

'We are the makers - IoT' Learning Scenario: EDA-Cube: farsi un'idea di come si sente il proprio partner".

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

Il seguente documento è stato sviluppato come derivato della stazione di biofeedback (IO2). Poiché la stazione di biofeedback è stata costruita pensando all'istruzione generale, il cubo EDA può essere utilizzato per scopi non tecnici, ad esempio come strumento per il monitoraggio dei dialoghi: Esso rivela le emozioni dell'utente per quanto la tecnologia EDA lo consenta.

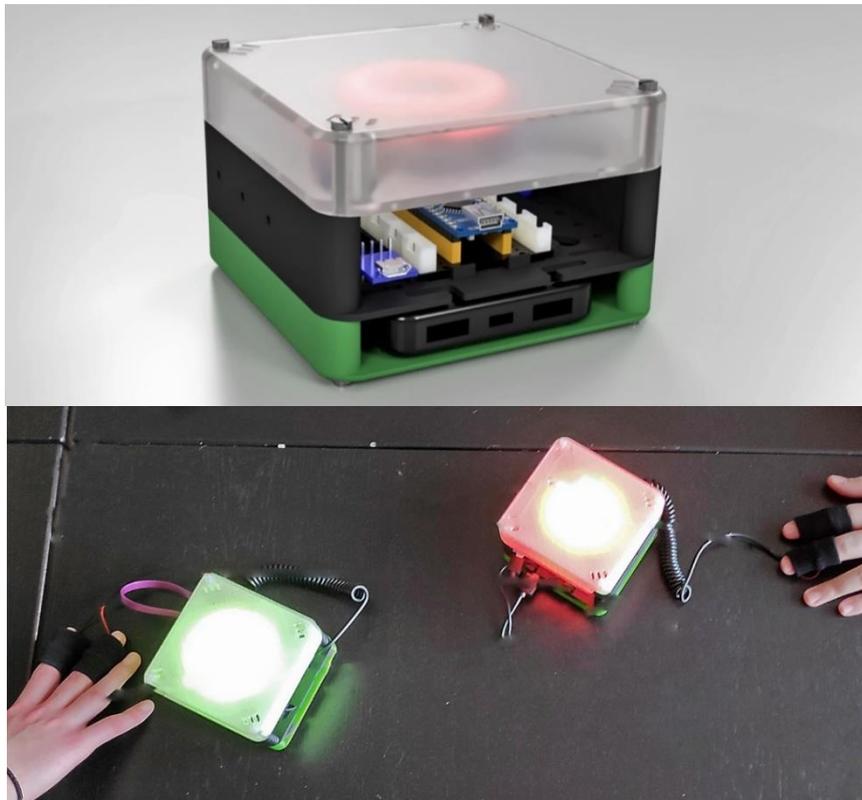


Figure 1: EDA-cubes in azione

E se due persone parlano insieme e vogliono essere educati l'uno con l'altro? Un grosso problema è che non si sa mai come si sente il proprio interlocutore. È arrabbiato, spaventato, nervoso, felice, stanco? Quali sono gli effetti delle sue parole? Sei - agli occhi del tuo partner - scortese o penseroso?

C'è una risposta immediata a questo, almeno un "suggerimento"?

E: E se indossate un orologio intelligente che è anche in grado di misurare le vostre reazioni emotive? Volete che i vostri sentimenti siano monitorati da una macchina? Siate consapevoli delle capacità della moderna sensorica! Costruiamo un dispositivo intelligente con un sensore "emozionale" che interpreti le reazioni del vostro corpo!

