

## RELAZIONE FINALE

### Progetto “**Brainwave lighting a thinking to deaf people**”

(a cura del Prof. Ing. Fabio Scanzani)

#### DESCRIZIONE DELLE ATTIVITA’.

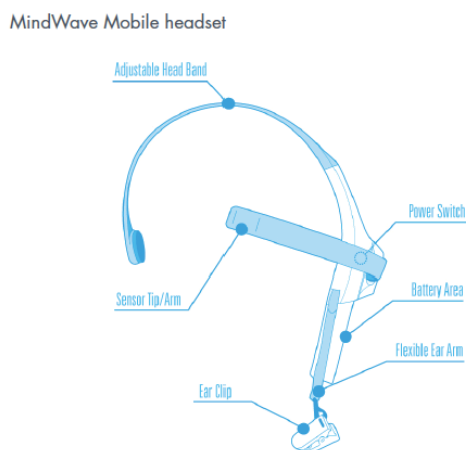
Il progetto si è svolto nel periodo compreso tra Marzo e Maggio 2019. Gli studenti che hanno partecipato con continuità all’iniziativa appartenevano tutti al triennio di IPSIA (delle classi IV e V MAT) e si sono alternati in attività di costruzione, osservazione, illustrazione a terzi (durante l’evento Rome Cup 2019) e documentazione multimediale delle attività di progetto ed analisi del tracciato encefalografico (EEG) rilevato attraverso un sofisticato sistema di misura basato sulla cuffia neurale commerciale denominata **Mindwave** della Neurosky interfacciata al PC e/o alla scheda Arduino.

I risultati ottenuti attraverso il dispositivo messo a punto dagli stessi studenti sotto la guida esperta del docente di Tecnologie Elettriche ed Elettroniche proponente, sono serviti a comprendere meglio la metodologia del biofeedback neurologico oggi oggetto di studio e di approfondimento a livello internazionale da numerosi gruppi di studio di scienziati operanti nell’ambito delle neuroscienze.

Nel corso della sperimentazione gli studenti oltre ad approfondire metodiche di acquisizione di segnali digitali attraverso il PC e la scheda ARDUINO, hanno potuto sperimentare, sia singolarmente che su altri numerosi studenti intervenuti nel corso delle 3 giornate trascorse presso Rome Cup 2019, ( vedi foto allegate) come un soggetto umano sottoposto alla misura del suo tracciato EEG concentrandosi su un’idea specifica e/o focalizzando la sua attenzione su un particolare stimolo visivo (per questo adatto anche ai soggetti affetti da sordità) possa incrementare i suoi livelli dell’attenzione (indicatore dell’attività cerebrale rilevato dalla cuffia attraverso la misura EEG) influenzando positivamente sull’accrescimento di tale livello di attenzione di cui avrà costantemente visione nel tempo (biofeedback positivo) . Nella maggioranza dei casi esaminati nel corso della sperimentazione (oltre 100 soggetti) questo esercizio si è dimostrato efficace nello **stimolare il cervello a migliorare** le performance del livello di attenzione con risultati particolarmente interessanti soprattutto per i soggetti affetti da sordità.

#### PRINCIPIO DI FUNZIONAMENTO DELLA CUFFIA CEREBRALE UTILIZZATA.

**La tecnologia innovativa** (denominata *ThinkGear*) impiegata dalla cuffia Mindwave di NeuroSky, consente di rilevare le onde cerebrali del tracciato elettrico EEG prelevato dal sensore sulla fronte di chi la indossa. Questo **sensore (elettrodo)** che tocca la fronte del soggetto, rileva la tensione ( differenza di potenziale elettrico) tra questo primario punto di contatto e quello di riferimento localizzato nella clip posta sull’orecchio ( reference) del soggetto. Il chip elettronico integrato nella cuffia, quindi, una volta campionato tale segnale EEG lo elabora trasformandolo in un segnale numerico (DATA) che viene poi inviato, tramite un interfaccia di tipo Bluetooth (che realizza anche il necessario isolamento galvanico per la prevenzione da rischi elettrici del soggetto) al PC o alternativamente alla scheda Arduino, principale obiettivo realizzativo del presente progetto ( v.di foto allegate). La codifica digitale del segnale EEG avviene direttamente all’interno della cuffia tramite **un algoritmo proprietario** brevettato da NeuroSky denominato **eSenseTM**.



