

'We are the makers - IoT' Learning Scenario: biofeedback con sensori IoT-salute

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

Il seguente documento è stato sviluppato e testato in un ambiente scolastico con circa 18 studenti di 13-17 anni nell'anno scolastico 2018/2019. Esso riflette l'esperienza con molti meandri e alcuni fallimenti. Poiché il campo dell'internet degli oggetti è complesso, il materiale didattico deve essere scelto con cura. Questo documento dovrebbe essere una raccomandazione, come punto di partenza.

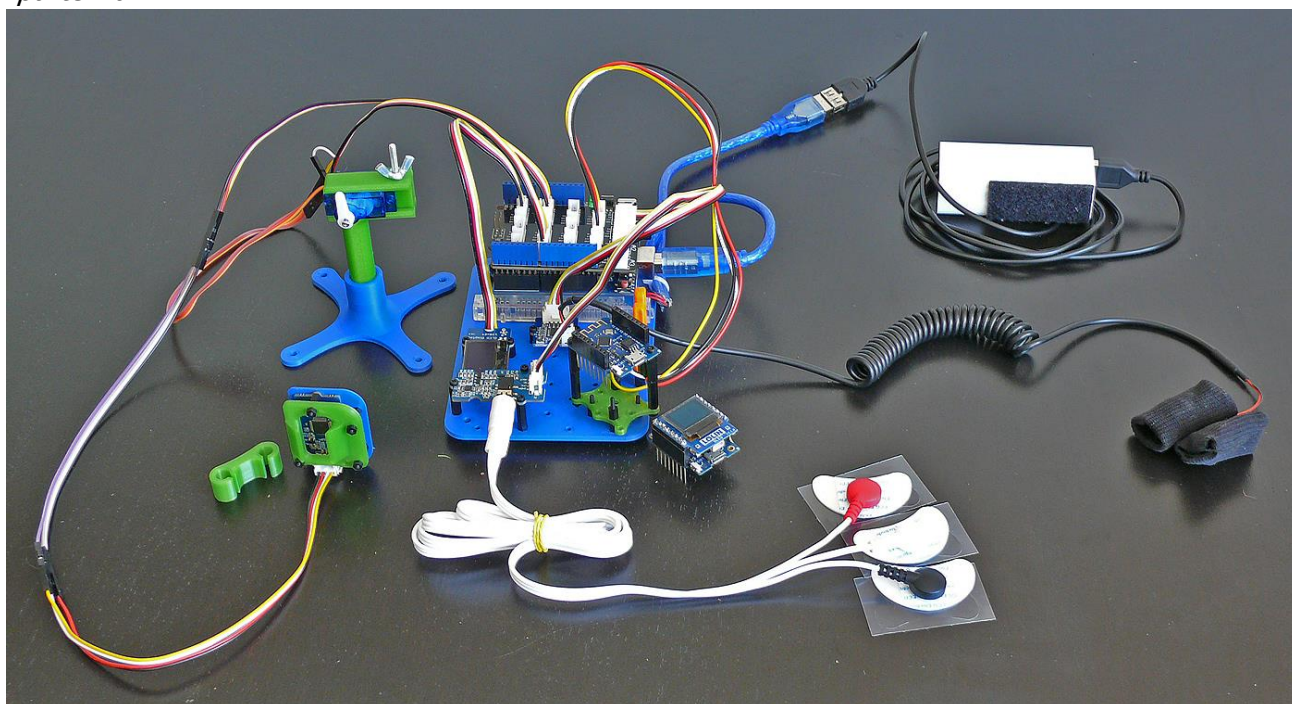
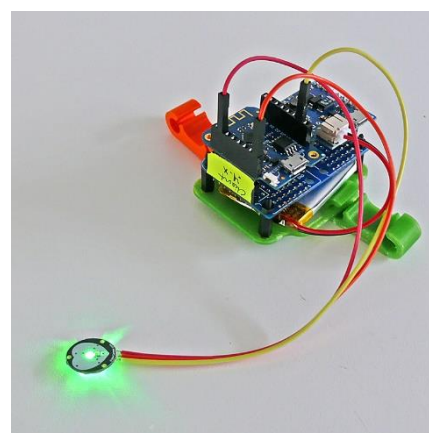


Figura 1: Prototipo di una stazione di biofeedback dell'internet degli oggetti

Che ne dite di una macchina della verità autoprogrammata o di un fitness tracker con un software di controllo autoprogrammato? O un pulsometro che mi aiuta a rilassarmi e tiene traccia del mio sforzo?

Che ne dite di un dispositivo che mi dia un feedback del mio tono muscolare e mi aiuti a controllarmi in situazioni di stress? Controllare l'uscita di una macchina con la potenza dei miei pensieri - possibile o no?

E: che ne dite di dispositivi per il controllo del sonno che aiutano a ottimizzare il mio sonno e informano un medico se non sto bene? Dove sono le possibilità e le minacce della moderna tecnologia dei biosensori dell'internet degli oggetti?



