Co-funded by the Erasmus+ Programme of the European Union

‘We are the makers - IoT’ Learning Scenario:
Biofeedback cu senzori IoT pentru monitorizarea stării de sănătate

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

Această lucrare a fost dezvoltată și testată în școală cu circa 18 elevi cu vârste cuprinse între 13 și 17 ani, în anul școlar 2018/2019. Ea reflectă experiența avută cu numeroase meandre și unele eșecuri. Deoarece domeniul IoT este unul foarte complex, materialele de predare trebuie alese cu mare grijă. Această lucrare se dorește o recomandare, un punct de plecare.

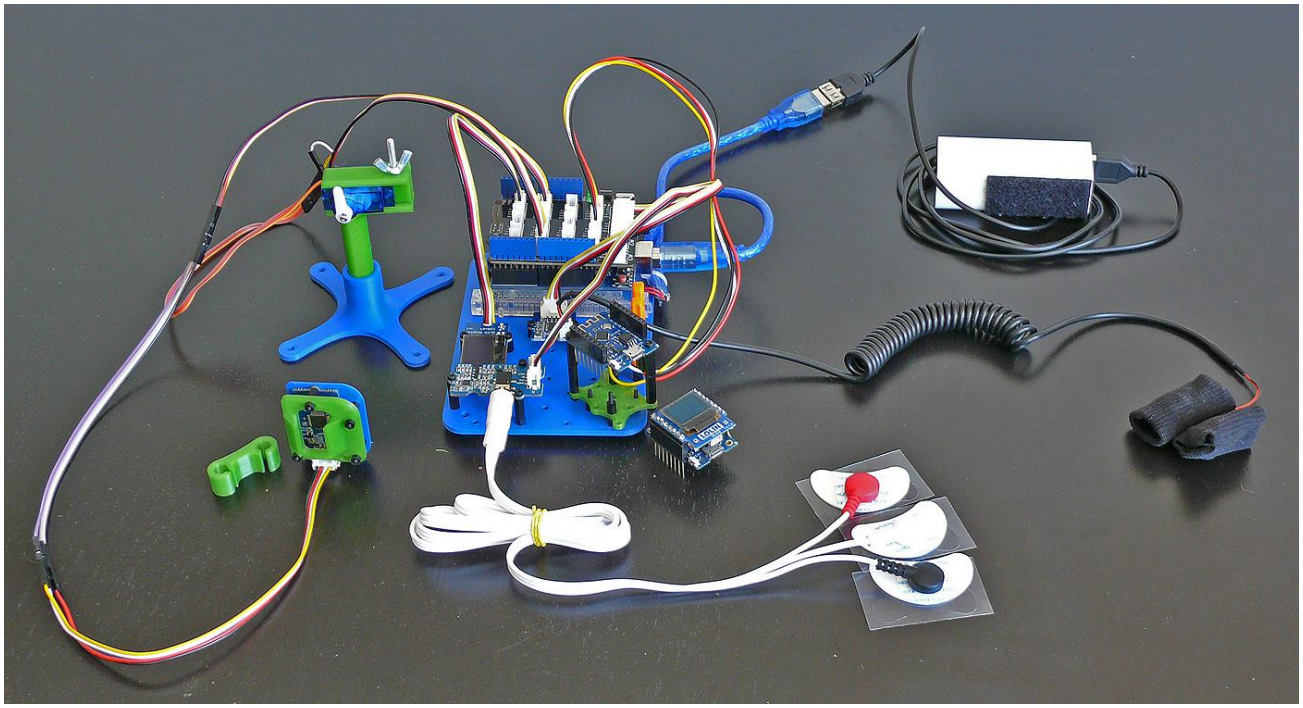
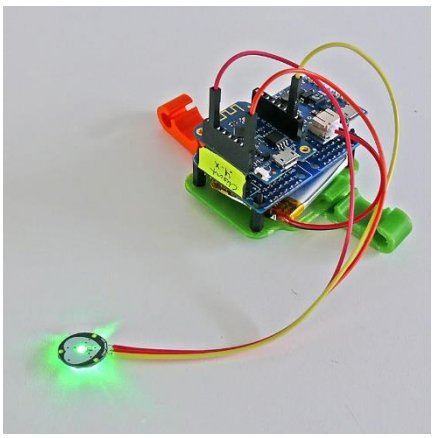


Figura 1: Prototipul unei stații de biofeedback IoT



Ce spuneți de un detector de minciuni făcut de noi sau de un dispozitiv de monitorizare a activității fizice cu un soft de control pe care să-l programăm noi? Sau de un pulsometru care să ne ajute să ne relaxăm și să urmărim efortul?
Ce spuneți de un dispozitiv care să ne ofere feedback cu privire la tonusul muscular și care să ne ajute să ne controlăm în situațiile de stres?
Controlul comportamentului unei mașini cu puterea gândului – e posibil sau nu?
Și: ce spuneți de dispozitivele de monitorizare a somnului care ne ajută să ne optimizăm somnul și informează un medic atunci când nu suntem bine? Care sunt oportunitățile și amenințările tehnologiei moderne cu biosenzori IoT?

Număr proiect: 2017-1-DE03-KA201-035615

1. Titlu	**Biofeedback cu senzori IoT pentru monitorizarea stării de sănătate**
2. Grup țintă	14 - 17 ani
3. Durată	Minim 5 săptămâni cu câte 2*45 min - lecții pe săptămână: în total circa 6-8 ore.
4. Nevoile de învățare	▪ Interacțiunea dintre componentele electronice și părțile corpului ▪ Monitorizarea și afectarea parametrilor biologici umani ▪ Lanțul de comunicare al dispozitivelor IoT ▪ Principiile senzorilor și actuatoarelor ▪ Diferite principii de măsurare a semnalelor biologice ▪ EMG I: Cum funcționează sistemul muscular? ▪ EMG II: Principiile amplificatoarelor (Instrumentation Amplifiers) ▪ Principiile rețelelor de comunicații fără fir ▪ Construcția și imprimarea 3D a unor componente care să ajute în realizarea de măsurători
5. Rezultatele învățării	▪ Cum funcționează un sistem IoT? ▪ Unde sunt posibilitățile și limitările sistemelor IoT pentru sănătate? ▪ Ce componente - hard și software - sunt esențiale pentru construirea unui dispozitiv IoT? ▪ Cum se furnizează biofeedback (feedback biologic) pentru a ajuta oamenii?
6. Metodologie	În acest scenariu, elevii vor construi și programa singuri un dispozitiv de semnale bio interactiv. Elevii vor folosi Arduino Serial Monitor și Serial plotter pentru vizualizarea și reprezentarea grafică a feedback-ului biologic.
7. Locație / Mediu	▪ un laborator cu un set de piese și componente electronice; ▪ fiecare grup de elevi trebuie să aibă un calculator sau laptop cu drepturi de administrator pentru instalarea diferitelor pachete software; ▪ Proiector pentru prezentarea pașilor și prezentarea lucrărilor elevilor; ▪ fiecare elev trebuie să țină un jurnal de laborator

