

'We are the makers – Scénario d'apprentissage IoT : EDA-Cube: avoir une idée de ce que ressent votre partenaire

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

L'article suivant a été développé en tant que dérivé de la station de rétroaction biologique (IO2). Comme la station de biofeedback a été construite avec une éducation générale à l'esprit, le cube EDA peut être utilisé à des fins non techniques comme par exemple un outil de surveillance des dialogues: il révèle les émotions de son utilisateur dans la mesure où la technologie EDA le permet.



Figure 1: Cubes EDA en action

Et si deux personnes parlent ensemble et veulent être polies l'une avec l'autre. Un gros problème est toujours que vous ne savez pas ce que ressent votre interlocuteur. Est-il / elle en colère, effrayé, nerveux, heureux, fatigué? Quels sont les effets de vos paroles? Êtes-vous - aux yeux de votre partenaire - impoli ou attentionné?

Y a-t-il une réponse immédiate à cela, au moins un «indice»?

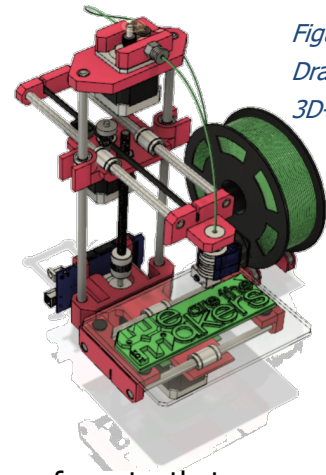
Et: que faire si vous portez une montre intelligente capable également de mesurer vos réactions émotionnelles? Voulez-vous que vos sentiments soient surveillés par une machine? Soyez conscient des capacités des capteurs modernes ! Construisons un appareil intelligent avec un capteur « d'émotion » qui interprétera les réactions de votre corps !

1. Titre du Scénario	Learn how monitor emotions using biosensorics measuring
2. Groupe cible	12- 17 ans
3. Durée	Au minimum 3 semaines de 2 leçons de 45min par semaine : en tout environ 6-8 heures.
4. Besoins couverts par l'exercice	<ul style="list-style-type: none"> ▪ Interaction entre les composants électroniques et le corps humain ▪ Surveillance et affectant les paramètres biologiques humains ▪ Chaîne de communication des appareils IoT ▪ Principes des capteurs et des acteurs ▪ Qu'est-ce que l'EDA? ▪ Principes des réseaux de communication sans fil ▪ Construction et impression 3D d'aides à la mesure.
5. Résultats attendus	<ul style="list-style-type: none"> ▪ Comment fonctionne un système IoT intelligent ? ▪ Quelles sont les possibilités et les menaces des systèmes EDA ? ▪ Quels composants - matériels et logiciels - sont essentiels pour créer un appareil IoT? ▪ Prise de conscience des biocapteurs et capacité à juger ces outils.
6. Méthodologies	Dans ce scénario, les élèves construiront et programmeront eux-mêmes un dispositif EDA interactif qui interprète les valeurs mesurées et les visualise. Les étudiants peuvent utiliser la technologie Wi-Fi pour transmettre les valeurs à un ordinateur pour un traitement ultérieur.
7. Lieu/ Environnement	<ul style="list-style-type: none"> ▪ Un laboratoire avec un ensemble de pièces et composants électroniques. ▪ Chaque groupe d'étudiants doit disposer d'un ordinateur ou d'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

8. Outils / matériaux / ressources

Imprimantes 3D

Deux ou trois imprimantes 3D sont nécessaires car les étudiants imprimeront leur Cubes EDA. Bien entendu, les élèves peuvent construire des pièces de machine par elles-mêmes.