1. Titolo dello scenario	Imparare a far crescere le piante con l'aiuto di un IoT-plantrobot
2. Gruppo target	14 - 17 anni
3. Durata	Almeno 5 settimane di 2*45min-lessons a settimana: in totale circa 6-8 ore.
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 ▪ Diversi principi di misurazione dei bio-segnali. ▪ EMG I: Come funziona il sistema di conduzione muscolare? ▪ EMG II: Principi degli amplificatori di strumentazione ▪ 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 IoT? ▪ Dove sono le possibilità e i limiti di un sistema di internet degli oggetti basato sulla salute? ▪ Quali componenti - hardware e software - sono fondamentali per costruire un dispositivo IoT? ▪ Come dare biofeedback per aiutare le persone?
6. Metodologie	In questo scenario gli studenti costruiranno, costruiranno e programmeranno da soli un dispositivo di bio-segnale interattivo partendo da zero. Gli studenti useranno anche il Serial Monitor e il Serial plotter Arduino per visualizzare e tracciare il biofeedback.
7. Posto/ 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; ▪ ogni studente deve tenere un diario di laboratorio

8. Strumenti/ 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

Componenti stampati 3d:

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

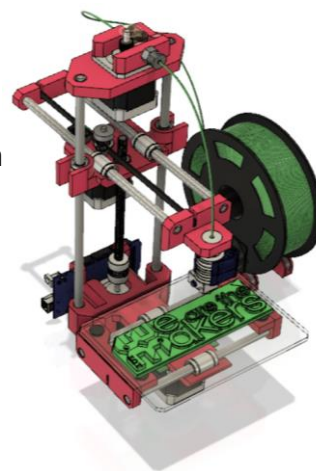


Figura 2: Simbolo della stampante 3D

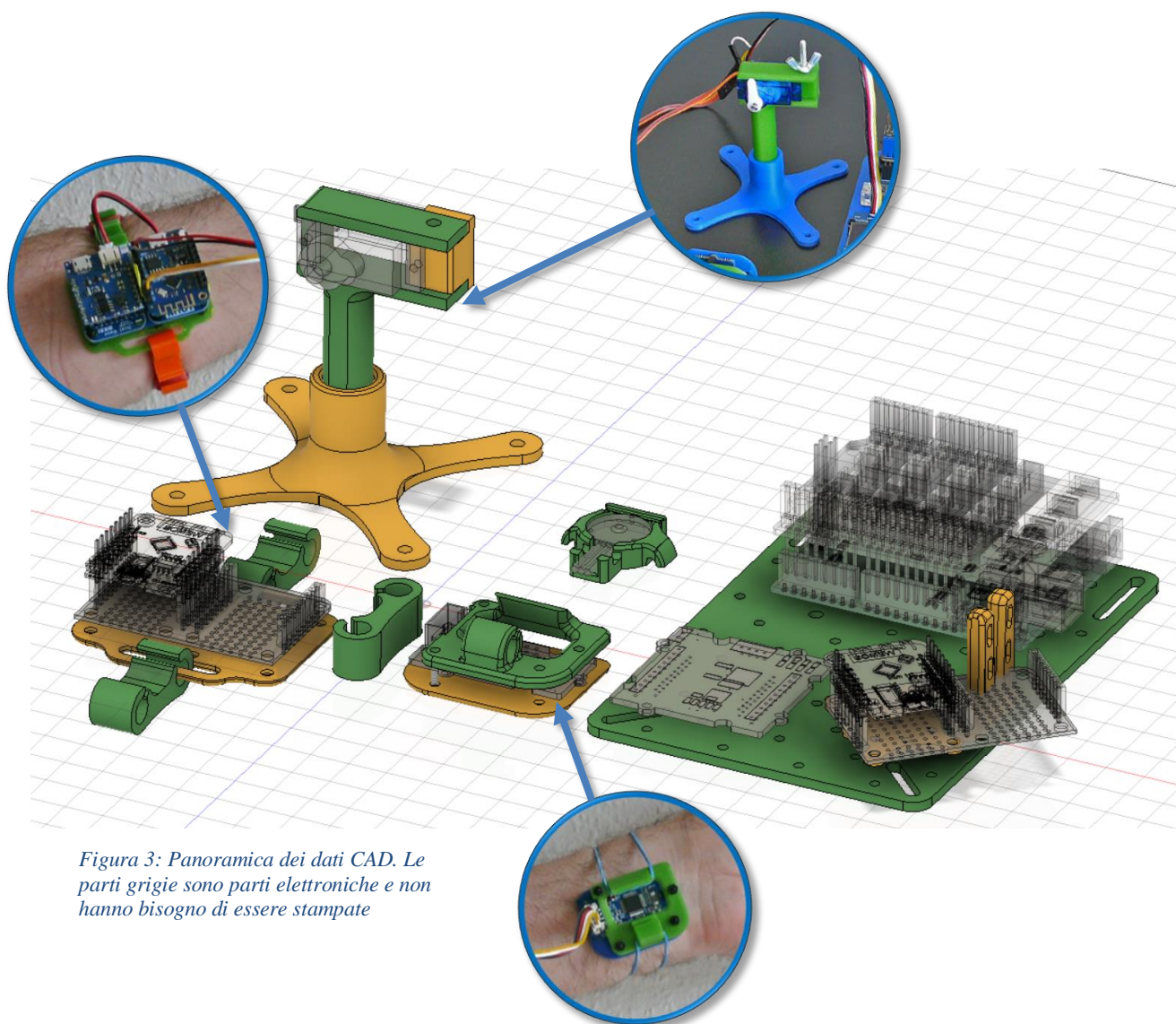


Figura 3: Panoramica dei dati CAD. Le parti grigie sono parti elettroniche e non hanno bisogno di essere stampate