Le misure di attività cerebrale **eSense** (da cui viene estrapolato il **livello di attenzione** ), **avviene totalmente in tempo reale** : l'agoritmo ThinkGear amplifica il segnale dell'attività cerebrale rilevato attraverso il sensore elettrico, rimuove il rumore ambientale e i movimenti dei muscoli ( considerati rumore elettrico spurio), e trasforma il livelli codificati in digitale e l'invia come dati seriali dalla cuffia al display remoto ( programma monitor sul PC o display LCD di Arduino).

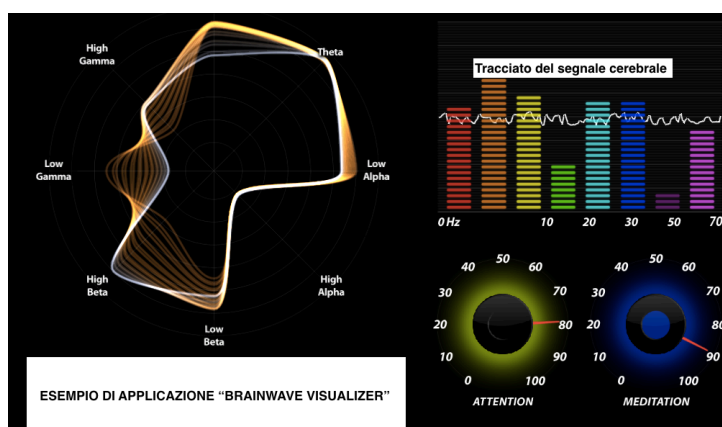
***Si osservi che i valori della misura di attenzione "eSense" non descrivono un numero esatto, ma descrivono gamme di attività di attività cerebrale per questo ha un valore numerico percentuale ( 0-100 %).***

In sostanza attraverso l'algoritmo di **eSense** e quindi il display, il soggetto viene messo in grado di leggere in tempo reale il valore calcolato della sua attenzione (simile alla concentrazione).

**E' stato verificato che il biofeedback ottenuto con la cuffia è legato al soggetto che la indossa e talvolta nei primi istanti di utilizzo può essere deludente nel risultato in quanto è come esercitare un muscolo non molto utilizzato, o meglio, che non ci è strettamente familiare (si pensi per similitudine a qualcosa di simile alla riabilitazione motoria in coloro che hanno subito un trauma o ha avuto un lungo periodo di non attività fisica) e potrebbe richiedere , a seconda dei soggetti, anche notevole tempo prima di ottenere una piena efficienza e consapevolezza del risultato.**

Tuttavia, data l'estrema dipendenza della misura dal soggetto che la indossa, si sono osservati anche casi in cui alcuni soggetti tendono ad ottenere addirittura i migliori risultati con eSense anche quando la utilizzano per la prima volta. Tutto questo è stato frutto di rilievo e di osservazione da parte degli studenti partecipanti.

Nella figura che segue è possibile vedere un tracciato tipico di eSense di Mindwave elaborato da un interfaccia grafica su PC proprio dell'applicazione Brainwave Visualizer data in dotazione con la cuffia da Neurosky utilizzata in tutta la fase di addestramento dei soggetti.