8. Unelte/ Materiale/ Resurse

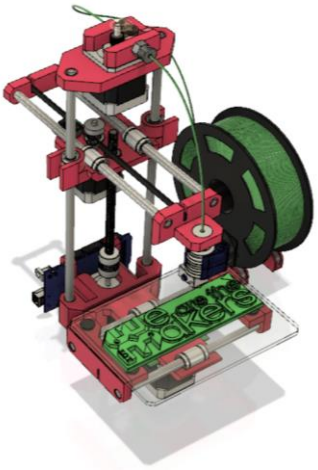


Figura 2:Simbolul unui imprimante 3D

Imprimante 3D

Sunt necesare circa 3-4 imprimante 3D întrucât elevii își vor tipări singuri sistemul IoT de biofeedback. Desigur, elevii ar putea să și proiecteze componentele.

Componente tipărite 3d:

Ca punct de plecare, toate părțile necesare sunt date în format .stl și ca fișiere Autodesk Fusion 360.

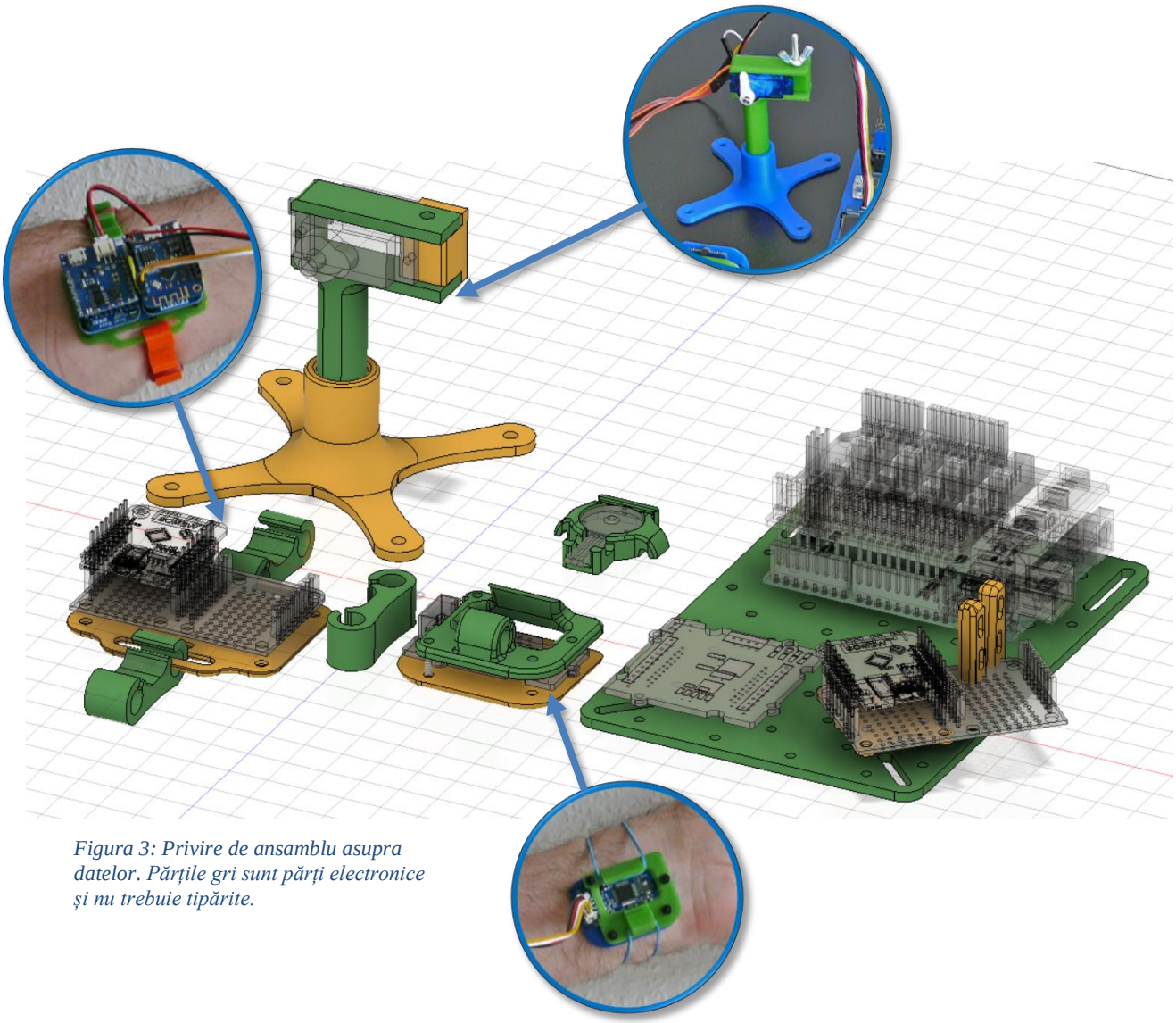


Figura 3: Privire de ansamblu asupra datelor. Părțile gri sunt părți electronice și nu trebuie tipărite.