1. Titolo dello scenario	Imparare a coltivare le piante con l'aiuto di un robot IoT-plant
2. Gruppo target	12- 17 anni
3. Durata	At minimum 3 weeks of 2*45min-lessons per week: in sum about 6-8 hours.
4. Esigenze di apprendimento coperte attraverso l'esercizio	<ul style="list-style-type: none"> ▪ Interazione tra parti elettroniche e corpi umani ▪ Monitoraggio e influenza dei parametri biologici umani ▪ Catena di comunicazione dei dispositivi IoT ▪ Principi di sensori e attori ▪ Cos'è l'EDA? ▪ Principi delle reti di comunicazione wireless ▪ Costruzione e stampa 3D degli helper per la misurazione.
5. Risultati attesi dell'apprendimento	<ul style="list-style-type: none"> ▪ Come funziona un sistema intelligente di IoT? ▪ Dove sono le possibilità e le minacce dei sistemi EDA? ▪ Quali componenti - hardware e software - sono fondamentali per costruire un dispositivo IoT? ▪ Acquisire consapevolezza della biosensorica e della capacità di giudicare questi strumenti.
6. Metodologie	In questo scenario gli studenti costruiranno, costruiranno e programmeranno da soli un dispositivo EDA interattivo che interpreta i valori misurati e li visualizza. Gli studenti possono utilizzare la tecnologia Wi-Fi per passare i valori ad un computer per l'ulteriore elaborazione.
7. Luogo/ Ambiente	<ul style="list-style-type: none"> ▪ un laboratorio con un set di parti e componenti elettronici. ▪ ogni gruppo di studenti deve avere un computer o un portatile con privilegi amministrativi per l'installazione di diversi pacchetti software ▪ Un proiettore per l'insegnamento delle lezioni e la presentazione dei lavori degli studenti;

8. Utensili / Materiali / Risorse

Stampanti 3D

Sono necessarie circa 2-3 stampanti 3D, poiché gli studenti stamperanno IoT-biofeedback-stazioni. Naturalmente, è possibile per gli studenti costruire parti di macchine da soli

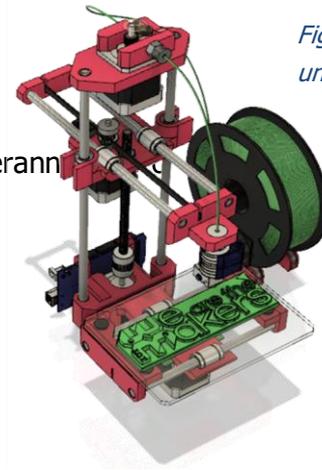


Figura 2: Disegno di una Stampante 3D

Componenti stampati 3d:

Come punto di partenza, tutte le parti necessarie sono fornite in formato stl e come Autodesk Fusion 360 Files.