### ***Il sensore eSense "ATTENTION" – Livello di attenzione.***

Il misuratore di attenzione eSense indica l'intensità del livello mentale di "attenzione" di un utente, ed è quello che si verifica durante un'intensa concentrazione del soggetto ovvero durante un'attività mentale diretta e stabile. Come già definito il suo valore varia numericamente da 0 a 100 %. Distrazioni, pensieri vaganti, mancanza di concentrazione o ansia possono abbassare il livello del misuratore di attenzione. È stato osservato che chiudendo gli occhi non si spengono le attività mentali che elaborano le immagini dagli occhi. Quindi chiudere gli occhi è spesso un metodo efficace per aumentare il livello di concentrazione. Distrazioni, pensieri vaganti, ansia, agitazione e, soprattutto, stimoli sensoriali possono alterare momentaneamente i livelli misurati. Per questo motivo è possibile ritenere che i sordi (che in generale hanno totalizzato performance di concentrazione migliori degli udenti) abbiamo la possibilità di utilizzare lo strumento con maggiore affidabilità.

### **PRIMI RISULTATI**

Si è osservato che, trattandosi di una forma attiva di biofeedback, il soggetto che indossa la cuffia proverà all'inizio diverse strategie di utilizzo finché non risconterà un inizio di successo. Considerato che **il cervello sfrutta metodi induttivi per l'apprendimento** sia passivo che attivo assai dipendenti dalle personali reazioni del singolo soggetto, ogni volta che il soggetto risconterà una reazione positiva segnalata da un livello elevato di attenzione più elevato sarà in grado di duplicare l'azione più facilmente con una pratica aggiuntiva migliorando nel tempo le sue performance soggettive (auto-apprendimento). Questa attività è stata costantemente monitorata e documentata dal gruppo di studenti che componeva il team di sperimentazione.

**Per quei ragazzi che hanno difficoltà nella concentrazione attivare il livello di attenzione può essere una strategia vincente ed esercitarla con la cuffia può essere molto interessante:** questa performance è stata osservata e documentata su tutti i soggetti partecipanti e posta a confronto per una migliore comprensione di come sia differente da soggetto a soggetto e di come sia possibile, con l'allenamento, migliorare le proprie performance mediante un controllo attivo delle stesse.

### **APPROFONDIMENTO TECNICO DELLA METODOLOGIA PROPOSTA NEL PROGETTO.**

L'ultimo secolo di ricerca sulle neuroscienze ha notevolmente accresciuto le nostre conoscenze sul cervello e in particolare, sul fatto che i segnali elettrici emessi dai neuroni nel cervello risuonano in esso come delle onde sonore. Le differenti tipologie e le frequenze di questi segnali elettrici possono essere misurate posizionando dei sensori elettrici sul cuoio capelluto. La cuffia utilizzata nel progetto è denominata Mindwave ed è prodotta dalla Neurosky, azienda americana leader nel settore e utilizza una tecnologia proprietaria denominata *Thinkgear*™, che misura in modo accurato i segnali elettrici analogici emessi nel corso dell'attività neurale del cervello comunemente noti come **onde cerebrali**, e li rielabora sotto forma di **segnali digitali**. Questa tecnologia di acquisizione ed elaborazione numerica dei segnali rende possibile utilizzare **tali segnali digitali in giochi ed applicazioni sul computer**.

La tabella seguente fornisce una sinossi generale di alcune delle frequenze più comunemente riconosciute che sono generate nel corso di diversi tipi di stati mentali / attività elettriche nel nostro cervello:

<i>Tipo di onda cerebrale</i>	<i>Intervallo di frequenze</i>	<i>Stati mentali e condizioni</i>
Delta	Da 0,1Hz a 3Hz	Sonno profondo, senza sogni, sonno non REM, attività profonda dell' inconscio
Theta	Da 4Hz a 7Hz	Pensiero Intuitivo, creativo, richiamo, fantasia, immaginario, sogno REM
Alfa	Da 8Hz a 12Hz	Stato rilassato, ma non sonnolento, tranquillo, cosciente, meditazione
Beta Low	Da 12Hz a 15Hz	Stato Precedente SMR, rilassato ma concentrato,
Beta Mid	Da 16 Hz a 20 Hz	Pensiero consapevole di sé e dintorni, immaginazione
Beta High	Da 21Hz a 30Hz	Stato di Allerta, Agitazione, pensiero fobico