*Figure 2:
Drawing of a
3D-Printer*

Composants imprimés en 3D :

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

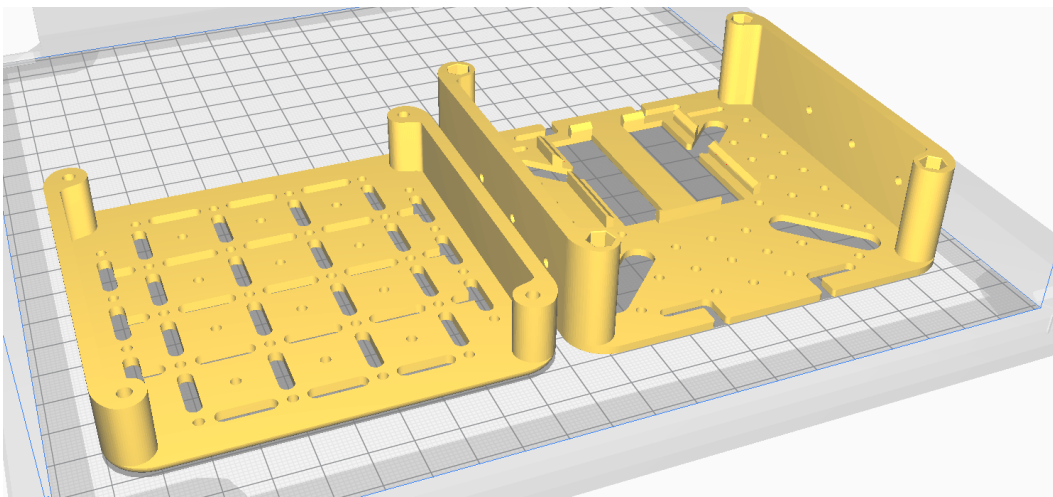


Figure 3: Fichiers STL des pièces PLA opaques

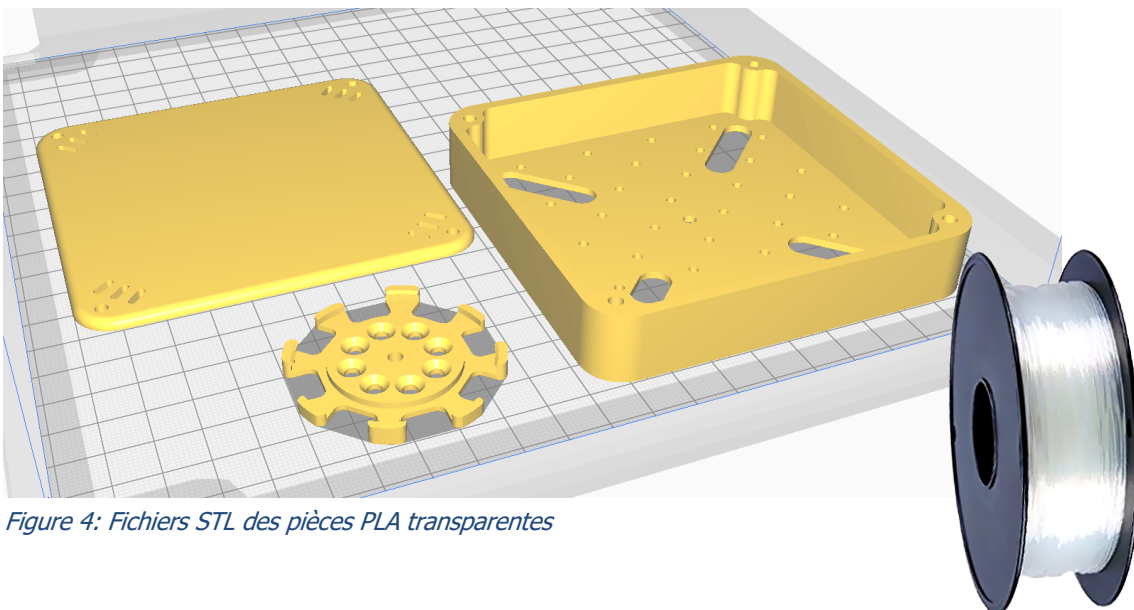


Figure 4: Fichiers STL des pièces PLA transparentes

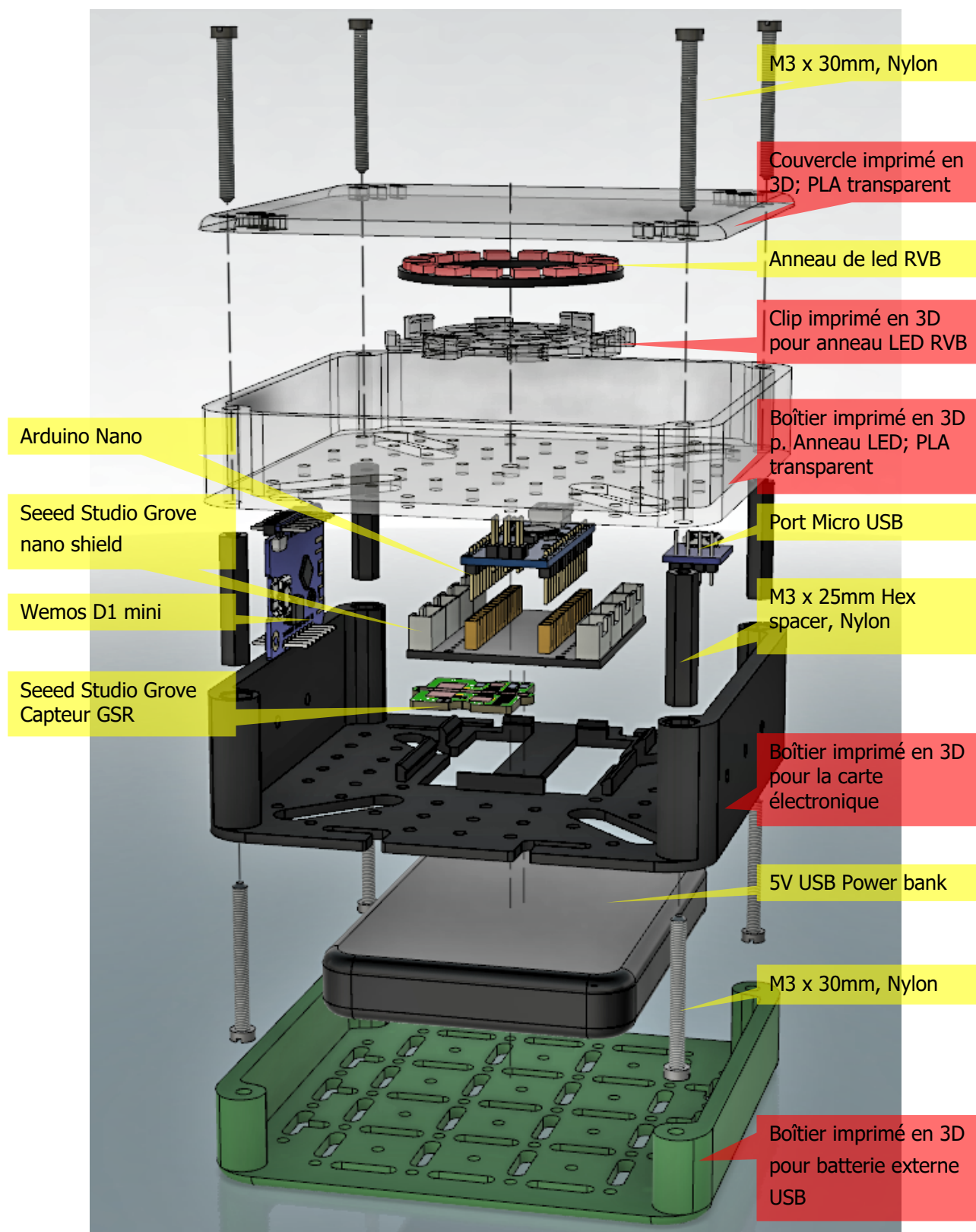


Figure 5: Dessin d'assemblage éclaté du cube

Composants Electroniques :

ATTENTION: Puisque nous faisons des expériences avec le corps humain, toutes les précautions doivent être prises! Ne connectez jamais un corps humain au réseau électrique domestique.

Le corps humain doit toujours être tenu complètement hors du réseau électrique !