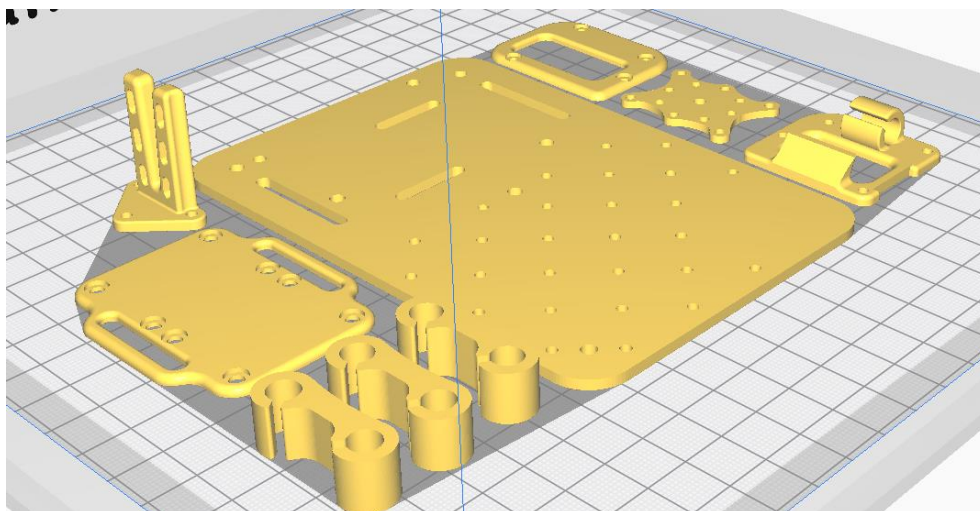
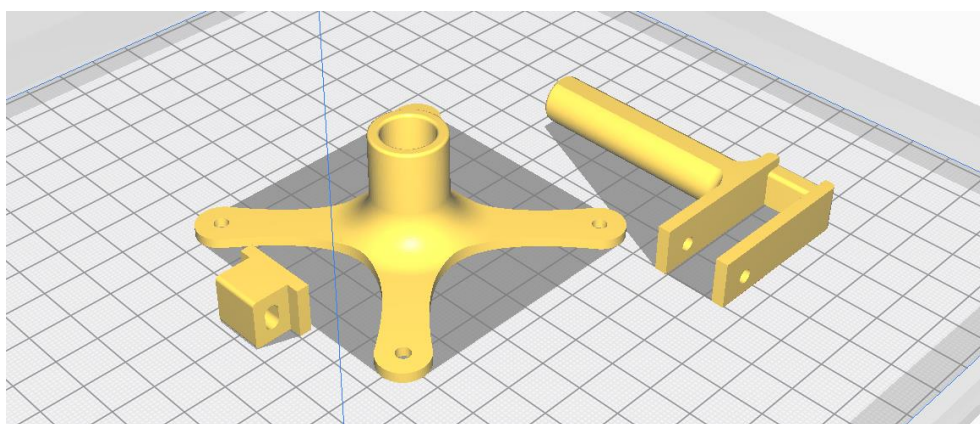


Figure 1: Set of .stl-files
ready for printing



Componenti elettroniche:

ATTENTIONE: *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.

In questo lavoro raccomandiamo il sistema Seeed Grove come base per la sua facilità d'uso:
(http://wiki.seeedstudio.com/Grove_System/) Tutti i componenti principali, ad eccezione di Wemos Chips, accumulatori e sensori cardiaci, appartengono allo standard di Grove:

Seeed Studio Componenti:

- 1: Grove Base Shield for 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. Grove Finger Clip heart rate http://wiki.seeedstudio.com/Grove-Finger-clip_Heart_Rate_Sensor/
5. Grove GSR http://wiki.seeedstudio.com/Grove-GSR_Sensor/

Sensori e attuatori regolari:

- 1: Arduino Uno (o equivalente)
- A: **2x** Wemos LOLIN D1 mini (o equivalente)
- B: Sensore da dito per battiti analogico (www.pulsesensor.com)
- C: Scheda batteria Wemos (https://wiki.wemos.cc/products:d1_mini_shields:battery_shield)
- D: Micro servo motore.

Parti varie:

- Nastro blister
- Banda di rame autoadesiva
- Distanziatori M3 in nylon (distanziatore esagonale)
- Distanziatori M2 in nylon (distanziatore esagonale) per Grove (ha fori da 2 mm)
- Fili per boschetti
- Dadi a leva WAGO
- Fili di ponticello
- Alimentatore USB per la ricarica dell'alimentatore USB
- Corrente massima dell'alimentatore USB
- Un saldatore per collegare i cavi ai componenti elettronici



Figura 2: Distanziatori in nylon

computer con il seguente software preinstallato:

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

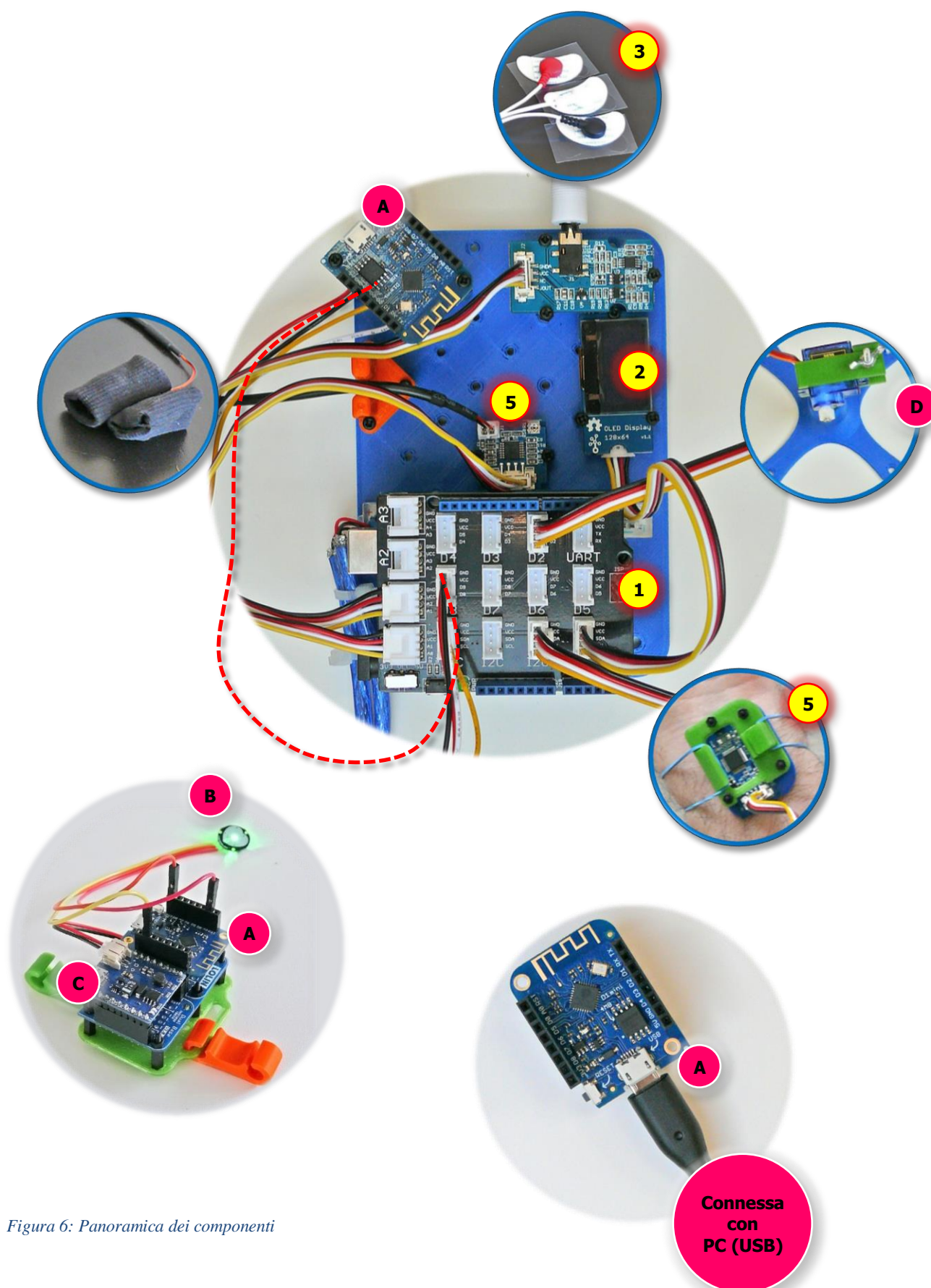


Figura 6: Panoramica dei componenti

Librerie Arduino per i componenti:

Alcuni componenti come il Wemos D1 Mini o alcune delle schede Grove hanno bisogno di librerie per il corretto funzionamento dell'IDE Arduino. Come importare una libreria è descritto qui:

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

OLED lib (Vedi):

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

Software I2C Master (Felias Fogg):

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

Preferenza URL for WEMOS-Boards (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.

Pulsossimetro:

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

Il Grove GSR, il Grove Finger Clip Sensor e il Grove EMG Detector non hanno bisogno di librerie, poiché possono essere controllati con semplici comandi Arduino.

Wemos D1 mini come collegamento senza fili tra componenti elettronici

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

Due Wemos stanno costruendo una coppia che è collegata tramite la porta ethernet Wifi 23 (che è Telnet). L'unico scopo è quello di sostituire il cavo di comunicazione seriale. Di solito un dispositivo elettronico sperimentale è collegato tramite 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 client dovrebbe essere avviato come secondo e si conatterà e riconetterà automaticamente.

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

Ecco i relativi estratti del codice sorgente del server e del client che devono essere adattati per la configurazione di 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 3: 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 4: 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 di Arduino e caricarli nelle apposite schede Wemos.

8b Qualche teoria sui potenziali di Axone e sulla misurazione EMG

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!

Il seguente articolo si basa su "EMG Fibel V1.1.pdf", "e Wikipedia-entra alle parole chiave "Action Potential", "Design of an EMG Detector", "EMG" al 'compendio elettronico":

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

I muscoli si contraggono perché ricevono segnali elettrici dai nervi: Il cosiddetto "Potenziale d'azione" è un cambiamento di potenziale elettrico delle terminazioni nervose intramuscolari. Il cambiamento avviene tra -80mV ("Potenziale a riposo") e +30mV ("Depolarizzazione"), il che rende in totale un cambiamento di potenziale teoricamente misurabile di 110mV. Poiché le terminazioni nervose sono all'interno delle fibre muscolari, (a scuola) possiamo misurare solo sulla pelle. Di conseguenza, gran parte dei potenziali d'azione della forza del segnale scompare a causa della resistenza elettrica della pelle e del tessuto connettivo. Un cambiamento tipicamente misurabile del potenziale elettrico è di circa 30mV.