Lo studio dell'impiego della cuffia su **persone affette da sordità, come quello che si vuole proporre, è sicuramente innovativo** ( non esiste molta letteratura scientifica in merito anche per la notevole originalità del dispositivo) e ha dimostrato ( seppure in forma ancora assai preliminare durata la breve durata del progetto) che potrebbe portare a risultati inaspettati in termini di biofeedback considerando che i sordi hanno il canale uditivo naturalmente attenuato e quindi meno perturbante rispetto ai soggetti udenti ( che verranno pure coinvolti a confronto) .

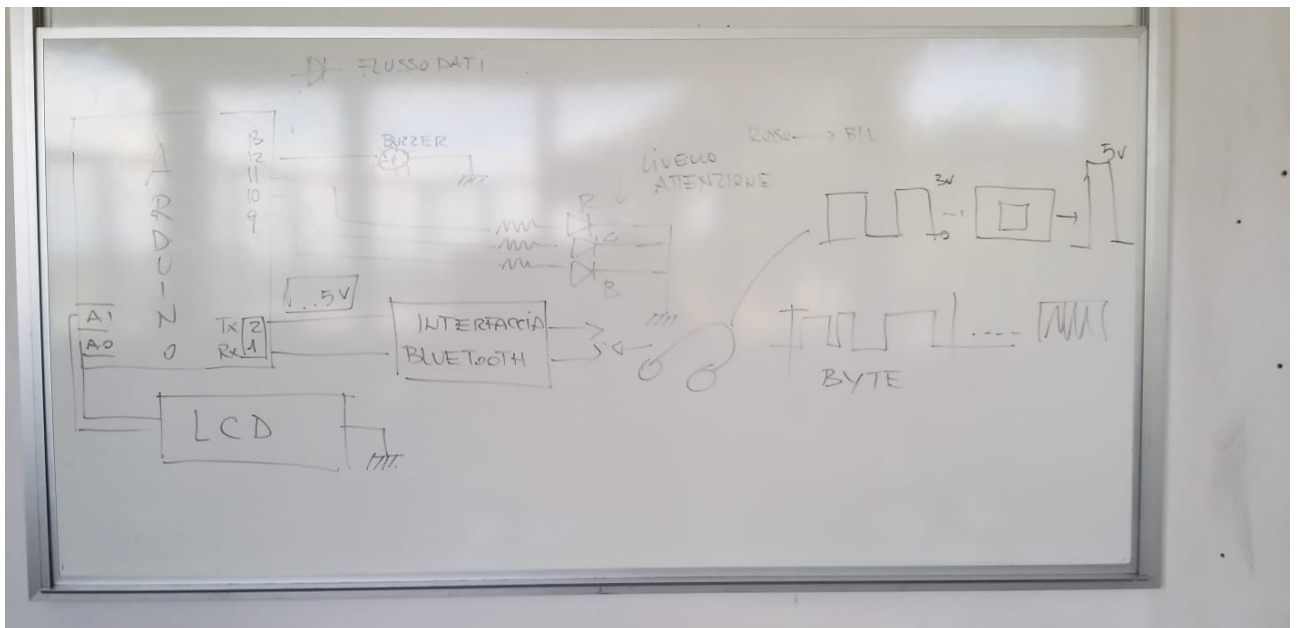
**La sperimentazione è stata condotta sempre enfatizzandone la sua componente ludica** (sicuramente migliorativa a fini dell'apprendimento degli studenti) **e sempre coadiuvata da ampie spiegazioni introduttive su quanto sperimentato e realizzato** di volta in volta.