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

9. Configuration des composants

Dans ce travail, nous recommandons le système Seeed Grove comme base pour sa facilité d'utilisation: (http://wiki.seeedstudio.com/Grove_System/) :

Composants Seeed Studio :

1x Grove Shield pour Arduino Nano

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

1x Grove GSR

http://wiki.seeedstudio.com/Grove-GSR_Sensor/

4 x Cable Seeed Studio :

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>

Microcontrôleurs:

1x Arduino Nano (ou équivalent)

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

1x Wemos LOLIN D1 mini (or equivalent)

https://www.wemos.cc/en/latest/d1/d1_mini.html

Pièces électroniques :

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&aq=chrome..69i57.3497j0j7&sourceid=chrome&ie=UTF-8>

Pièces diverses :

- 4x Entretoises en nylon M3 25 mm (entretoise hexagonale)
- 8x Vis en Nylon M3 30mm
- Entretoises en nylon M2 (entretoise hexagonale) pour Grove (a des trous de 2 mm)
- Petite banque d'alimentation USB, max. taille 15 mm x 100 mm x 60 mm
- Petit câble micro USB pour connecter la banque d'alimentation avec Nano
- Un fer à souder pour attacher les câbles aux composants électroniques



Des ordinateurs avec les logiciels suivants préinstallés :

- Autodesk Fusion 360 (ou tout autre logiciel de modélisation 3D, par ex. Wings3D)
- Le trancheur CURA,
- Une connexion Internet pour télécharger les bibliothèques
- L'IDE Arduino