I microcontrollori in grado di calcolare questi segnali utilizzano convertitori analogico-digitali. Questi ADC hanno di solito un campo di tensione di ingresso di 0-3,3V (tipo Wemos) o 0-5V (tipo Arduino Uno). La risoluzione di entrambi gli ADC è di 10bit, il che significa che i microcontrollori possono dividere l'intero campo di misura di 3,3V in 1024 passi discreti: $3.3V / 2^{10} = 3300mV / 1024 = 3,2mV$. Se misuriamo le variazioni di potenziale cutaneo con un microcontrollore solitario in piedi, possiamo ottenere solo un range tra 0 e 10 dei 1024 possibili valori discreti. Questo è molto meno. Inoltre, il potenziale cutaneo del corpo umano cambia con l'influenza dei campi elettrici esterni, il che porta a derive di potenziale 1000 volte più grandi della potenza del segnale.

Pertanto, abbiamo bisogno di una componente elettronica a monte che a) amplifichi il nostro segnale da 0,03V a 3,3V e b) sia in grado di compensare la deriva del campo elettrico. Il cosiddetto amplificatore per strumentazione è un circuito con tre ingressi: Un ingresso per '+', uno per '-' e uno per il riferimento. Mentre il bot '+' - e '-' -elettrodi' ha lo scopo di misurare la differenza di potenziale di 0,03V, l'elettrodo di riferimento si occuperà dell'influenza dei campi elettrici esterni per compensare la deriva.

All'interno di questo strumento di precisione elettronica, il segnale viene amplificato e questo è semplicemente tutto ciò che fa! Ora possiamo misurare e digitalizzare il potenziale di uscita con il microcontrollore.

Gli elettrodi devono essere posizionati con cautela, poiché dovrebbe misurare i potenziali nervosi, e più breve è la distanza dalle estremità nervose, migliore è il segnale: Un segnale nervoso viaggia attraverso il muscolo con una velocità di circa 5m/s. Se misuriamo nel mezzo del bicipite, entrambi gli elettrodi +/- - elettrodi dovrebbero essere posizionati a circa 10 cm di distanza l'uno dall'altro, mentre l'elettrodo di riferimento dovrebbe essere posizionato in un punto lontano da lì, ad esempio sulla mano. Come misura prevista dovrebbe esserci un tempo di transito del segnale da elettrodo ad elettrodo di ("onda di depolarizzazione"):

$$\frac{\text{distance of electrodes in cm}}{\frac{500 \text{ cm}}{\text{second}}} = 20 \text{ milliseconds}$$

In una misurazione EMG della pelle, vedremo una sovrapposizione di molti segnali provenienti da molte fibre muscolari diverse e quindi una forma d'onda di 20ms sarà difficilmente riconoscibile.

Per contrarre un muscolo con forza variabile il valore del potenziale d'azione non cambia. C'è solo un cambiamento di occorrenza dei cambiamenti di tensione nervosa: Più un muscolo si contrae con forza, più alto è il "tasso di fuoco" del potenziale d'azione dei nervi.

8c Alcune teorie su 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 misurazione 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 la variazione 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: l'immagine di un ragno, un video 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 umano. Questo può essere misurato con 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, non ultime, 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 anche delle limitazioni: dove ha senso e dove non ha senso l'Internet degli oggetti?

Lesson 3&4 (90 min)

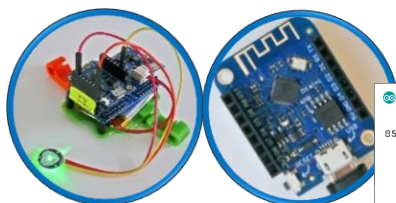


Figure 5: Necessary parts

Biosensori di base (sensore a impulsi): Agli studenti viene presentato il primo sensore che non ha bisogno di essere isolato

galvanicamente: Il sensore di impulsi. Si compone di due parti diverse: un semplice LED verde e un fototransistor più un

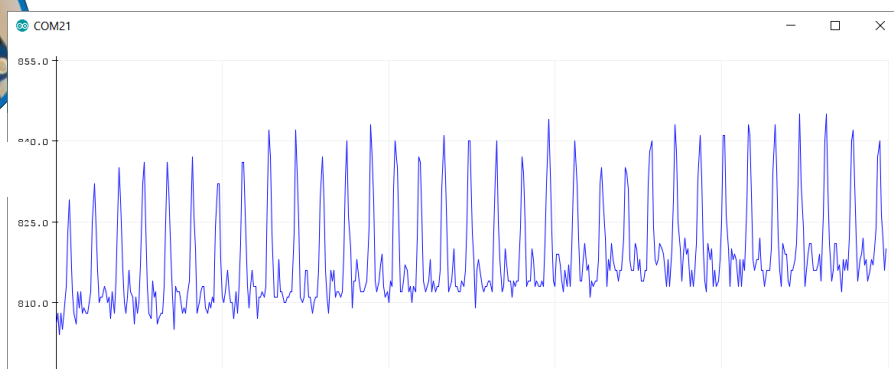


Figure 6: Sample data measured by pulse sensor

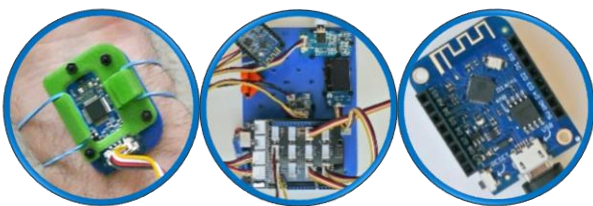
circuito amplificatore direttamente collegato. Deve essere posizionato direttamente sopra una vena, ad esempio la punta delle dita o l'auricolare.