L'impiego della piattaforma ARDUINO, come abbiamo accennato non essenziale ai fini dell'elaborazione digitale del segnale cerebrale, è stata scelta principalmente da me scelta come veicolo ai fini della comprensione delle attuali interfacce BCI ( Brain Computer Interface), interfacce che a breve popoleranno il mondo digitale e per sperimentare in modo diretto e proattivo, soprattutto da un punto di vista funzionale ed elettrico, come avviene ( soprattutto per gli studenti IPSIA che hanno partecipato) il trattamento di segnali digitali provenienti da sensori del tipo HI (Human Interface) di tipo digitale come appunto Mindwave.

Per i soggetti sordi (ma anche per gli udenti) questo aspetto è stato sicuramente fonte di notevole di curiosità e stimolo all'approfondimento di quanto sta avvenendo in ambito scientifico sull'evoluzione delle interfacce uomo macchina.

**L'interfaccia ARDUINO realizzata** (vedi foto allegate) non è stata di semplice comprensione e realizzazione da parte dei ragazzi , soprattutto per quanto riguarda la parte di sviluppo SW ARDUINO dedicato, che avrebbe richiesto un maggior tempo di metabolizzazione da parte loro, ma la sua costruzione ha consentito un preventiva attività di sperimentazione sviluppata in alcuni (3) incontri propedeutici in cui si è avuto modo di sperimentare come è possibile monitorare con ARDUINO -

tramite l'impiego di un visualizzatore di tipo analogico e/o digitale ( LCD, LED RGB) - l'attività cerebrale in termini di sensore eSense di tipo A (Attention).



SCHEMA DI PRINCIPIO DELL'INTERFACCIA ARDUINO

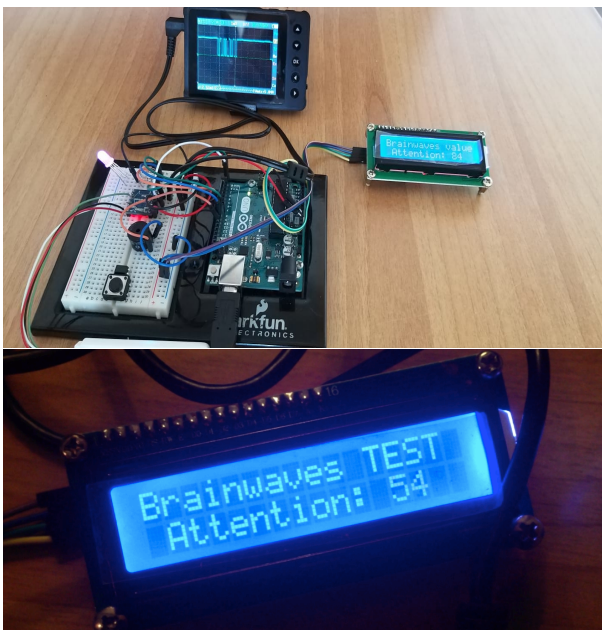


IMMAGINE DEL PROTOTIPO REALIZZATO DI INTERFACCIA SU ARDUINO

La realizzazione del prototipo è stata comunque totalmente sviluppata nel corso dei tre incontri finali tenutisi nel corso del mese di maggio 2019 ed ha portato ad un risultato finale in linea con quanto previsto nella programmazione iniziale del progetto presentato. La partecipazione all'evento ROME CUP 2019 è stata invece estremamente efficace ed ha portato ad una piena comprensione da parte loro dei meccanismi fisiologici implicati nell'uso della cuffia neurale come interfaccia BCI .

Si auspica il proseguo del progetto nel corso del prossimo anno scolastico ( 2019-20) all'interno del progetto PON - IOT ( Internet of Things) già aggiudicato all'istituto MAGAROTTO.