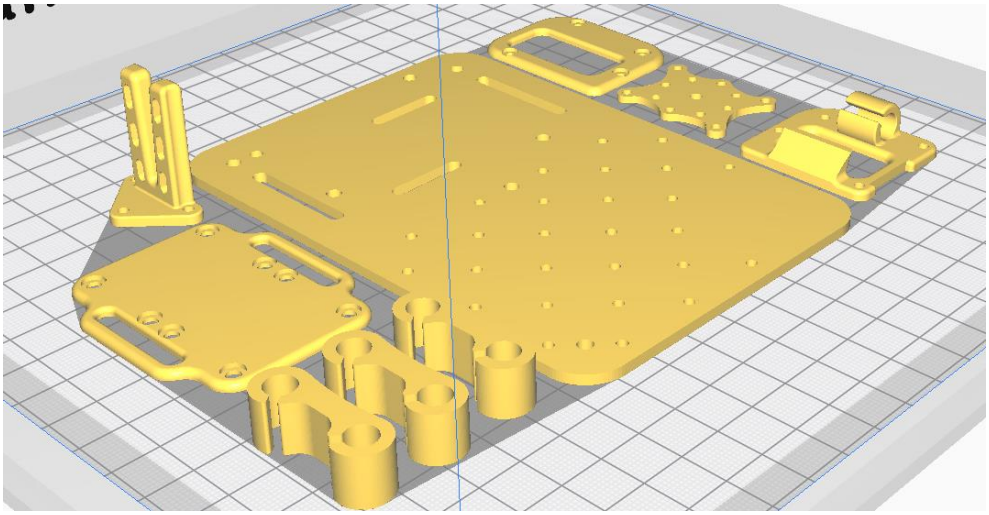
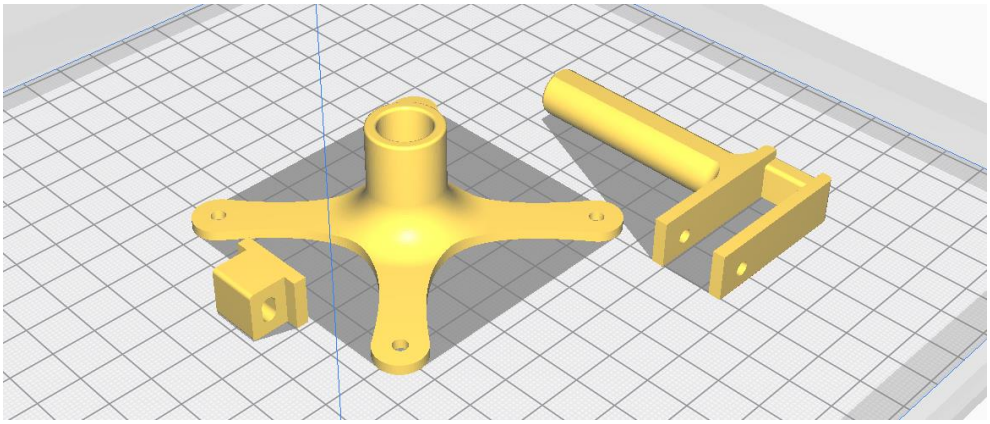


Figura 4: Setul de fișiere .stl pregătite pentru tipărire



Componente electronice:

ATENȚIE: Deoarece se vor realiza experimente asupra corpului uman, trebuie luate toate măsurile de precauție! Niciodată nu conectați un corp uman la rețeaua de alimentare.

Avertismentul se referă și la adaptoarele AC care sunt conectate la priză. Acest tip de circuite trebuie evitate. Se vor folosi doar baterii cu tensiuni mici de circa 3-5V.

În această lucrare, se recomandă sistemul Seeed Grove ca bază deoarece e ușor de folosit (http://wiki.seeedstudio.com/Grove_System/). Toate componentele principale, cu excepția chip-urilor Wemos, acumulatorilor și senzorilor de ritm cardiac, aparțin pachetului Grove standard:

Componente Seeed Studio:

1: Grove Base Shield pentru Arduino-Uno (http://wiki.seeedstudio.com/Base_Shield_V2/)

2. Grove OLED 128x64 (http://wiki.seeedstudio.com/Grove-OLED_Display_0.96inch/)

3. Grove EMG Detector (http://wiki.seeedstudio.com/Grove-EMG_Detector/)

4. Clește de ritm cardiac pentru deget Grove http://wiki.seeedstudio.com/Grove-Finger-clip_Heart_Rate_Sensor/

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

Senzori și actuatoare:

1: Arduino Uno (sau echivalent)

A: **2x** Wemos LOLIN D1 mini (sau echivalent)

B: Senzor analog de puls pentru deget (www.pulsesensor.com)

C: Wemos battery shield (https://wiki.wemos.cc/products:d1_mini_shields:battery_shield)

D: Micro servo motor.

Componente diverse:

- Bandă
- Bandă de cupru auto-adezivă
- Distanțiere Nylon M3 (hexagonale)
- Distanțiere Nylon M2 (hexagonale) pentru Grove (au găuri de 2mm)
- Fire Grove
- Piulițe WAGO
- Fire cu Jumper
- Sursă de tensiune USB pentru încărcarea acumulatorului extern USB
- Acumulator extern USB
- Ciocan de lipit pentru lipirea cablurilor la componentele electronice

Figura 5: Distanțiere Nylon

calculatoare cu următoarele softuri preinstalate:

- Autodesk Fusion 360 (sau orice alt software de modelare 3D, ex. Wings3D)
- CURA,
- O conexiune la internet pentru descărcarea bibliotecilor
- Arduino IDE
- Processing IDE