Figure 6: M2 Nylon Standoffs

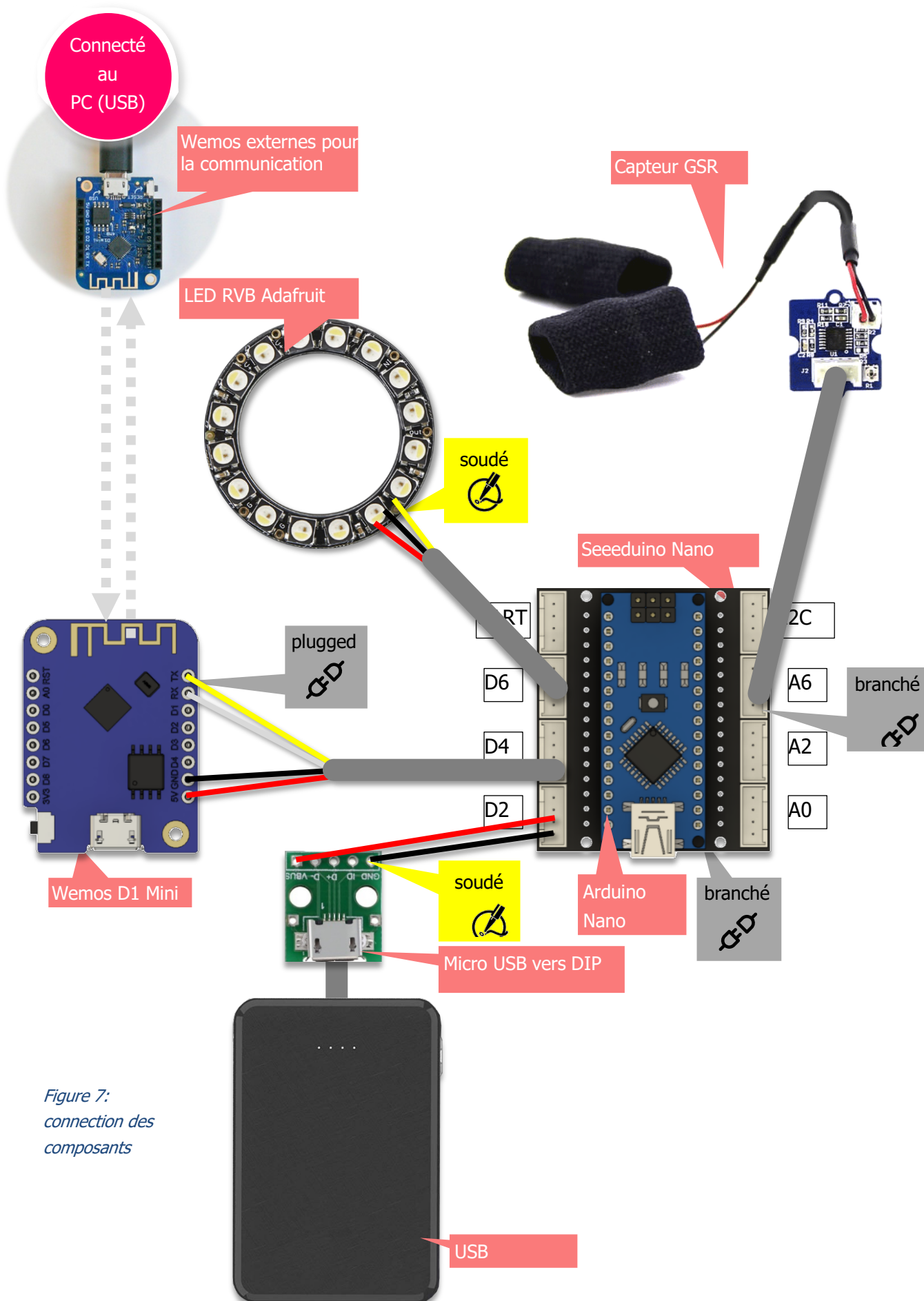


Figure 7:
connection des
composants

Bibliothèques Arduino pour les composants :

Le Wemos D1 Mini a besoin de bibliothèques pour que l'IDE Arduino fonctionne correctement.
L'importation d'une bibliothèque est décrite ici : <https://www.arduino.cc/en/Guide/Libraries>

Neopixel (Adafruit):

https://github.com/adafruit/Adafruit_NeoPixel/archive/master.zip

URL de préférence pour les cartes WEMOS (ESP8266):

Pour installer les wemos, un gestionnaire de carte doit être installé. La procédure est décrite ici :

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

1. Ouvrez la fenêtre des préférences dans l'IDE Arduino.
2. Entrez l'URL suivante dans le champ «Gestionnaire de carte additionnel»:
http://arduino.esp8266.com/stable/package_esp8266com_index.json
3. Ouvrez le gestionnaire de cartes à partir du menu Outils> Tableau et recherchez la plateforme esp8266.
4. 4. Sélectionnez la version actuelle dans la liste déroulante et cliquez sur le bouton « installer ».
5. 5. Sélectionnez « (LOLIN) Wemos D1 R2 et Mini » dans le menu Outils> Tableau après l'installation.

Le capteur Grove GSR n'a besoin d'aucune bibliothèque car il peut être contrôlé avec de simples commandes d'entrée analogique Arduino.

Wemos D1 mini comme connexion sans fil entre composants électroniques

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

Deux Wemos construisent une paire qui est connectée via le port Ethernet Wi-Fi 23 (Telnet). Le seul 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 système d'alimentation domestique, une connexion sans fil doit être établie.

Par conséquent, la communication série habituelle (UART) est traduite en Wifi et envoyée par un Wemos, reçue par les autres Wemos et retraduite en communication série. Pour des raisons de compatibilité, le débit en bauds est fixé à 9600 bauds, car la communication série logicielle 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 existe un canal libre inutilisé ou un réseau faible en arrière-plan,
3. Établissement d'un point d'accès Wifi en utilisant le premier canal disponible, également combiné avec DHCP
4. En attente d'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 au point 1. (analyse des réseaux)

Le client doit être démarré en deuxième et se connectera automatiquement.

Comment configurer le serveur et le client Wemos, comme expliqué dans « Better Server source code » :

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



Figure 8: extrait du code source du serveur



Figure 9: extrait du code source du client

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

Change the **IP-adress** to

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

Change the **ssid** to

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

... compile the scripts inside the Arduino IDE and upload them to the appropriate Wemos boards.

8c Une théorie du biofeedback

Ce texte se veut un bref aperçu et peut être considéré comme un ensemble de mots-clés importants. Il ne se veut pas un manuel !

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

La réaction d'un corps humain au stress ou aux influences extérieures se produit la plupart du temps automatiquement et inconsciemment. Par exemple, si un humain ment ou a peur, sa peau commence à transpirer. Cette sueur peut être mesurée comme un changement de conductivité électrique puisque 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 dans l'esprit de cette personne.

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

- Influence de la fréquence cardiaque avec changement de fréquence respiratoire, surveillée par des capteurs de pouls
- Influencer les réactions de peur avec un changement d'activité électrodermique, surveillé par des capteurs GSR (Une réaction de peur pourrait être provoquée par 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 une contraction du muscle trapèze dans le cou de l'homme. Cela peut être mesuré avec EMG.