## Immagini tratte dalla partecipazione all'evento "Rome CUP 2019"





**(Preparato da Prof. Ing. Fabio Scanzani ( Docente TEEA IPSIA MAGAROTTO))**

**In Allegato:**

**Sketch IDE in C++ utilizzato nella sperimentazione su piattaforma Arduino.**

```

////////////////////////////////////
// Arduino Interface with Mindwave
// This is example code provided by F.Scanzani and is provided
// license free.
////////////////////////////////////
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// libreria LCD seriale X coll. con 4 fili (collegare A5 a SCL di LCD e A4 a SDA di LCD)
LiquidCrystal_I2C lcd(0x27, 16, 2); // libreria LCD I2C disp 2 righe

const int redPin = 9; // LED RED PIN
const int greenPin = 10; // LED GREEN PIN
const int bluePin = 11; // LED BLUE PIN

const int tinyLedPin = 13; // LED STREAMING
const int speakerPin = 12; // SPEAKER

int tinyLedState = HIGH;

// checksum e data variables
byte generatedChecksum = 0;
byte checksum = 0;
int payloadLength = 0;
byte payloadData[64] = {0};
byte poorQuality = 0;
byte attention = 0;

////////////////////////////////////
// Microcontroller Setup //
////////////////////////////////////

void setup()
{
// inizializzazione delle porte utilizzate per gli indicatori ( LED RGB, SPEAKER)
pinMode(redPin, OUTPUT);
pinMode(greenPin, OUTPUT);
pinMode(bluePin, OUTPUT);
pinMode(tinyLedPin, OUTPUT);
pinMode(speakerPin, OUTPUT);

// Inizializzazione delle comunicazioni seriali
Wire.begin();// inizializza l'uso della comunicazione I2C
lcd.begin(); // inizializza l'uso del display LCD come output di dati
Serial.begin(115200); // inizializza la connessione seriale del dongle-cuffia

// inizializzazione del display LCD
lcd.clear();
lcd.backlight();
lcd.setCursor(0,0);
lcd.print ("BrainWavesLights");
lcd.setCursor(0,1);
lcd.print("a Think Vers.01");

// inizializzazione della cuffia Brainwaves
connectHeadset(); // comando di connessione alla cuffia B.

// qui finisce SETUP

```



```
// Programma principale ( LOOP)
```

```
void loop()
{
    int att = getAttention(); // legge l'attenzione dalla cuffia e l'assegna alla variabile "att"

    if (att > 0) // se il livello di attenzione è maggiore di 0
    {setBlueToRed(att); // assegna un colore tra rosso (0%) e blu (100%)
    toggleTinyLed(); // fai lampeggiare il led 13 quando invia un pacchetto

    // Stampa il valore dell'attenzione
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print ("Brainwaves value ");
    lcd.setCursor(0,1);
    lcd.print(" Attention: ");
    lcd.print (att, DEC);
    }
}
```

```
// fine programma principale (LOOP)
```

```
/** funzioni utilizzate dal programma principale **/
```

```
void connectHeadset() // connetti la cuffia
{
    delay(3000); // aspetta tre secondi
    Serial.write(0xc2); // CONNETTE IL DONGLE
    sayHeadsetConnected(); // fai sentire che la cuffia si è connessa
    setRed(); // led rosso ( non connesso)
}
```

```
byte readOneByte() // leggi il pacchetto seriale
{
    while (!Serial.available()) { // leggi tutti i dati disponibili
        delay(1); //aspetta 5 ms ognuno
    };
    return Serial.read();
}
```

// questa è la funzione che legge il valore dell'attention (valore compreso tra 0.0 e 1.0); se dato negativo (-1, -2...-70 ) significa che c'è un errore e scarta il pacchetto

```
float getAttention() // funzione di lettura del pacchetto
```