Se la vena è gonfiata dal momento che il cuore sta pompando, il sangue rifletterà la luce verde e il fototransistor riconoscerà un valore elevato. Se la vena si contrae dal momento che il cuore si contrae, il sangue mancante porterà ad un maggiore assorbimento della luce all'interno del tessuto connettivo. Il fototransistor misurerà un valore più piccolo.

Gli studenti devono essere introdotti alla programmazione analogica di Arduino e al calcolo dei valori analogici: deve essere introdotto il convertitore analogico-digitale (ADC) e come funziona. Un'ottima introduzione può essere trovata qui:

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

Lezioni 5&6 (90 min)



Biofeedback della frequenza cardiaca:

Gli studenti imparano che molte reazioni del corpo corrispondono a processi emotivi inconsciamente. Ma se riusciamo ad accedere a queste reazioni corporee nascoste possiamo iniziare a controllarle: Il biofeedback è un monitoraggio in tempo reale delle proprie risposte fisiche con l'obiettivo di

raggiungere il controllo delle emozioni

(<https://www.artofmanliness.com/articles/hack-your-mind-like-a-twenty-first-century-soldier-using-biofeedback-to-become-more-resilient/>).

Se il software del sensore di polso funzionerà correttamente, gli studenti potranno iniziare a misurare le variazioni del polso e cercare di influenzare la frequenza cardiaca durante la respirazione: Qual è l'influenza della frequenza respiratoria sulla frequenza cardiaca? Cosa succede se respiriamo più velocemente o più lentamente? Cosa significa per le situazioni di stress? Possiamo diventare consapevoli della nostra frequenza cardiaca?

Utilizzare questa procedura di impostazione per un esperimento di biofeedback:



Figura 11: App. per l'elaborazione degli screenshot: Influenza della brigosità / contrasto con la frequenza cardiaca

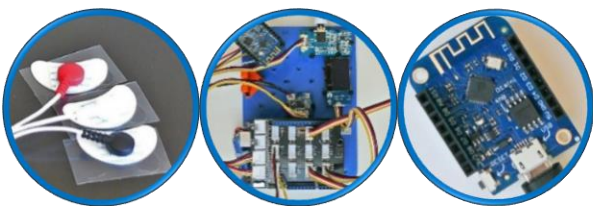
a) I2C Grove clip per il dito di frequenza cardiaca attaccata a un dito della mano dello studente
b) Collegare la stazione Arduino con il PC e avviare l'elaborazione del codice sorgente. Più veloce è la frequenza cardiaca, più scura è l'immagine: Influenzare la luminosità/contrasto con il controllo mentale dell'attività cardiaca.

c) Fare quanto segue:

- Inspirare lentamente un respiro profondo per 4 secondi.
- Trattenere il respiro per 4 secondi.
- Espirare lentamente il respiro per 4 secondi.
- Trattenere il respiro vuoto per 4 secondi.
- Ripetere fino a quando il respiro è sotto controllo.

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

Lezioni 7&8



Nozioni di base di EMG:

Insegnare le parole chiave e le relazioni del potenziale attivo dei nervi, della tensione muscolare e del rilassamento e dei principi di misura. Gli studenti devono capire come funziona il codice sorgente dell'amplificazione del segnale muscolare.

Come posizionare i tre elettrodi al braccio: All'inizio, la persona che esegue il test deve lavare il braccio con acqua e sapone e successivamente le posizioni in cui verranno posizionati gli elettrodi devono essere pulite con alcool e ovatta. Poi si possono posizionare tutti e tre gli elettrodi.

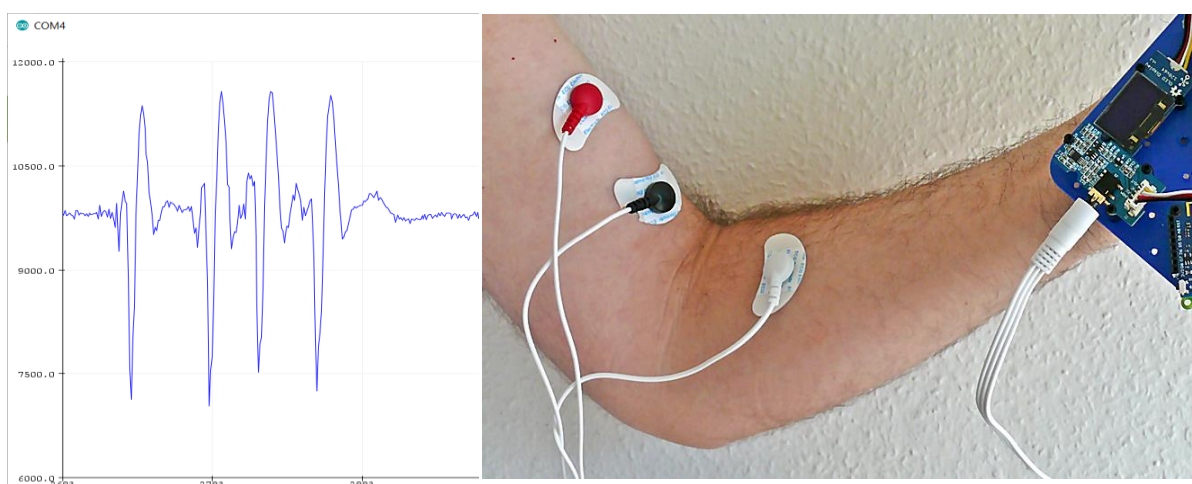
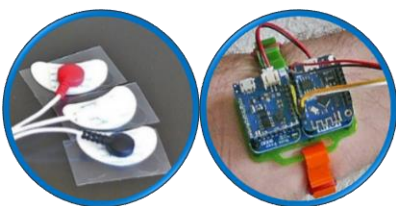