Număr proiect: 2017-1-DE03-KA201-035615

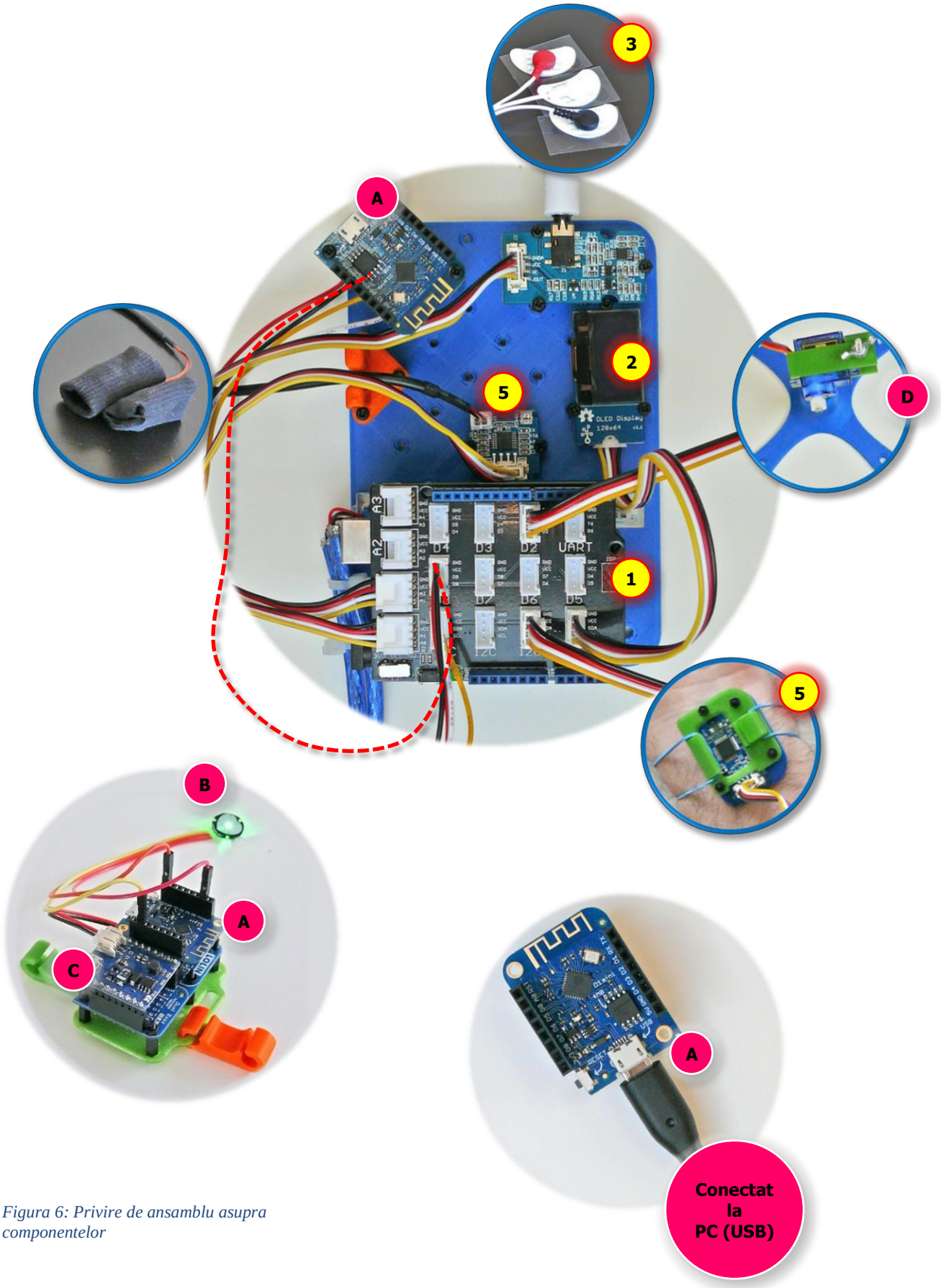


Figura 6: Privire de ansamblu asupra componentelor

Biblioteci Arduino pentru componente:

Unele componente cum ar fi Wemos D1 Mini sau unele plăci Grove necesită anumite biblioteci pentru Arduino IDE pentru a funcționa corespunzător. Modul în care se pot importa biblioteci e prezentat aici: https://www.arduino.cc/en/Guide/Libraries

OLED lib (Seeed):
https://github.com/Seeed-Studio/OLED_Display_128X64/archive/master.zip

Software I2C Master (Felias Fogg):
https://github.com/felias-fogg/SoftI2CMaster

Opțiunile pentru plăcile WEMOS (ESP8266):
Pentru a instala wemos, trebuie instalată așa numita "board-definition" (definiția plăcii). Descrierea se găsește la adresa:
http://arduino.esp8266.com/Arduino/versions/2.0.0/doc/installing.html

1. În Arduino IDE deschideți fereastra *Preferences*.
2. Introduceți următoarea adresă în câmpul "Additional Board Manager": http://arduino.esp8266.com/stable/package_esp8266com_index.json
3. Deschideți *Boards Manager* din *Tools > Board* și căutați platforma esp8266.
4. Selectați versiunea curentă din caseta derulantă și executați clic pe butonul "install".
5. Selectați "(LOLIN)Wemos D1 R2 and Mini" din *Tools > Board* după instalare.

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

Grove GSR, clește de ritm cardiac pentru deget Grove și detectorul Grove EMG nu necesită nicio bibliotecă, deoarece pot fi controlate prin comenzi Arduino simple.

Wemos D1 mini – conexiune wireless între componentele electronice

- **Plăcile Wemos trebuie să fie pregătite de către profesor, nu de către elevi, înainte de începerea lecției!**
- Plăcile Wifi Wemos-ESP8266 sunt o alternativă mai ieftină a tehnologiei Xbee.

Două Wemos formează o pereche care este conectată prin portul Wifi ethernet 23 (Telnet). Singurul scop este de a **înlocui comunicarea serială prin cablu**. De obicei un dispozitiv electronic experimental este conectat prin cablu USB la PC. Pentru a crea un sistem complet autonom care nu este conectat la rețeaua de alimentare cu energie electrică, e necesară stabilirea unei conexiuni wireless.

Astfel, comunicarea serială (UART) este tradusă în Wifi și trimisă de un Wemos și recepționată de celălalt Wemos și re-tradusă în comunicare serială. Pentru compatibilitate, rata de transfer (baud rate) e fixată la 9600 baud, deoarece comunicarea serială software a unui Arduino Uno este limitat la 9600 baud.

O pereche Wemos D1 mini constă într-un server și un client. Serverul trebuie să fie conectat la un PC. Acesta ar trebui pornit primul și realizează următorii pași:

1. Scanează toate rețelele wifi disponibile,
2. Determină dacă există un canal nefolosit sau o rețea slabă
3. Stabilește un punct de acces Wifi folosind primul canal disponibil găsit, combinat cu DHCP
4. Așteaptă UN (doar unul!) Client care să se conecteze.
5. Dacă acest Client se deconectează, va aștepta până acesta se reconectează.
6. Dacă Serverul e resetat se reiau pașii de la 1 (scanarea rețelelor)

Clientul trebuie pornit al doilea și se va conecta și reconecta automat.