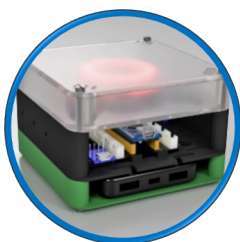
9. Plan de leçon: description étape par étape de l'activité / du contenu

Leçons 1 & 2 (90min) :

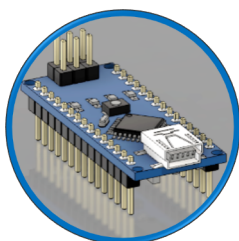


Les étudiants découvriront l'IoT par des exemples: des robots aspirateurs avec télécommandes d'applications, des stations météorologiques basées sur Internet, une agriculture intelligente et enfin des applications de santé. Les étudiants doivent examiner le fonctionnement de ces appareils et les composants 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, les menaces et les limites doivent être discutées: où l'IoT a-t-il un sens et où pas ?

Leçons 3 & 4 (90min) :



Impression 3D et assemblage de l'appareil: en option, les étudiants peuvent imprimer en 3D le boîtier de l'appareil, puis ils doivent connecter eux-mêmes toutes les pièces électroniques. Les étudiants doivent développer une compréhension plus approfondie de la façon dont les pièces s'assemblent et donc construire un appareil complet. À quoi sert un anneau LED, à quoi sert la connexion Wi-Fi? Comment un signal est-il traité de son origine à la fin - jusqu'à l'utilisateur? À partir d'un signal biologique converti en un signal analogique à l'intérieur de l'appareil, en le convertissant en un signal numérique à l'intérieur du CAN du microcontrôleur, en traitant les données à l'aide du logiciel et en communiquant les résultats via une lumière LED ou en transmettant des informations via Wi-Fi.

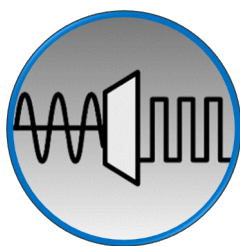


Leçons 5 & 6 (90min) :

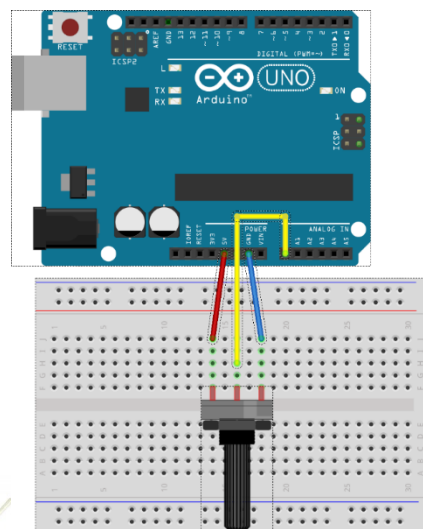
Introduction de la programmation Arduino: connexion Arduino IDE et configuration de la communication d'Arduino Nano avec l'IDE / l'ordinateur. La structure fondamentale de la plate-forme Arduino doit être expliquée: que sont les broches d'entrée / sortie à usage général (GPIO), qu'est-ce que la logique numérique et quelle est la différence entre l'entrée et la sortie numériques. Les scripts simples sont écrits et modifiés à l'aide des exemples de base fournis avec l'IDE Arduino :

```
"01.Basics → Blink",
"01.Basics → DigitalReadSerial",
"04.Communications → SerialEvent",
```

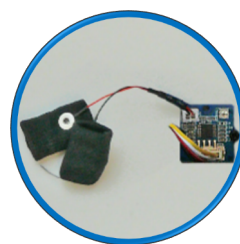
Leçons 7 & 8 (90min) :



Comment un signal analogique est-il traité par une machine numérique? La théorie de la conversion du signal analogique-numérique est introduite et peut être étudiée en utilisant une configuration simple de diviseur de tension. En utilisant l'Arduino, n'importe quel capteur simple basé sur une résistance peut être utilisé pour construire un circuit simple, par ex. un LDR, une thermorésistance ou même un simple potentiomètre.