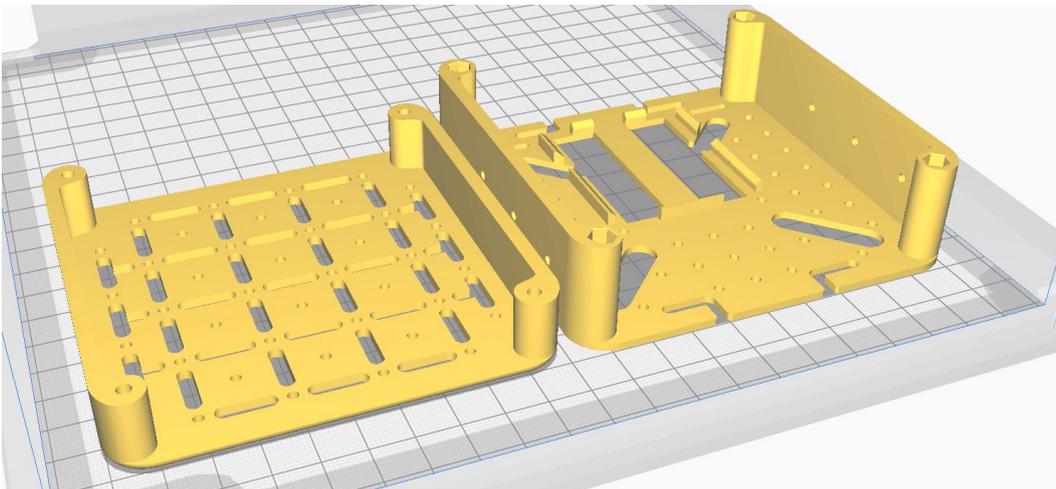


Figura 3: Dati STL di parti PLA regolari

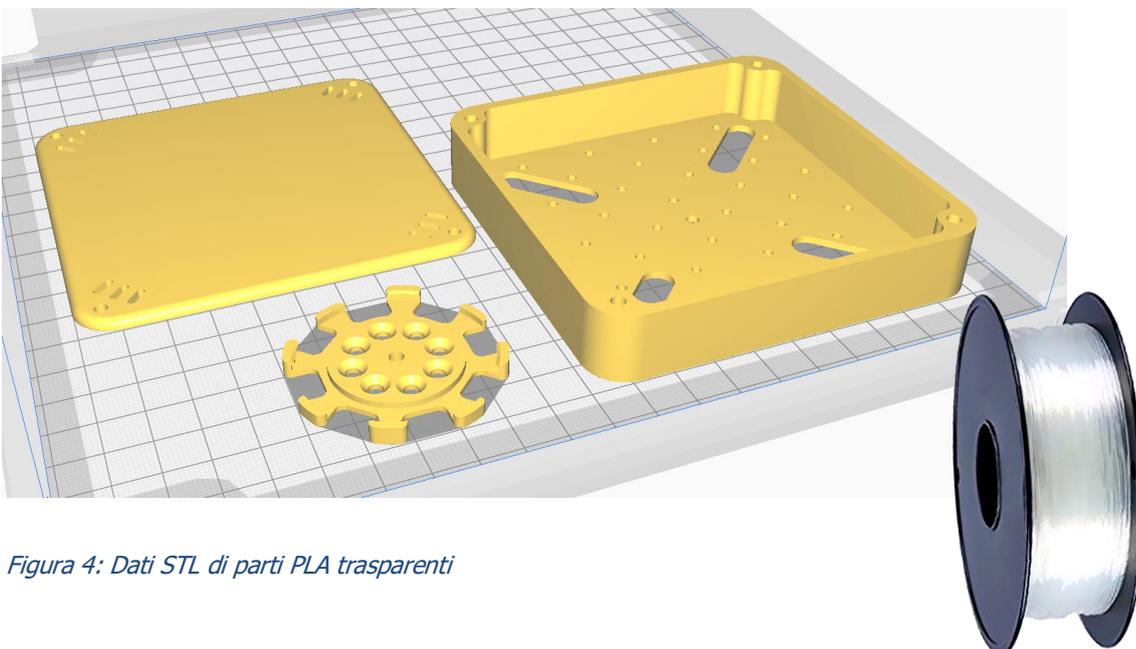


Figura 4: Dati STL di parti PLA trasparenti

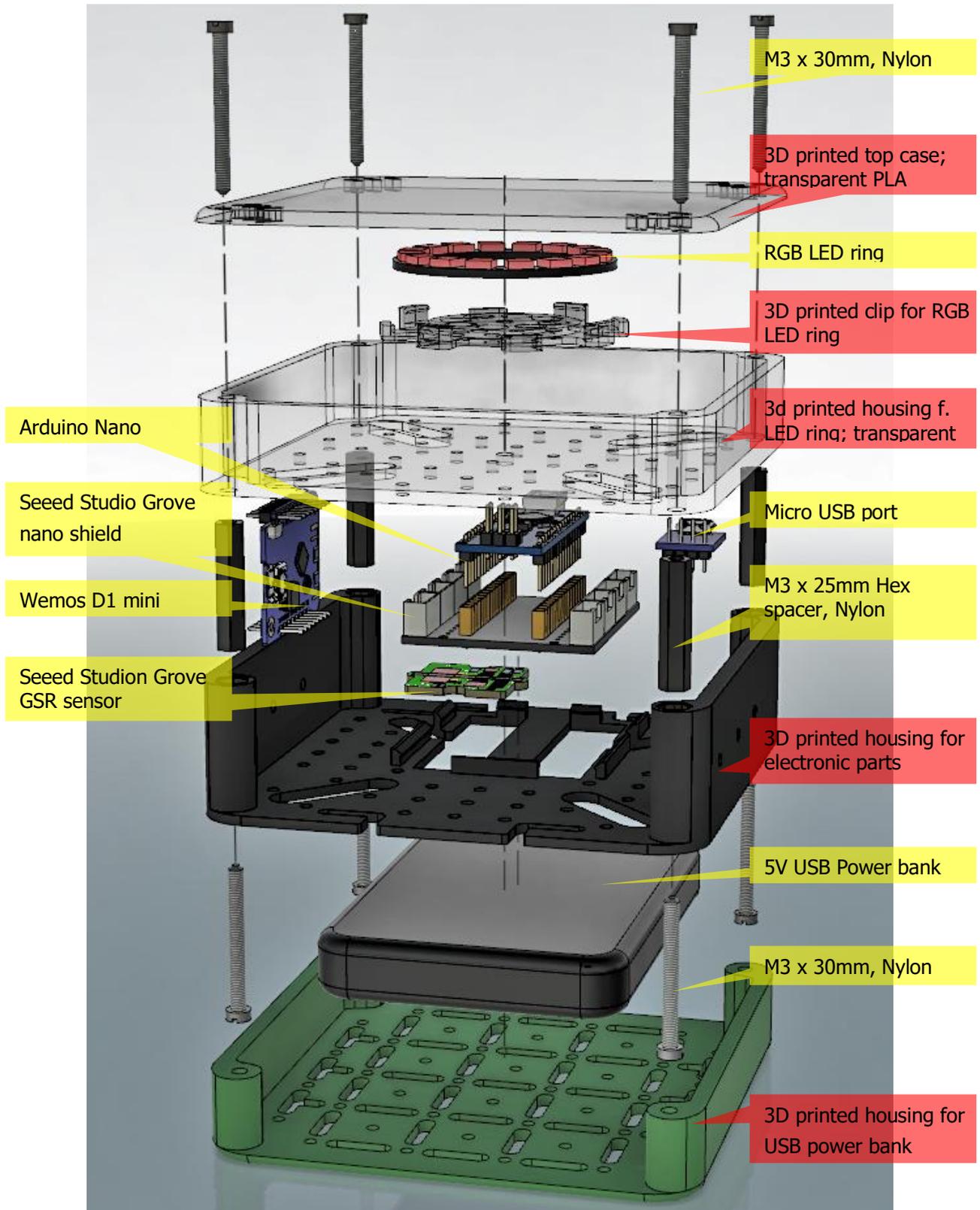


Figura 5: Disegno esploso del montaggio del cubo

Componenti elettroniche:

ATTENZIONE: Poiché stiamo facendo esperimenti con il corpo umano, è necessario prendere ogni precauzione! Non collegare mai un corpo umano al sistema elettrico domestico. Il corpo umano deve essere sempre tenuto completamente fuori dalla rete elettrica!

Questo include anche gli adattatori CA che sono inseriti nella presa a muro. Questo tipo di circuiti deve essere evitato. Utilizzare solo batterie e accumulatori a bassa tensione di circa 3-5V.

9. Setup componenti

In questo lavoro, raccomandiamo il sistema Seeed Grove come base, poiché la sua facilità d'uso: (http://wiki.seeedstudio.com/Grove_System/):

Seeed Studio Components:

1x Grove Shield for 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 Seeed Studio Cables:

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

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

2x Grove - Universal 4 Pin Buckled 5cm Cable

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

1x Grove - Universal 4 Pin 20cm Unbuckled Cable

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

Microcontrollers:

1x Arduino Nano (or equivalent)

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

1x Wemos LOLIN D1 mini (or equivalent)

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

Electronic parts:

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&oq=Micro+USB+to+DIP+Adapter&aqs=chrome..69i57.3497j0j7&sourceid=chrome&ie=UTF-8](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)