Cum se configurează Wemos Server și Wemos Client, explicat în "Better Server sourcecode":

În continuare sunt prezentate cele mai importante excepții din codurile sursă ale serverului și clientului ce trebuie adaptate pentru configurarea perechilor individuale de plăci Wemos:

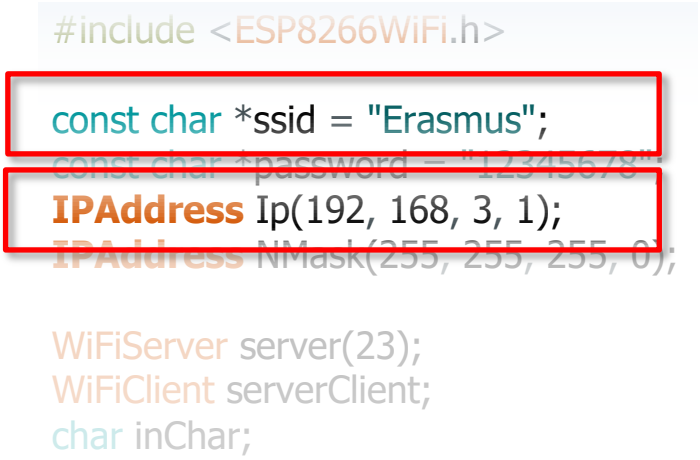


Figura 7: tăierea codului sursă a serverului

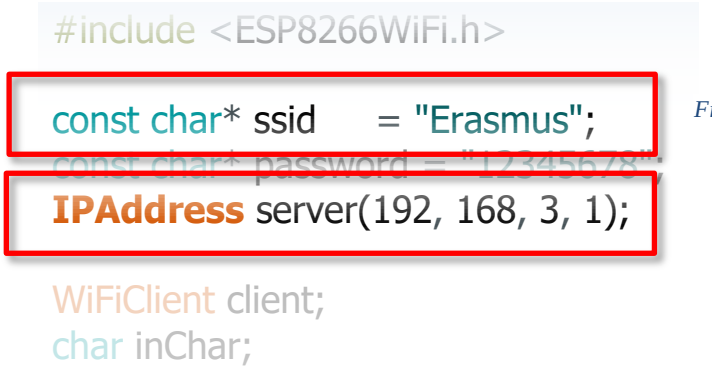


Figura 8: tăierea codului sursă a clientului

- Ambele linii de cod subliniate trebuie să fie exact la fel pentru o pereche Wemos.
- Ambele linii de cod subliniate trebuie să fie adaptate pentru fiecare pereche Wemos.

Schimbați IPAddress cu
192.168.1.1 sau 192.168.2.1 sau 192.168.4.1 sau 192.168.5.1 …etc.

Schimbați **ssid** cu
"Erasmus1" sau "Erasmus2" sau "Erasmus4" sau "Erasmus5" …etc.

… compilați scripturile în Arduino IDE și încărcați-le pe plăcile Wemos corespunzătoare.

8b Teorie despre Potențialul de acțiune și măsurarea EMG

Acest text este o prezentare generală și poate fi considerată o colecție de termeni cheie. Nu este un manual!

Următorul articol se bazează pe "EMG Fibel V1.1.pdf", și descrierile Wikipedia pentru "Action Potential", "Design of an EMG Detector", "EMG" din 'electronic compendium":

- 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-kompendium.de/public/schaerer/emg1.htm

Mușchii se contractă deoarece primesc semnale electrice de la nervi. Așa numitul "Potențial de acțiune" este o schimbare a potențialului electric a terminațiilor nervoase intramusculare. Schimbarea are loc între -80mV ("Resting Potential" – potențialul de repaus) și +30mV ("Depolarisation" - depolarizarea), adică, în teorie, o schimbare de potențial de 110mV. Terminațiile nervoase se găsesc în fibrele musculare. Noi (la școală) putem realiza măsurătiori doar la nivelul pielii. În consecință, o parte din puterea semnalului se pierde datorită rezistenței electrice a pielii și țesuturilor. O schimbare în potențialul electric măsurabilă are de obicei circa 30mV.

Microcontrolerele capabile să proceseze aceste semnale folosesc convertoare analog-digitale. Aceste convertoare AD au, de obicei, o tensiune de intrare în intervalul 0-3.3V (tipul Wemos) sau 0-5V (tipul Arduino Uno). Rezoluția ambelor convertoare AD este de 10 biți, ceea ce înseamnă că microcontrolerele pot împărți intervalul de măsurare de 3.3V în 1024 valori discrete: $3.3V / 2^{10} = 3300mV / 1024 = 3,2mV$. Dacă am măsura schimbrile de potențial la nivelul pielii cu un singur microcontroler, am putea obține doar un domeniu de 10 valori din 1024 valori posibile. Este mult prea puțin. În plus, potențialul pielii se modifică sub influența câmpurilor electrice externe, ceea ce conduce la devieri de 1000 de ori mai mari decât puterea semnalului.

Prin urmare, avem nevoie de o componentă electronică plasată înainte care: a) amplifică semnalul nostru de la 0.03V la 3.3V și b) e capabilă dă compenseze deviațiile câmpului electric. Așa numitul amplificator instrumental este un circuit cu trei intrări: o intrare pentru '+', una pentru '-' și una pentru referință. În timp ce electrozii '+' și '-' au ca scop măsurarea diferenței de potențial de 0.03V, electrodul referință vizează influența câmpurilor electrice externe pentru a compensa devierea.
În interiorul acestui instrument electronic de precizie, semnalul este amplificat.
Acum putem măsura și digitiza potențialul de ieșire cu microcontrolerul.

Electrozii trebuie plasați cu atenție, deoarece trebuie să măsoare potențialul nervilor. Cu cât distanța până la terminațiile nervoase e mai mică, cu atât semnalul este mai bun. Un semnal de la nervi trece prin mușchi cu o viteză de circa 5m/s. Dacă măsurătoarea se realizează la mijlocul mușchiului biceps, electrozii + și – ar trebui plasați la circa 10cm unul față de celălalt, iar electrodul de referință ar trebui plasat într-un punct cât mai departe de ceilalți electrozi (ex. pe mână). Timpul de tranzit a semnalului de la un electrod la celălalt este de ("unda de depolarizare"):

$$\frac{distanța\ dintre\ electrozi\ în\ cm}{\frac{500\ cm}{secundă}} = 20\ milisecunde$$

În măsurarea EMG la nivelul pielii, vom vedea o suprapunere de mai multe semnale de la mai multe fibre musculare diferite și deci o formă de undă de 20ms va fi cu greu recunoscută. Pentru a contracta un mușchi cu putere variabilă, valoarea potențialului de acțiune nu se modifică. Există doar o modificare a apariției modificărilor de tensiune ale nervilor. Cu cât un mușchi este contractat mai mult cu aât este mai mare 'rata de foc' a potențialelor de acțiune ale nervilor.