Leçons 9 & 10 (90min) :

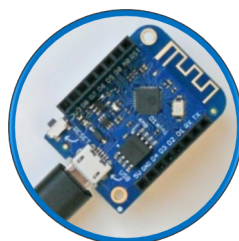


Présentation du GSR-Sensor: apprentissage de l'amplification du signal à l'aide d'amplificateurs opérationnels, mesure de l'humidité de la peau à basse tension. Comment mesurer la résistance de la peau humaine ? Quelle tension est inoffensive pour l'utilisateur ?

Contexte biologique de l'activité électrodermique et sa signification / interprétation https://en.wikipedia.org/wiki/Electrodermal_activity

La programmation du capteur gsr peut être réalisée en utilisant et en adaptant les scripts Arduino pré-construits des leçons précédentes.

Leçons 10 & 11 (90min) :



Présentation du Wemos D1 Mini: Cette carte microcontrôleur peut être utilisée comme un dispositif de transmission sans fil qui complète le caractère IoT du GSR-Cube. Deux Wemos sont nécessaires pour connecter deux nœuds différents - un nœud est le Cube lui-même et l'autre nœud est par exemple l'ordinateur d'un élève vers lequel tous les signaux traités sont transférés.

Étant donné que la technologie de réseau Wi-Fi de Wemos est très compliquée et nécessite donc une unité d'enseignement séparée, tous les appareils Wemos doivent être préinstallés et configurés avec soin avant le cours. Cela doit être fait par l'enseignant.

Il est important de souligner la possibilité d'abus : il existe de nombreux aspects de la protection de la vie privée qui peuvent être facilement illustrés avec cette configuration. Quelles sont les conséquences du transfert quelque part des données collectées (données personnelles) conduisant à une perte de contrôle de leur sphère privée ?

Leçons 13 & fin :



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

Cette leçon est basée sur des réactions émotionnelles lors du visionnage d'un « film d'horreur » : les montées et les descentes d'une montagne russe peuvent avoir un effet énorme sur les sentiments de la personne testée. Comment l'influencer ?

Et quelques 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 ?

Et maintenant: Programmation Freestyle et test des capacités du cube EDA ! 😊

Essayez de faire des expériences avec vos camarades de classe. Faites un tour de discussion et gardez une trace de la sortie de votre cube EDA. Pouvez-vous reconnaître ce que ressent votre collègue ?

10. Retour d'information

À la fin de la leçon, les étudiants doivent avoir une connaissance approfondie du fonctionnement des principes de l'IoT dans les dispositifs médicaux et de la manière dont le cube EDA 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.

11. Évaluation

Les élèves tiennent leur journal du travail, qui peut être révisé par l'enseignant. Les élèves peuvent également présenter les résultats de leurs expériences. De plus, un test standard en classe doit être effectué à la fin des cours.

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

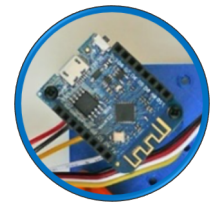
void setup() {
    Serial.begin(9600);
    WiFi.setSleepMode(WIFI_NONE_SLEEP);
    WiFi.mode(WIFI_STA);
    WiFi.setOutputPower(10); // 10: 10mW, 14: 25mW, 17: 50mW, 20: 100mW
    WiFi.begin(ssid, password);

    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()) {
        char c = client.read();
        Serial.print(c);
    }
    while (Serial.available() > 0) {
        inChar = Serial.read();
        if (client.connected()) {
            client.write(inChar);
            delay(1);
        }
    }
}
```

Code source client Wemos

*Figure 10: Code
source du client
Wemos lié au cube
EDA*



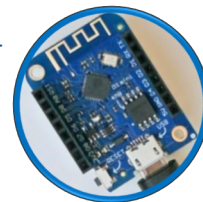
Code source du serveur Wemos

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

Figure 11: Ce script doit
être compilé pour le
Wemos connecté au PC



```
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;
        }
    }
}
```

Code Arduino Nano pour le cube EDA

```
#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);
}
```

*Figure 12: un exemple de
code source fonctionnel
pour le cube EDA pendant
toutes les leçons*