Figure 7: Biceps activity and corresponding electrode placement

Gli studenti dovrebbero sperimentare diverse velocità di movimento delle braccia, diversi pesi di sollevamento e l'effetto di rilassamento. Cosa succede se - proprio con questo posizionamento degli elettrodi - si chiude e si apre la mano? Cosa succede se si sta rimuovendo l'elettrodo bianco?

Lezioni 9&10



Esperimento EMG Biofeedback:

Il seguente esperimento si basa sulla tesi di dottorato di ricerca di Michael Schnoz:

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

Se una persona di prova digita le lettere sullo schermo del computer in modo veloce e altamente concentrato, probabilmente causerà un dolore al collo della persona di prova che può essere misurato e successivamente influenzato dalla persona di prova stessa.



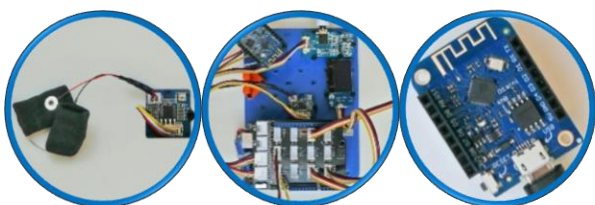
Figura 13: posizionamento degli elettrodi nel collo

Per mettere la persona di prova in condizioni di stress, può utilizzare il seguente software di tutorial di dattilografia:

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

Più veloce la persona lavora, più la persona - e probabilmente il suo collo - sarà tesa. La persona di prova può ora cercare di influenzare questa tensione.

Lezioni 11&12



Biofeedback GSR

<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?

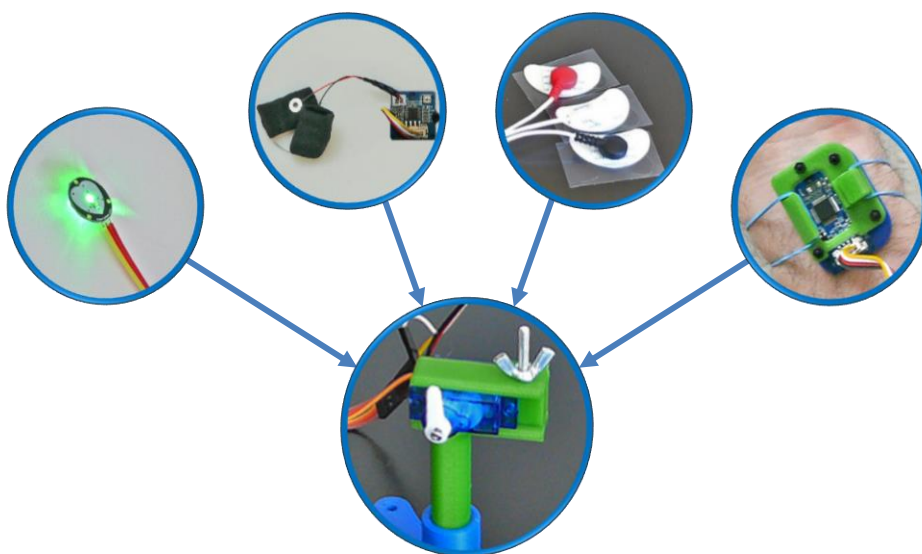
Che ne dite di alcune immagini di ragni o serpenti?

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?

Lezioni 13 per concludere finish (270 min):

Programmazione freestyle! E felice biofeedback! 😊

Provare a combinare l'uscita dei diversi sensori con il servomotore. Quali possibilità avete di controllare l'angolo di rotazione dei servi?



10. Feedback	<p>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.</p>
11. Valutrazione	<p>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.</p>

Wemos Client Sourcecode

```
#include <ESP8266WiFi.h>

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

WiFiClient client;
char inChar;

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

  while (WiFi.status() != WL_CONNECTED) {
    delay(5);
  }

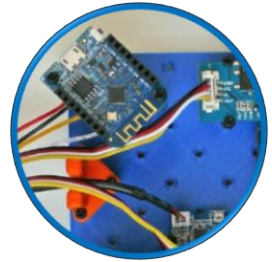
  Serial.print("WiFi Channel: ");
  Serial.println(WiFi.channel());

  if (client.connect(server, 23)) {
    Serial.print("Local IP: ");
    Serial.println(WiFi.localIP());
    pinMode(LED_BUILTIN, OUTPUT);
    digitalWrite(LED_BUILTIN, LOW);
  }
}

void loop() {
  if (!client.connected()) {
    digitalWrite(LED_BUILTIN, HIGH);
    unsigned long startzeit = micros();
    client.connect(server, 23);
    Serial.println(micros() - startzeit);
  } else {
    digitalWrite(LED_BUILTIN, LOW);
  }

  if (client.available()) {
    char c = client.read();
    Serial.print(c);
  }

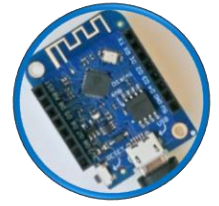
  while (Serial.available() > 0) {
    inChar = Serial.read();
    if (client.connected()) {
      client.write(inChar);
      delay(1);
    }
  }
}
```