8c Teorie despre Biofeedback

Acest text este o prezentare generală și poate fi considerată o colecție de termeni cheie. Nu este un manual!

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

Răspunsul corpului uman la stres sau influențe externe are loc, de cele mai multe ori, automat și inconștient. Spre exemplu, dacă o persoană minte sau simte frică, începe să transpire. Această transpirație poate fi măsurată ca o schimbare a conductivității electrice, deoarece transpirația conține electroliți. Dacă un calculator reprezintă vizual această schimbare, persoana poate corela starea sa emoțională cu semnalul măsurat și poate încerca să își influențeze reacția sau să învețe cum să își controleze emoțiile. În acest fel persoana poate deveni conștientă de unele emoții pe care nu le sesiza anterior.

Există mai multe exemple și experimente pe care elevii le pot testa singuri:

- Să influențeze pulsul prin schimbarea modului în care respiră, monitorizat cu senzori de puls. Să influențeze reacțiile la frică cu schimbarea activității electrodermale, monitorizat cu senzori GSR (reacția se poate obține: cu o imagine a unui păianjen, un film youtube cu un roller coaster)
- Un poligraf (detector de minciuni) este – printre altele – bazat pe schimbarea activității electrodermale ce poate fi măsurată cu senzori GSR
- Coactivarea mușchilor: Tastarea la calculator în condiții de stres conduce la contractarea mușchiului trapez, la nivelul gâtului. Acest lucru poate fi măsurat cu EMG.

9. Plan de lecție: Descrierea pa cu pas c activității/conținutului

Lecțiile 1 & 2 (90min):

Elevilor li se va prezenta conceptul de IoT prin exemple: roboți de aspirare ce pot fi controlați de la distanță prin intermediul unei aplicații, stații meteo bazate pe web, agricultura inteligentă și la sfârșit aplicațiile de sănătate. Elevii trebuie să examineze modul în care funcționează aceste dispozitive și ce componente sunt necesare pentru crearea acestora: un sistem bazat pe microcontroler controlează și coordonează o serie de senzori și actuatoare. În plus comunică și coordonează cu alte sisteme asemănătoare, de obicei prin rețele de comunicare wireless. Componente necesare: senzori, actuatoare, dispozitive de comunicare. Se va discuta despre posibilități, amenințări și limitări: unde ar trebui și unde nu ar trebui folosit IoT.

Lecțiile 3&4 (90 min)

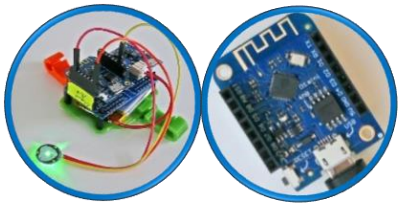


Figura 8: Componente necesare

Elemente de bază privind biosenzorul (senzor de puls): Elevilor le este prezentat primul senzor care nu trebuie să fie izolat galvanic: senzorul de puls. El este format din două părți diferite: un LED verde simplu și un fototranzistor, plus un circuit amplificator direct atașat. Trebuie să fie plasat direct deasupra unei vene, de ex. vârfurile degetelor sau vârfurile urechii.

Dacă vena este umflată deoarece inima pompează, sângele va reflecta lumina verde, iar fototranzistorul va măsura o valoare ridicată. Dacă vena se contractă, când inima se contractă, sângele lipsă va conduce la o absorbție mai mare a luminii în țesutul conjunctiv. Fototransistorul va măsura o valoare mai mică.

Elevii ar trebui să fie introduși în programarea analogică Arduino și calculul valorilor analogice: trebuie prezentat convertorul analog-digital (ADC) și cum funcționează acesta. O introducere foarte bună poate fi găsită la adresa:

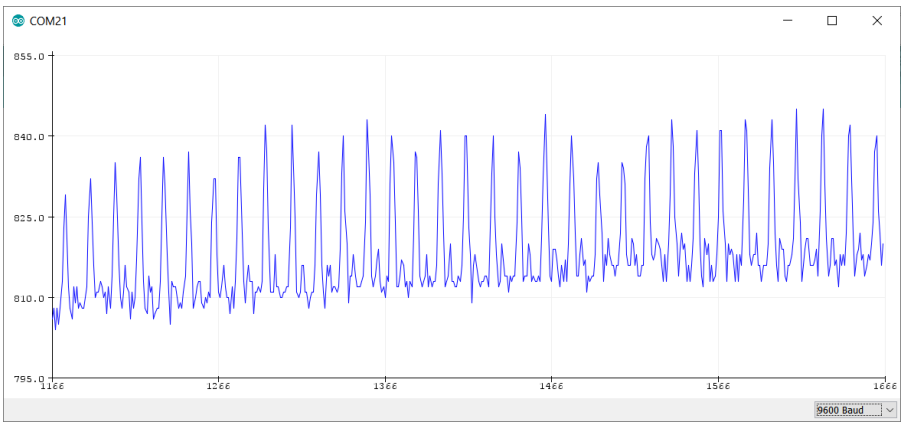
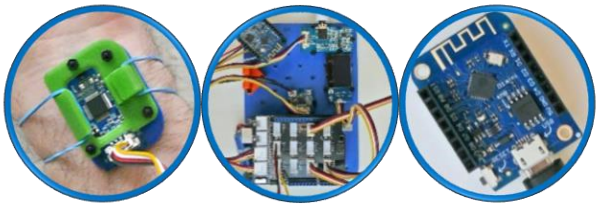


Figura 9: Date eșantion măsurate de senzorul de puls

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

Lecțiile 5&6 (90 min)



Biofeedback-ul ritmului inimii: Elevii învață că multe reacții corporale corespund unor procese emoționale inconștiente. Dar dacă vom avea acces la aceste reacții ale corpului, putem înecrca să le controlăm: biofeedback-ul este o monitorizare în timp real a răspunsurilor fizice cu scopul de a încerca controlul emoțiilor (https://www.artofmanliness.com/articles/hack-your-mind-like-a-twenty-first-century-soldier-using-biofeedback-to-become-more-resilient/).

