

'We are the makers - IoT' Learning Scenario:

Cuburi EDA: formează-ți o idee cu privire la ce simte colegul tău!

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

Această lucrare a fost dezvoltată pornind de la activitatea din IO2, cu privire la stația de biofeedback (IO2). Stația de biofeedback a fost dezvoltată având în minte educația generală. Această activitate poate fi folosită cu scop non-tehnic, spre exemplu ca instrument de monitorizare a dialogurilor: pentru a dezvălui emoțiile utilizatorului, în măsura în care tehnologia EDA poate face acest lucru.

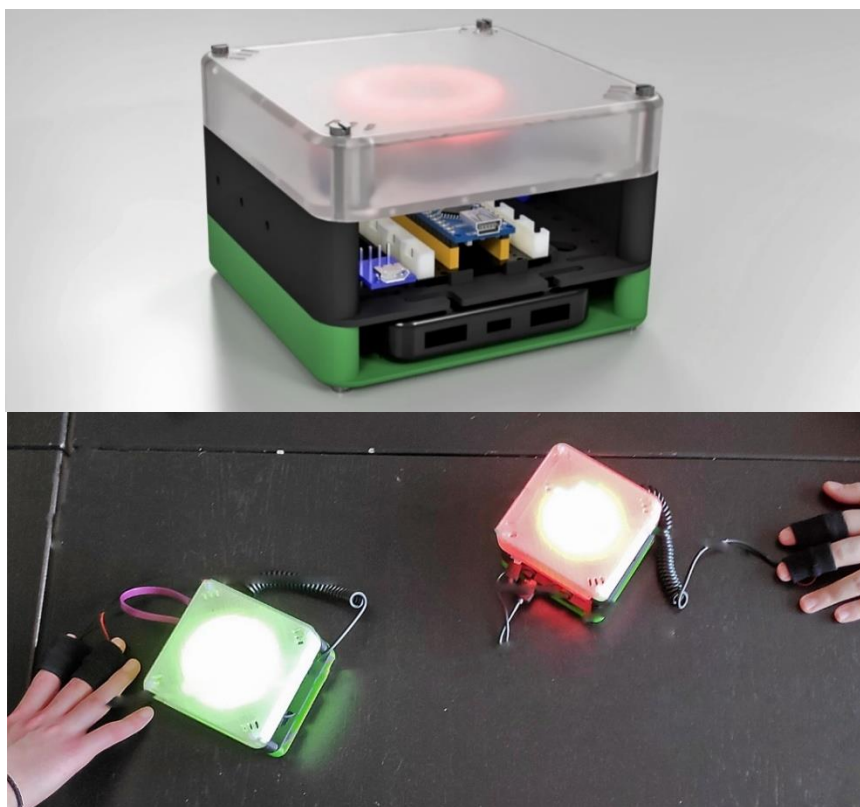


Figura 1: Cuburi EDA

Cum ar fi dacă doi oameni ar comunica și și-ar dori să fie politicoși unul cu celălalt? O mare problemă e că niciodată nu știi cum se simte partenerul de dialog. E supărat, speriat, nervos, fericit sau obosit? Care sunt efectele dialogului? În ochii partenerului ești nepolitic sau grijuliu? Poți avea un răspuns imediat la acest lucru, sau cel puțin un "indiciu"?

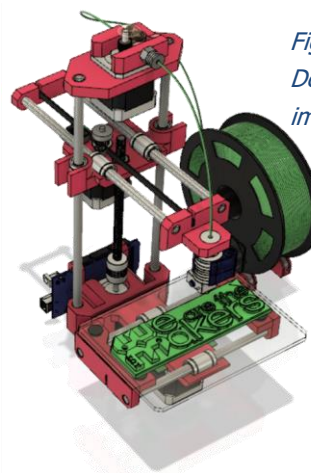
Și: Cum ar fi dacă ai purta un ceas inteligent care e capabil să-ți măsoare reacțiile emoționale? Ți-ai dori ca emoțiile tale să fie monitorizate de o mașină? Fii conștient de capacitățile senzorilor moderni! Hai să construim un dispozitiv inteligent care include un senzor de „emoții” care îți va interpreta reacțiile corpului tău!