Parti varie:

- 4x M3 Nylon Standoffs 25mm (Hex spacer)
- 8x Viti M3 Nylon 30mm
- Distanziatori M2 in nylon (distanziatore esagonale) per Grove (ha fori da
- Piccolo banco di alimentazione USB, dimensioni massime 15mm x 100mm
- Piccolo cavo micro USB per il collegamento del Power Bank con Nano
- Un saldatore per fissare i cavi ai componenti elettronici

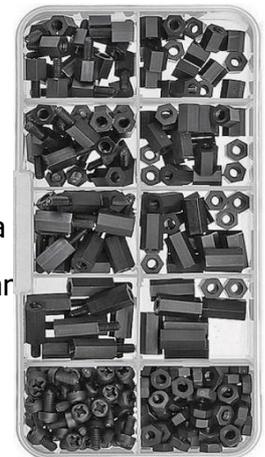


Figure 2: M2 Nylon Standoffs

Computer con i seguenti software installati

- Autodesk Fusion 360 (o qualsiasi altro software di modellazione 3D, ad esempio Wings3D)
- Software di slicing CURA,
- Una connessione internet per scaricare le biblioteche
- Arduino IDE
- Elaborazione IDE

Librerie Arduino per i componenti:

Il Wemos D1 Mini ha bisogno di librerie per il corretto funzionamento dell'IDE Arduino. Come importare una libreria è descritto qui: <https://www.arduino.cc/en/Guide/Libraries>

Neopixel (Adafruit):

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

URL di preferenza per le schede WEMOS (ESP8266):

Per installare i wemos è necessario installare la cosiddetta "definizione di bordo". Essa è descritta qui:

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

1. All'interno della finestra delle Preferenze di Arduino IDE aprire la finestra delle Preferenze.
2. Inserire il seguente URL nel campo "Additional Board Manager":
http://arduino.esp8266.com/stable/package_esp8266com_index.json
3. Aprire Boards Manager da Strumenti > Menu Board e trovare la piattaforma esp8266.
4. Selezionare la versione corrente da una casella a discesa e fare clic sul pulsante "installa".
5. Selezionare "(LOLIN) Wemos D1 R2 e Mini" dal menu Strumenti > Menu Board dopo l'installazione.