Dacă software-ul senzorului de puls funcționează corect, elevii pot începe să măsoare modificările pulsului și să încerce să influențeze ritmul cardiac cu ajutorul respirației: Care este influența ritmului respirator asupra ritmului cardiac? Ce se întâmplă, dacă respirăm mai repede sau mai încet? Ce înseamnă acest lucru pentru situațiile de stres? Putem fi conștienți de ritmul nostru cardiac?

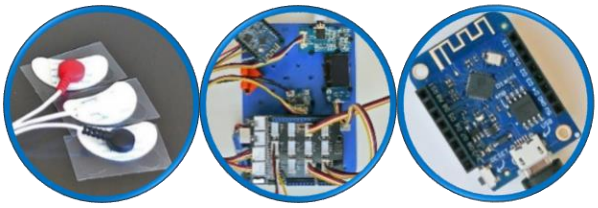
Utilizați această procedură de configurare pentru un experiment de biofeedback:

Figura 10: Captură din Processing App: influențarea luminozității/contrastului cu ajutorul pulsului

a) Atașați cleștele pentru măsurarea pulsului I2C Grove la degetul unui elev
b) Conectați stația Arduino cu PC și porniți codul sursă de procesare. Cu cât pulsul e mai ridicat cu atât imaginea va deveni mai întunecată: influențați luminozitatea/contrastul prin controlul mental al activității inimii.
c) Realizați următoarele:

- Inspirați încet timp de 4 secunde.
- Țineți-vă respirația timp de 4 secunde.
- Expirați încet timp de 4 secunde.
- Așteptați timp de 4 secunde fără să inspirați.
- Repetați până aveți respirația sub control. (https://en.wikipedia.org/wiki/Dave_Grossman_(author))

Lecțiile 7&8



Bazele EMG: Se predau cuvintele cheie și relațiile potențialului activ a nervilor, încordarea și relaxarea mușchilor și principiile de măsurare. Elevii trebuie să înțeleagă cum funcționează codul sursă care amplifică semnalul de la mușchi.

Cum se plasează cei trei electrozi pe braț: La început, persoana de test trebuie să-și spele brațul cu apă și săpun și apoi trebuie șterse cu o bucată de vată îmbibată în alcool zonele pe care vor fi plasate electrozii. Apoi pot fi plasați electrozii.

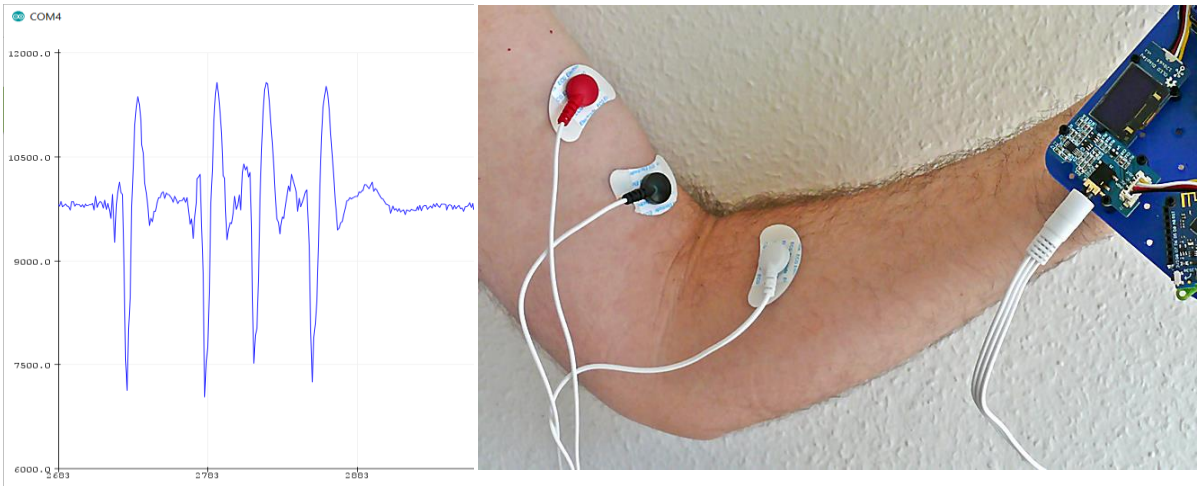
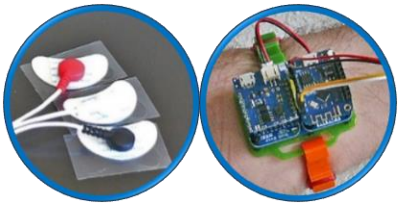


Figura 11: Activitatea la nivelul bicepșilor și plasarea corespunzătore a electrozilor

Elevii ar trebui să experimenteze prin mișcarea brațelor cu viteze diferite, prin ridicarea de diferite greutăți și efectul relaxării. Ce se întâmplă dacă - cu exact acest amplasament de electrozi - se va închide și deschide mâna? Ce se întâmplă dacă cineva scoate electrodul alb?

Lecțiile 9&10



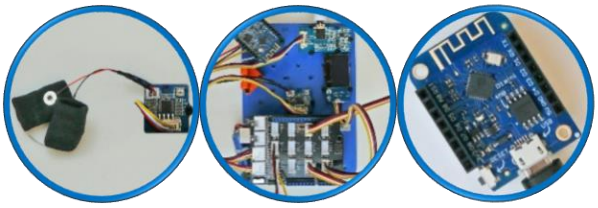
Experiment de biofeedback EMG:

Următorul experiment se bazează pe teza de doctorat a lui Michael Schnoz: https://www.research-collection.ethz.ch/handle/20.500.11850/149225

Figure 12: placement of electrodes in the neck

Dacă o persoană de test va tasta caractere la un calculator, la viteză mare și contrare maximă, probabil va avea dureri de gât ce pot fi măsurate și apoi influențate de persoana testată. Pentru a pune persoana de test în condiții de stres se poate folosi tutorialul:
https://portableapps.com/apps/education/tipp10_portable
Cu cât persoana va tasta mai repede cu atât va fi mai încordată – și probabil și gâtul său. Persoana de test va putea acum să influențeze această încordare.