| | |
|---------------------------------|--|
| 1. Titlu | Cuburi EDA: formează-ți o idee cu privire la ce simte colegul tău! |
| 2. Grup țintă | 12- 17 ani |
| 3. Durată | Minim 3 săptămâni cu 2*45min lecții/săptămână: în total circa 6-8 ore. |
| 4. Nevoile de învățare | <ul style="list-style-type: none"> ▪ Interacțiunea dintre componentele electronice și părțile corpului ▪ Monitorizarea și afectarea parametrilor biologici umani ▪ Lanțul de comunicare al dispozitivelor IoT ▪ Principiile senzorilor și actuatorilor ▪ Ce este EDA (en. Electrodermal activity, ro. activitate electrodermală) ▪ Principiile rețelelor de comunicații fără fir ▪ Construcția și imprimarea 3D a unor componente care să ajute în realizarea de măsurători. |
| 5. Rezultatele învățării | <ul style="list-style-type: none"> ▪ 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? ▪ Înțelegerea biosenzorilor și dobândirea abilității de a judeca aceste instrumente |
| 6. Metodologie | În cadrul acestei activități, elevii vor construi și programa, de la zero, un dispozitiv EDA interactiv care interpretează valorile măsurate și le reprezintă vizual. Elevii pot folosi tehnologia WiFi pentru a transmite valorile la un calculator pentru o procesare ulterioară. |
| 7. Locație / Mediu | <ul style="list-style-type: none"> ▪ 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; |

8. Unelte/ Materiale/ Resurse

Imprimante 3D

Sunt necesare circa 2-3 imprimante 3D, întrucât elevii își vor tipări singuri sistemul IoT creat. Desigur, elevii ar putea să-și proiecteze singuri componentele.



*Figura 2:
Desenul unei
imprimante 3D*

Componente tipărite 3d:

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

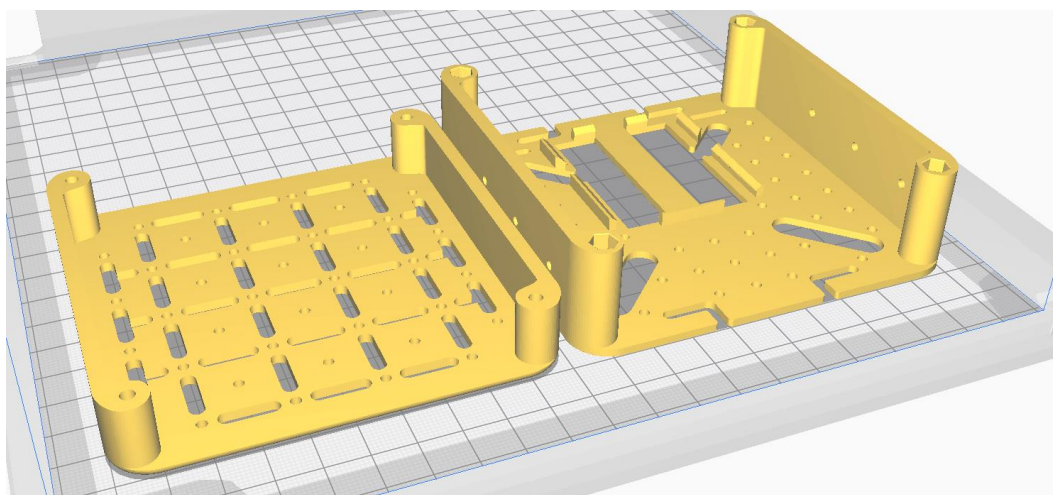


Figura 3: Componentele din PLA colorat

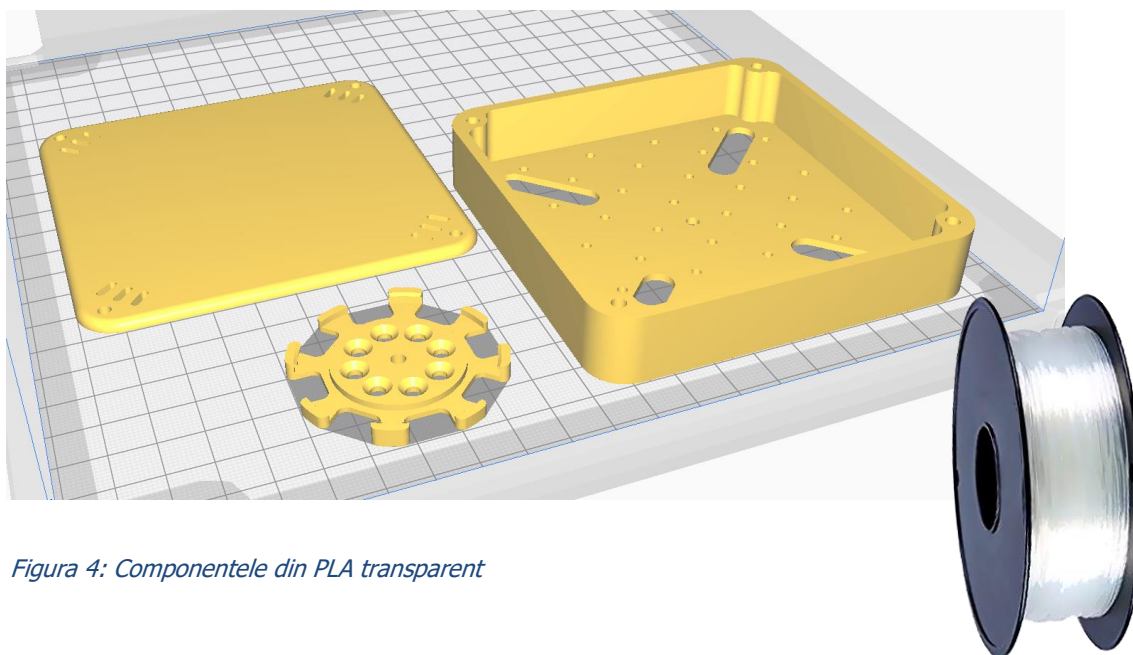


Figura 4: Componentele din PLA transparent

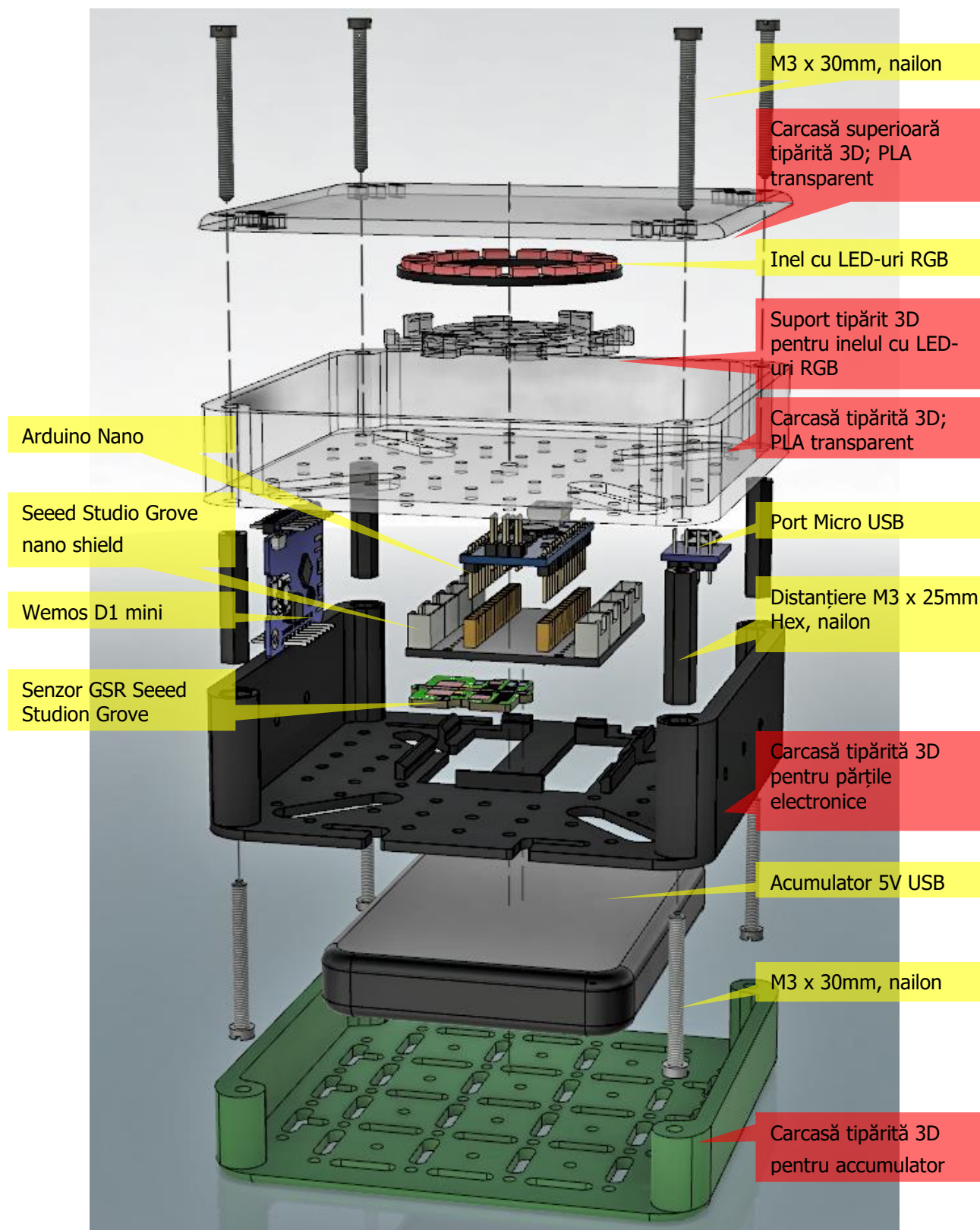


Figura 5: Desen cu modul de asamblare a cubului

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.

9. Configurarea componentelor

În această lucrare, se recomandă sistemul Seeed Grove ca bază deoarece e ușor de folosit (http://wiki.seeedstudio.com/Grove_System/).

Componente Seeed Studio:

1x Grove Shield pentru 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 cabluri Seeed Studio:

1x Grove – cablu de conversie cu patru pini tată la Grove 4 pini

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

2x Grove – Cabluri universale cu 4 pini de 5cm

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

1x Grove - Cabluri universale cu 4 pini de 20cm

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

Microcontroler:

1x Arduino Nano (sau echivalent)

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

1x Wemos LOLIN D1 mini (sau echivalent)

https://wiki.wemos.cc/products:d1:d1_mini

Componente electronice:

1x Inel Adafruit RGB-LED

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

1x Adaptor Micro USB la DIP

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

Componente diverse:

- Distanțiere de nailon 4x M3 25mm (spațiere Hex)
- Șuruburi de nailon 8x M3 30mm
- Distanțiere de nailon M2 (spațiere Hex) pentru Grove (are găuri de 2mm)
- Acumulator extern mic USB, max. 15mm x 100mm x 60mm
- Cablu micro USB mic pentru conectarea acumulatorului cu Nano
- Un fier de lipit pentru a lipi cablurile la componentele electronice



