.putNumber(gsr_average);
}
```

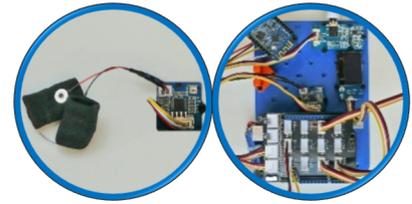
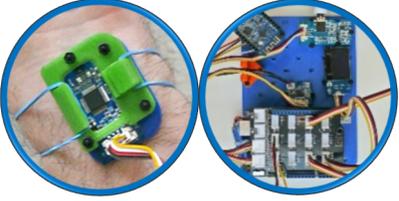


Figure 18: Source code for Arduino station lessons 11&12

Lesson 5&6 source code with processing app

Arduino sourcecode for Grove Finger clip sensor	Processing sourcecode for Grove Finger clip Sensor:
	
<pre> #define SDA_PORT PORTD #define SDA_PIN 3 #define SCL_PORT PORTD #define SCL_PIN 2 #include <SoftI2CMaster.h> #include <SoftWire.h> #include <SeeedOLED.h> #include <Wire.h> #include <SoftwareSerial.h> SoftwareSerial Serial_89(8, 9); SoftWire SWire = SoftWire(); void setup() { Wire.begin(); SWire.begin(); Serial.begin(9600); Serial_89.begin(9600); SeeedOled.init(); SeeedOled.clearDisplay(); SeeedOled.setNormalDisplay(); SeeedOled.setPageMode(); SeeedOled.setTextXY(1, 0); SeeedOled.putString("heartrateprototype"); delay(2000); } void loop() { SWire.requestFrom(0xA0 >> 1, 1); while (SWire.available()) { unsigned char c = SWire.read(); Serial.println(c, DEC); Serial_89.println(c, DEC); SeeedOled.clearDisplay(); SeeedOled.setTextXY(1, 2); SeeedOled.putString(c); } delay(500); } </pre>	<pre> PImage img; int heartrate; import processing.serial.*; Serial ardCom; String payload; String[] liste; void setup() { String portName = Serial.list()[0]; ardCom = new Serial(this, portName, 9600); size(1000, 500); frameRate(20); img = loadImage("biofeedback.png"); image(img, 0, 0); loadPixels(); } void draw() { if (ardCom.available() > 0) { payload = ardCom.readStringUntil('\n'); if (payload != null) { heartrate = parseInt(payload.trim()); println(heartrate); } } float percentage = -0.7/40*heartrate + 2.15; for (int x = 0; x < img.width; x++) { for (int y = 0; y < img.height; y++) { int loc = x + y*img.width; float r, g, b; r = red (img.pixels[loc]); g = green (img.pixels[loc]); b = blue (img.pixels[loc]); float bright = 64 - 64*percentage; float cont = percentage; r = r*cont - bright; g = g*cont - bright; b = b*cont - bright; r = constrain(r, 0, 255); g = constrain(g, 0, 255); b = constrain(b, 0, 255); color c = color(r, g, b); pixels[y*width + x] = c; } } updatePixels(); } </pre>