Lecțiile 11&12



Biofeedback GSR
https://www.youtube.com/watch?v=ZultgAFrxuc

Această lecție se bazează pe reacțiile emoționale la vizualizarea unui "film de groază": suișurile și coborâșurile unui roller coaster poate avea un efect imens asupra emoțiilor persoanei de test. Cum pot fi acestea influențate?

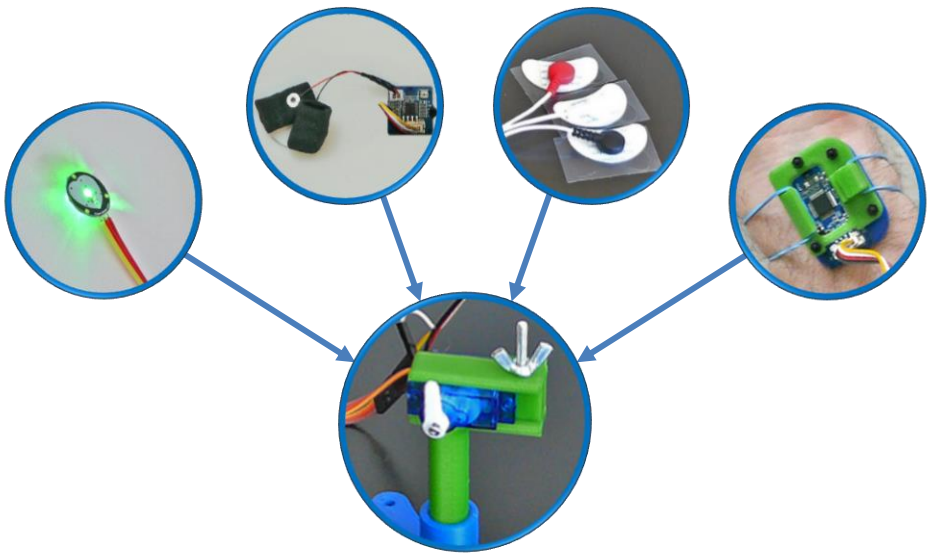
Ce efect ar avea vederea unor imagini cu păienjeni sau șerpi?

Care e efectul unor stimuli plăcuți cum e muzica? Care e efectul muzicii disco/muzicii clasice? Se obține un efect special când se ascultă melodia preferată?

Lecțiile 13 până la final (270 min):

Programarea liberă! Și biofeedback fericit! 😊

Încercați conectarea ieșirii a unor senzori diferiți la servo motor. Ce posibilități există pentru controlul unghiului de rotație a servomotorului?



10. Feedback	La sfârșitul lecțiilor, elevii ar trebui să dețină cunoștințe bine fundamentate cu privire la modul în care principiile IoT funcționează în cazul dispozitivelor medicale și cum biofeedback-ul poate ajuta la înțelegerea caracteristicilor ascunse ale corpului lor. Pe parcursul lecțiilor, profesorul asigură suport și îndrumare cu privire la aspecte importante de electronică și informatică medicală. În plus, se predau aspecte biologice ale activității musculare.
11. Evaluare	Elevii păstrează un jurnal al activităților derulate care poate fi verificat de profesor. În plus, la sfârșit elevi vor fi evaluați cu un test în clasă standard.

Cod sursă client Wemos

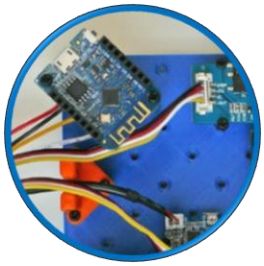


Figura 13: Codul sursă pentru clientul Wemos atașat la satia Arduino

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

Codul sursă server Wemos

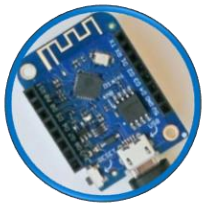


Figura 14: Acest script trebuie compilat pentru Wemos-ul conectat la PC

```
#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;
    }
  }
}
```

Exemplu de cod sursă EMG

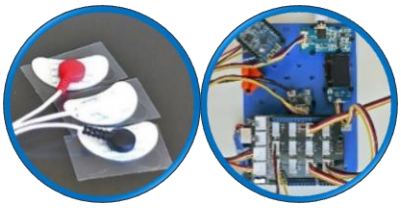


Figura 15: Scriptul pentru lecțiile 7&8 pentru stația Arduino

```
#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);
}
```

Senzorul de puls pe clientul Wemos folosind AnalogRead

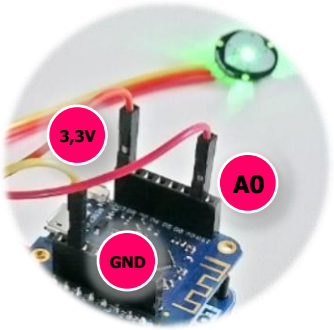


Figura 16: Conexiunile Wemos – Senzor de puls

```
#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);
  }
}
```

Codul sursă GSR pe stația 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);
}
```

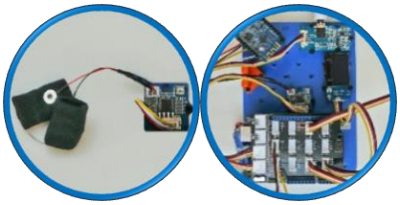
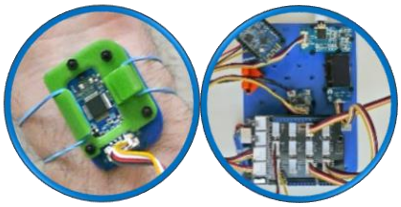


Figura 17: Codul sursă pentru stația Arduino – lecțiile 11&12

Lecția 5&6 codul sursă cu Processing App

Codul sursă Arduino pentru senzorul pentru deget Grove	Codul sursă Processing pentru senzorul pentru deget Grove
	

```
#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.println(c, DEC);
    Serial_89.println(c, DEC);
    SeeedOled.clearDisplay();
    SeeedOled.setTextXY(1, 2);
    SeeedOled.putString(c);
  }
  delay(500);
}
```

```
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();
}
```