*Figura 6: M2 Distanțiere
din nailon*

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

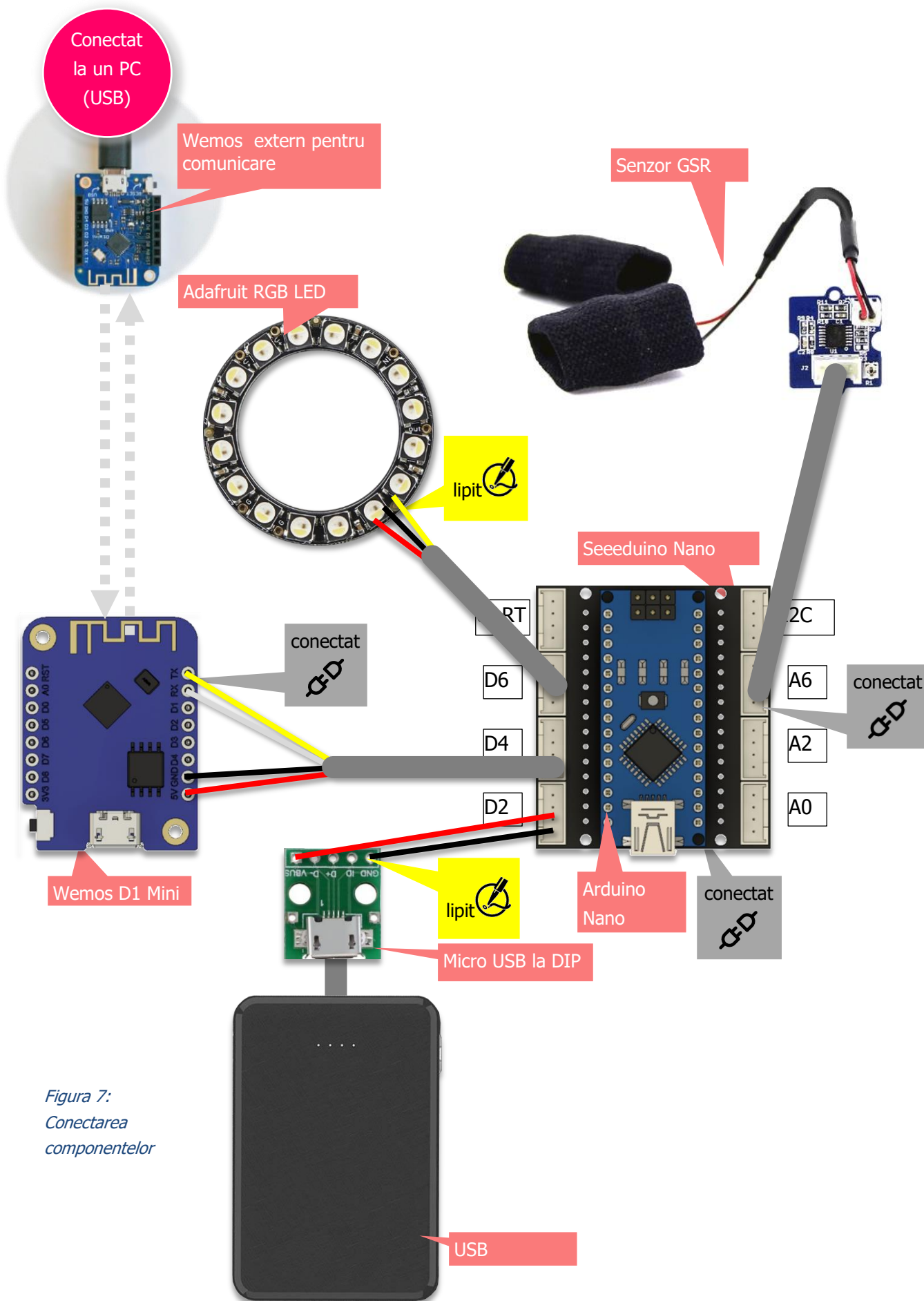


Figura 7:
Conectarea
componentelor

Biblioteci Arduino pentru componente:

Pentru a funcționa corespunzător, Wemos D1 Mini necesită biblioteci pentru Arduino IDE. Modul în care se pot importa biblioteci e prezentat aici: <https://www.arduino.cc/en/Guide/Libraries>

Neopixel (Adafruit):

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

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 următorul URL în câmpul "Additional Board Manager":
http://arduino.esp8266.com/stable/package_esp8266com_index.json
3. Deschideți Boards Manager din meniul Tools > Board și identificați 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.

Senzorul Grove GSR nu necesită nicio bibliotecă, deoarece poate fi controlat 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:

- Scanează toate rețelele wifi disponibile,
- Determină dacă există un canal nefolosit sau o rețea slabă
- Stabilește un punct de acces Wifi folosind primul canal disponibil găsit, combinat cu DHCP
- Așteaptă UN (doar unul!) Client care să se conecteze.
- Dacă acest Client se deconectează, va aștepta până acesta se reconectează.
- 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:

```
#include <ESP8266WiFi.h>
```

```
const char *ssid = "Erasmus";  
const char *password = "12345678";  
IPAddress Ip(192, 168, 3, 1);  
IPAddress NMask(255, 255, 255, 0);
```

Figura 8: tăierea codului sursă a serverului

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

```
#include <ESP8266WiFi.h>
```

```
const char* ssid      = "Erasmus";  
const char* password = "12345678";  
IPAddress server(192, 168, 3, 1);
```

Figura 9: tăierea codului sursă a clientului

```
WiFiClient client;  
char inChar;
```

- 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.

8c Teorie cu privire la 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 se întâmplă de cele mai multe ori în mod automat și inconștient. Spre exemplu dacă o persoană spune o minciună sau îi este frică, pielea sa începe să transpire. Această transpirație poate fi măsurată ca o schimbare a conductivității electrice, deoarece transpirația conține electroliți. Dacă se folosește un calculator pentru a reprezenta vizual această schimbare, omul poate corela starea sa emoțională cu semnalul măsurat și poate încerca să-și influențeze reacțiile și să învețe să-și controleze emoțiile. Emoțiile inconștiente anterior devin astfel conștiente.