Il sensore Grove GSR non necessita di librerie in quanto può essere controllato con semplici comandi di ingresso analogico Arduino.

Wemos D1 mini come collegamento senza fili tra componenti elettronici

- **Le tavole di Wemos devono essere preparate dall'insegnante e non dagli studenti prima dell'inizio della lezione!**
- Le Wemos-ESP8266-Wifi-Board sono intese come alternativa meno costosa all'affidabile ma anche costosa tecnologia Xbee.

Due Wemos stanno costruendo una coppia che è collegata tramite la porta ethernet Wi-Fi 23 (che è Telnet). L'unico scopo è quello di sostituire il cavo di comunicazione seriale. Di solito un dispositivo elettronico sperimentale è collegato via cavo USB con il PC. Per ottenere un design completamente autonomo che non sia collegato all'impianto elettrico domestico, è necessario stabilire una connessione senza fili.

Pertanto, la consueta comunicazione seriale (UART) viene tradotta in Wifi e inviata da un Wemos, ricevuto dagli altri Wemos e ritradotto in comunicazione seriale. Per ragioni di compatibilità, la velocità di trasmissione è fissata a 9600 baud, poiché la comunicazione Software-Seriale di un Arduino Uno è limitata a 9600 baud.

Una mini coppia Wemos D1 è composta da un Server e da un client. Il server deve essere collegato al PC. Dovrebbe essere avviato all'inizio e sta facendo i seguenti passi:

1. Scansione di tutte le reti wifi disponibili,
2. Determinare, se c'è un canale libero non utilizzato o una rete debole in sottofondo,
3. Stabilire un punto di accesso Wifi utilizzando il primo canale libero, anche in combinazione con DHCP
4. In attesa di UN SOLO (uno solo!) Cliente che si connette.
5. Se il client si disconnette, il server aspetterà finché il client non si ricollega.
6. Se il Server viene resettato, iniziare da 1. (scansione delle reti)

Il cliente dovrebbe essere avviato come secondo e si conatterà e riconetterà automaticamente.

Come configurare il Server e il Client Wemos, spiegato su "Better Server source code":

Qui ci sono i relativi estratti del codice sorgente del server e del client che devono essere adattati per la configurazione delle singole coppie di schede 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 serverClient;
char inChar;
```

Figure 4: cutting of server sourcecode

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

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

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