*Figura 14: Codice sorgente
del client Wemos collegato
alla stazione Arduino*

Wemos Server Sourcecode

Figura 15: Questo script dovrebbe essere compilato per i Wemos collegati al PC



```
#include <ESP8266WiFi.h>

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

WiFiServer server(23);
WiFiClient sClient;
char inChar;

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

    Serial.println("Starting server");
    server.begin();
    server.setNoDelay(true);

    Serial.print("Server IP: ");
    Serial.println(WiFi.softAPIP());
    pinMode(LED_BUILTIN, OUTPUT);
    digitalWrite(LED_BUILTIN, HIGH);
}

void loop() {
    uint8_t i;
    if (server.hasClient()) {
        if (!sClient || !sClient.connected()) {
            if (sClient) sClient.stop();
            sClient = server.available();
            digitalWrite(LED_BUILTIN, LOW);
        }
        else digitalWrite(LED_BUILTIN, HIGH);
        if (sClient.available()) {
            digitalWrite(LED_BUILTIN, LOW);
            while (sClient.available()) {
                inChar = sClient.read();
                Serial.write(inChar);
            }
        }
        else digitalWrite(LED_BUILTIN, HIGH);

        if (Serial.available()) {
            size_t len = Serial.available();
            uint8_t sbuf[len];
            Serial.readBytes(sbuf, len);
            if (sClient.connected()) {
                sClient.write(sbuf, len);
                Serial.write(sbuf, len);
            }
        }
    }
}

int SSID_scan() {
    int frei = 0;
    Serial.println("scan start");
    WiFi.disconnect();
    delay(100);
    int n = WiFi.scanNetworks();
    if (n == 0) {
        Serial.println("no networks found");
        frei = 1;
    } else {
        int belegt[n];
        int staerke[n];
        Serial.print(n);
        Serial.println(" networks found.");
        for (int i = 0; i < n; ++i) {
            belegt[i] = WiFi.channel(i);
            staerke[i] = WiFi.RSSI(i);
            delay(10);
        }
        for (int i = 0; i < 12; ++i) {
            int diff = belegt[i + 1] - belegt[i];
            if (diff > 1) {
                frei = belegt[i] + 1;
                break;
            }
        }
        if (frei != 0) {
            Serial.print("done. free channel: ");
            Serial.println(frei);
            return frei;
        } else {
            int maxnummer = 0;
            int maxstaerke = staerke[maxnummer];
            for (int j = 0; j < n; j++) {
                if (maxstaerke > staerke[j]) {
                    maxnummer = j;
                    maxstaerke = staerke[maxnummer];
                }
            }
            frei = belegt[maxnummer];
            Serial.print("done. weakest channel: ");
            Serial.println(frei);
            return frei;
        }
    }
}
```

EMG Sourcedcode example

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

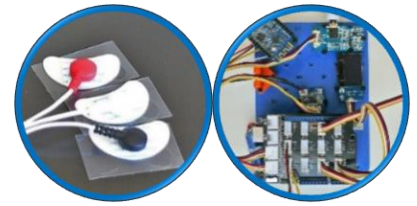
int max_analog_dta      = 300; // max analog data
int min_analog_dta      = 100; // min analog data
int static_analog_dta   = 0;   // static analog data
int level = 5;

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

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

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

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



*Figura 16: Script per le lezioni 7&8
alla stazione di Arduino*

Pulse-Sensor on Wemos Client using AnalogRead

```
#include <ESP8266WiFi.h>

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

WiFiClient client;
char inChar;

void setup() {
  Serial.begin(9600);
  pinMode(A0, INPUT);
  WiFi.setSleepMode(WIFI_NONE_SLEEP);
  WiFi.mode(WIFI_STA);
  WiFi.setOutputPower(10); // 10: 10mW, 14: 25mW, 17: 50mW, 20: 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);
  }
  delay(50);
  if (client.connected()) {
    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);
  }
}
```

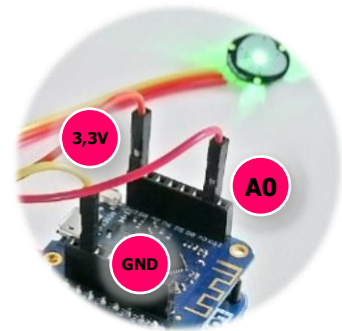


Figura 17: Cablaggio Wemos -
Sensore di impulsi

GSR Source code on Station client

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

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

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

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

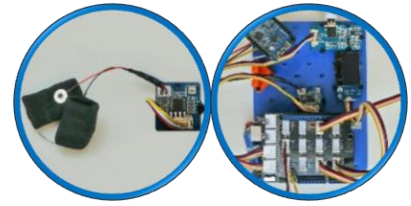


Figura 18: Codice sorgente per le lezioni della stazione di Arduino 11&12

Lezione 5&6 codice sorgente con l'applicazione di elaborazione

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