Exist numeroase exemple și experimente pe care elevii le pot testa asupra lor:

- Pot încerca să-și influențeze ritmul cardiac prin modificarea ritmului respirator, monitorizați de un senzor de puls
- Pot încerca să controleze reacțiile la frică cu schimbarea activității electrodermale, monitorizați de senzori GSR (reacția la frică poate fi testată, spre exemplu, cu imaginea unui păianjen sau cu un film video cu un roller coaster)
- Pot crea un poligraf (detector de minciuni) – printre altele – se bazează pe schimbările activității electrodermale ce poate fi măsurate cu senzori GSR
- Coactivitatea mușchilor: Tastarea la calculator în condiții de stres conduce la contractarea mușchilor trapez, de la nivelul gâtului. Această contracție poate fi măsurată cu EMG.

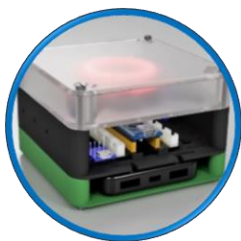
9. Descrierea pas cu pas a activității/conținutului

Lecțiile 1 & 2 (90min):



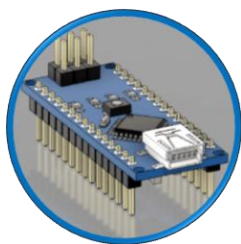
Elevilor li se va prezenta conceptul de IoT prin exemple: roboți de aspirare ce pot fi controlați de la distanță prin intermediul unei aplicații, stații meteo bazate pe web, agricultura inteligentă și la sfârșit aplicațiile de sănătate. Elevii trebuie să examineze modul în care funcționează aceste dispozitive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

Lecțiile 3 & 4 (90min):



Tipărirea 3D și asamblarea dispozitivului: Opțional, elevii pot imprima 3D carcasa dispozitivului, iar după aceea ar trebui să conecteze singuri toate piesele electronice. Elevii ar trebui să înțeleagă foarte bine modul în care piesele se potrivesc împreună și să construiască dispozitivul complet. Care este scopul inelului cu LED-uri RGB? Pentru ce este utilă conexiunea WiFi? Cum este procesat un semnal de la sursă până la utilizator? Se pleacă de la un semnal bio.

Acesta este transformat într-un semnal analogic în interiorul dispozitivului, apoi convertit într-un semnal digital de către convertorul analog-digital al microcontrolerului. Datele sunt prelucrate folosind software-ul și rezultatele sunt fie comunicate prin intermediul LED-urilor RGB fie sunt transmise folosind Wi-Fi.



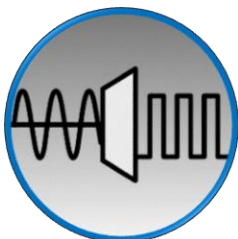
Lecțiile 5 & 6 (90min):

introducere în programarea Arduino: Configurarea conexiunii și comunicării dintre Arduino Nano și Arduino IDE / calculatorul. Trebuie explicată structura fundamentală a platformei Arduino: ce sunt pinii cu scop general de intrare/ieșire (GPIO), ce este logica digitală și care este diferența dintre intrările și ieșirile digitale.

Se vor scrie și modifica scripturi simple folosind exemple de bază care sunt livrate cu Arduino IDE:

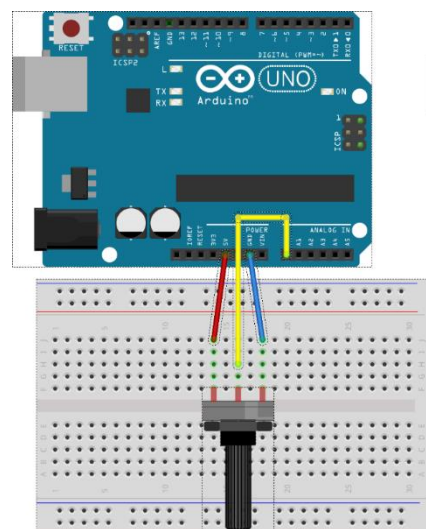
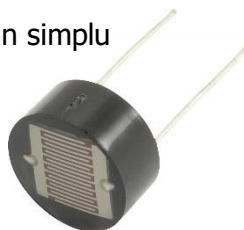
```
"01.Basics → Blink",
"01.Basics → DigitalReadSerial",
"04.Communications → SerialEvent",
```

Lecțiile 7 & 8 (90min):

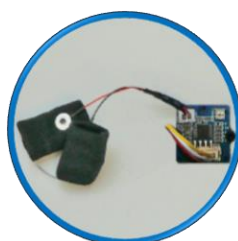


Cum este procesat un semnal analogic de către o mașină digitală? Se vor prezenta informații de bază despre conversia analog-digitală și se poate preda prin configurarea unui separator de tensiune simplu. Folosind Arduino, orice senzor simplu bazat pe un rezistor poate fi utilizat pentru a construi un

circuit simplu, ex. un LDR, un termistor sau chiar și un simplu potențiometru.