```
WiFiClient client;
```

```
char inChar;
```

Figure 5: cutting of client sourcecode

- Entrambe le linee di codice sottolineate devono essere esattamente le stesse per una coppia di Wemos.
- Entrambe le linee di codice sottolineate devono essere adattate per ogni singola coppia di Wemos.

Cambia l' **IP-adress** con

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

Cambia l' **ssid** con

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

... compilare gli script all'interno dell'IDE Arduino e caricarli nelle apposite schede Wemos.

8c Qualche teoria di Biofeedback

Questo testo è inteso come una breve panoramica e può essere considerato come una raccolta di parole chiave importanti. Non è inteso come un libro di testo!

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

La risposta del corpo umano allo stress o alle influenze esterne avviene il più delle volte automaticamente e inconsciamente. Ad esempio, se un essere umano mente o ha paura, la sua pelle comincia a sudare. Questo sudore può essere misurato come una variazione della conduttività elettrica, poiché il sudore contiene elettroliti. Se il computer di misura visualizza questo cambiamento, l'umano può correlare il suo stato emotivo con il segnale misurato e può cercare di influenzare la sua reazione e imparare a controllare le sue emozioni. Le precedenti emozioni nascoste sono diventate consapevoli alla mente di questa persona.

Ci sono molti esempi ed esperimenti che gli studenti possono provare da soli:

- Influenzare la frequenza cardiaca con il cambiamento della frequenza respiratoria, monitorata da sensori di polso

- Influenzare le reazioni di paura con il cambiamento dell'attività elettrodermica, monitorate da sensori GSR
(Una reazione di paura potrebbe essere la foto di un ragno, un video di YouTube di un ottovolante)

- Un poligrafo (macchina della verità) è - tra le altre cose - basato sul cambiamento dell'attività elettrodermica e può essere misurato con sensori GSR

- Coattività dei muscoli: La digitazione al computer in condizioni di stress porta alla contrazione del muscolo trapezio nel collo dell'uomo. Questo può essere misurato con l'EMG.

9. Piano di lezioni: Descrizione passo dopo passo dell'attività/contenuto

Lezioni 1 & 2 (90min):



Gli studenti saranno introdotti con esempi di IoT: Robot sottovuoto con app remote, stazioni meteorologiche basate su internet, smart farming e infine applicazioni per la salute. Gli studenti dovrebbero esaminare come funzionano questi dispositivi e quali componenti sono necessari: un sistema basato su microcontrollore controlla e coordina i sensori e gli attori collegati. Inoltre, comunica e coordina con altri sistemi di tipo simile spesso tramite reti di comunicazione wireless. Parti necessarie: Sensori, attori, dispositivi di comunicazione. Occorre discutere delle possibilità e delle minacce e delle limitazioni: dove ha senso l'Internet degli oggetti e dove no?

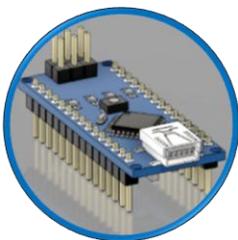
Lezioni 3 & 4 (90min):



Stampa 3D e montaggio del dispositivo: Opzionalmente gli studenti possono stampare in 3D l'involucro del dispositivo, e dopo di che dovrebbero collegare tutte le parti elettroniche da soli. Gli studenti dovrebbero sviluppare una comprensione più approfondita di come le parti si incastrano tra loro e quindi costruire un dispositivo completo. Qual è lo scopo di un anello a LED, a cosa serve la connessione Wi-Fi? Come viene elaborato un segnale dall'origine alla fine - all'utente? Partendo da un segnale biologico che viene convertito in un segnale analogico all'interno del dispositivo, convertendolo in un segnale digitale all'interno dell'ADC del microcontrollore, elaborando i dati tramite il software e comunicando i risultati attraverso la luce a LED o trasmettendo informazioni tramite Wi-Fi.

Lezioni 5 & 6 (90min):

Introduzione della Programmazione Arduino: Arduino IDE connessione e configurazione della



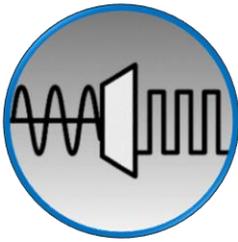
comunicazione di Arduino Nano con l'IDE / il computer. La struttura fondamentale della piattaforma Arduino deve essere spiegata: cosa sono i pin GPIO (general purpose input/output), cos'è la logica digitale e qual è la differenza tra ingresso e uscita digitale. I semplici script sono scritti e modificati utilizzando gli esempi di base che vengono forniti con l'IDE Arduino:

"01.Basics → Blink",

"01.Basics → DigitalReadSerial",

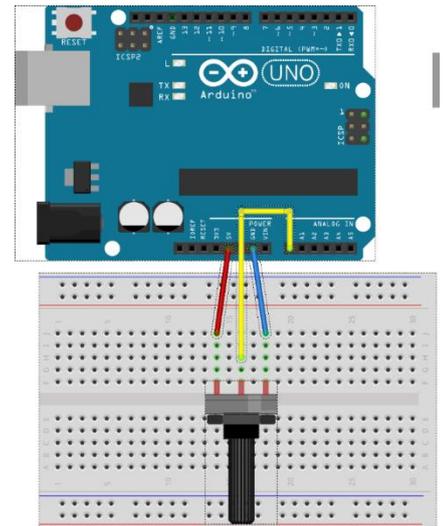