Lecțiile 9 & 10 (90min):

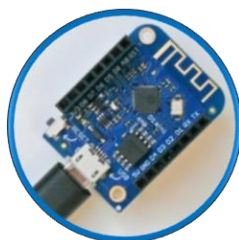


Prezentarea senzorului GSR: Se va preda amplificarea semnalelor folosind amplificatoare operaționale, prin măsurarea umidității pielii cu tensiuni mici. Cum se poate măsura rezistența pielii umane? Ce tensiune e inofensivă pentru utilizator?

Noțiuni privind activitatea electrodermală și semnificația/interpretarea acesteia:

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

Programarea senzorului GSR se poate realiza prin folosirea și adaptarea scripturilor incluse în Arduino IDE din lecțiile anterioare.



Lecțiile 10 & 11 (90min):

Introducere în D1 Mini: Acest microcontroler poate fi utilizat ca dispozitiv de transmisie wireless care completează caracterul IoT a cubului GSR. Sunt necesare două dispozitive Wemos pentru a conecta două noduri diferite – un nod este cubul EDA și celălalt este, spre exemplu, calculatorul unui elev către care sunt transmise semnalele procesate.

Deoarece tehnologia WiFi Wemos e foarte complicată și prin urmare necesită o unitate de învățare separată, toate dispozitivele Wemos trebuie preinstalate și configurate cu atenție înainte de lecție, de către profesor.

Este importantă sublinierea posibilității unui abuz: Există multe aspecte ale protecției vieții private care pot fi ilustrate cu ușurință cu această configurare. Care sunt consecințele transferării semnalelor provenite de la propriul corp în altă parte? Acest lucru duce la pierderea controlului asupra sferei private?

Lecția 13 & finalul (liber):



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

Această lecție se bazează pe reacțiile emoționale obținute în timpul vizualizării unui „film de groază”: un film cu un roller coaster poate avea un efect imens asupra emoțiilor persoanelor testate. Cum pot fi acestea influențate?

Ce spuneți de imagini cu păienjeni sau șerpi? Sau de ceva plăcut precum muzica? Care este efectul muzicii disco/clasice? Se obține un efect special la ascultarea melodiei preferate?

Și acum: programare liberă! Și biofeedback fericit! 😊

Încercați câteva experimente cu ceilalți colegi. Organizați o rundă de discuții și urmăriți ieșirea cubului vostru EDA. Îți poți da seama cum se simte colegul tău?

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 elevii vor fi evaluați, în clasă, cu un test standard.

Codul sursă a clientului Wemos

*Figura 10: Codul sursă
a clientului Wemos
atașat la cubul EDA*



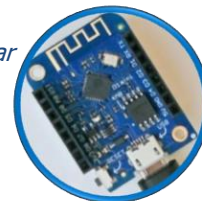
```
#include <ESP8266WiFi.h>
const char* ssid      = "Erasmus";
const char* password = "12345678";
IPAddress server(192, 168, 3, 1);
WiFiClient client;
char inChar;

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);
        }
    }
}
```

Codul sursă a serverului Wemos



*Figura 11: Acest script ar
trebui compilat pentru
dispozitivul Wemos
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;

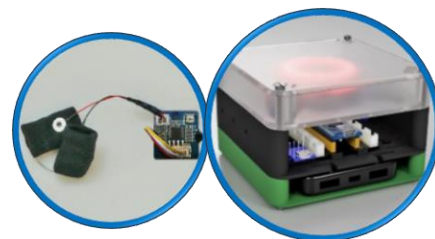
void setup() {
    Serial.begin(9600);
    unsigned int c_frei = SSID_scan();
    Serial.println("Configuring access point");
    WiFi.softAPConfig(Ip, Ip, NMask);
    WiFi.softAP(ssid, password, c_frei, false, 1);
    Serial.print("Channel: ");
    Serial.println(c_frei);
    Serial.println("Starting 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;
        }
    }
}
```

Codul plăcii Arduino Nano

*Figura 12: un cod sursă
funcțional pentru cubul
EDA-Cube pentru toate
lecțiile*



```
#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);
}
```