"04.Communications → SerialEvent",

Lezioni 7 & 8 (90min):

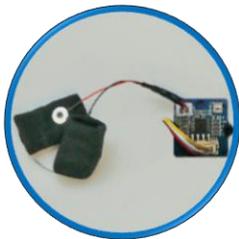


Come viene elaborato un segnale analogico da una macchina digitale? Viene introdotta la teoria della conversione del segnale analogico-digitale e può essere insegnata usando un semplice divisore di tensione. Utilizzando l'Arduino, qualsiasi semplice sensore basato su resistore può essere utilizzato

per costruire un semplice circuito, ad esempio un LDR, un termoresistore o anche un semplice potenziometro.



Lesson 9 & 10 (90min):



Vi presentiamo il sensore GSR: Insegnamento dell'amplificazione del segnale usando amplificatori operazionali, misurando l'umidità della pelle con basse tensioni. Come si può misurare la resistenza della pelle umana? Quale tensione è innocua per l'utente?

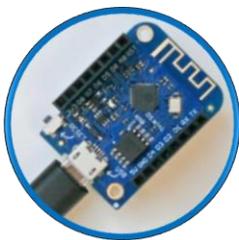
Background biologico dell'attività elettrodermica e suo

significato/interpretazione

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

La programmazione del sensore gsr può essere effettuata utilizzando e adattando gli script di Arduino pre-costruiti delle lezioni precedenti.

Lesson 10 & 11 (90min):



Vi presentiamo il Wemos D1 Mini: Questa scheda a microcontrollore può essere utilizzata come dispositivo di trasmissione senza fili che completa il carattere IoT del GSR-Cube. Due Wemos sono necessari per collegare due nodi diversi - un nodo è il Cube stesso e l'altro nodo è ad esempio il computer di uno studente dove tutti i segnali elaborati vengono trasferiti.

Poiché la tecnologia Wi-Fi di rete di Wemos è molto complicata e quindi necessita di un'unità didattica separata, tutti i dispositivi Wemos devono essere preinstallati e configurati con cura prima della lezione. Questo deve essere fatto dall'insegnante.

È importante sottolineare la possibilità di abusi: Ci sono molti aspetti della protezione della privacy che possono essere facilmente illustrati con questa impostazione. Quali sono le conseguenze di un trasferimento dei propri dati segnaletici da qualche parte che porta alla perdita di controllo della sfera privata?

Lezioni 13 & fine (inizio):



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

Questa lezione si basa su reazioni emotive con la visione di un "film di paura": Il saliscendi delle montagne russe può avere un enorme effetto sui sentimenti delle persone in prova. Come influenzarlo?

Che ne dite di alcune immagini di ragni o serpenti? O di qualcosa di delizioso / piacevole come la musica? Qual è l'effetto della musica da discoteca/ musica classica? C'è un effetto speciale mentre si ascolta la propria canzone preferita?

E ora: Programmazione freestyle! E buon biofeedback! 😊

Provate a fare qualche esperimento con i vostri compagni di classe. Fate un giro di discussione e tenete traccia dell'output del vostro EDA-cube. Riesci a riconoscere come si sente il tuo collega?

10. Feedback

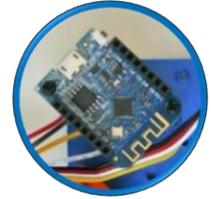
Alla fine della lezione, gli studenti dovrebbero avere una conoscenza approfondita di come funzionano i principi dell'internet degli oggetti nei dispositivi medici e di come il biofeedback possa aiutare a comprendere le caratteristiche nascoste del nostro corpo. Durante la lezione, sono stati insegnati importanti aspetti dell'elettronica, dell'informatica medica e delle basi della costruzione. Inoltre, sono stati insegnati gli aspetti biologici delle attività muscolari.

11. Valutazioni

Gli studenti tengono il loro diario del lavoro, che può essere rivisto dall'insegnante. Gli studenti possono anche presentare i risultati dei loro esperimenti. Inoltre, alla fine delle lezioni deve essere effettuato un test standard in classe.

Wemos Client Sourcecode

*Figura 10: Codice
sorgente per il client
Wemos allegato
all'EDA-Cube*



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

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

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

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

Wemos Server Sourcecode

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

Figura 11: Questo script dovrebbe essere compilato per i Wemos collegati al PC



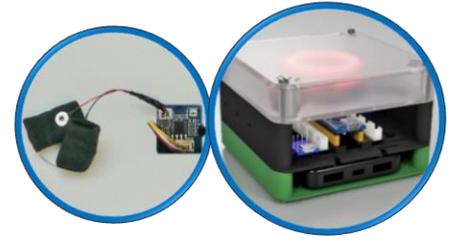
```

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;
    }
  }
}
}
}

```

EDA-Cube Arduino Nano code

Figura 12: un codice sorgente di esempio funzionante per l'EDA-Cube durante tutte le lezioni



```
#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);
}
```