// questa è la funzione che legge il valore dell'attention (valore compreso tra 0.0 e 1.0); se dato negativo (-1, -2...-70 ) significa che c'è un errore e scarta il pacchetto

```
{
    byte generatedChecksum = 0; // Inizializza le variabili utilizzate
    byte checksum = 0;
    int payloadLength = 0;
    byte payloadData[64] = {0};
    int poorQuality = 0;
    float attention = 0;
    Serial.flush(); // previene il riempiment del buffer seriale ( memoria di appoggio)

    /* legge il Sincronismo */
    if (170 != readOneByte()) return -1; // legge i due caratteri di sincronismo (AA . AA)
    if (170 != readOneByte()) return -1;

    /* legge la lunghezza del pacchetto */
}
```

```

payloadLength = readOneByte();
if (payloadLength > 169) return -2;    // Legge la lunghezza in Byte del "pacchetto dati"

/* legge il Checksum del pacchetto*/
generatedChecksum = 0;
for (int i = 0; i < payloadLength; i++) {    //
    payloadData[i] = readOneByte();    // Carica il pacchetto dentro un array di memoria
    generatedChecksum += payloadData[i]; // ne calcola il checksum
}

// verifica della correttezza del check sum
generatedChecksum = 255 - generatedChecksum; // fa il complemento a 255 del checksum
checksum = readOneByte(); // legge il checksum inviato nel pacchetto
if (checksum != generatedChecksum) return -3; // se non sono uguali esce e da errore

/* lettura dei dati contenuti nel pacchetto*/
for (int i = 0; i < payloadLength; i++) {    // esegue la lettura sequenziale dei codici dati

    switch (payloadData[i]) {

        case 0xD0:    // se trova il codice 0xD0 la cuffia è connessa correttamente
            sayHeadsetConnected();
            break; // contaciclo
        case 4:    // se trova il codice 04 legge il dato di attenzione
            i++;    // passa al successivo dato
            attention = payloadData[i]; // assegna alla variabile attention il valore letto
            break;

        case 2:// se trova il codice 02 legge la qualità del dato di attenzione
            i++;// passa al successivo dato
            poorQuality = payloadData[i];
            if (200 == poorQuality) {
                setYellow();    // se la qualità nn è buona mette il led su giallo
                return -4;
            }
            break;
        case 0xD1: // non trova la cuffia
        case 0xD2: // la cuffia si è disconnessa
        case 0xD3: // la cuffia ha rifiutato di dare il dato
        case -70: // fallimento della lettura
            wave(speakerPin, 900, 500);
            setRed();
            return -5;
            break; // reinizia da capo

        case 0x80:    // salta i 3 byte di dati RAW
            i = i + 3;
            break;// reinizia da capo
        case 0x83:    // salta i 25 byte dati di EEG_POWER
            i = i + 25;
            break; // reinizia da capo

    } // switch finisce qui

} // il ciclo di lettura dati pacchetto finisce qui

return (float) attention ;    // esce dalla funzione di lettura dei dati dalla cuffia
}

/** fine del programma di lettura del pacchetto */

```

```
/** Funzioni usate dal programma **/
```

```
void setBlueToRed(float redPercent)
{
    int red = redPercent * 255;
    int blue = (1 - redPercent) * 255;
    setColor(red,0,blue);
}
void setGreen()
{
    setColor(0, 255, 0);
}
void setYellow()
{
    setColor(255, 255, 0);
}
void setRed()
{
    setColor(255, 0, 0);
}
void setWhite()
{
    setColor(100, 100, 100);
}
void sayHeadsetConnected()
{
    setGreen;
    wave(speakerPin, 440, 40);
    delay(25);
    wave(speakerPin, 300, 20);
    wave(speakerPin, 540, 40);
    delay(25);
    wave(speakerPin, 440, 20);
    wave(speakerPin, 640, 40);
    delay(25);
    wave(speakerPin, 540, 40);
    delay(25);
}
void setColor(int red, int green, int blue) // funzione che crea i colori con RGB
{
    analogWrite(redPin, red);
    analogWrite(greenPin, green);
    analogWrite(bluePin, blue);
}
void toggleTinyLed() // funzione che fa lampeggiare il led quando arriva un pacchetto
{
    tinyLedState = !tinyLedState;
    digitalWrite(tinyLedPin, tinyLedState);
}
void wave(int pin, float frequency, int duration) // funzione che serve per far suonare il buzzer
{
    float period = 1 / frequency * 1000 * 1000; // microseconds
    long int startTime = millis();
    while (millis() - startTime < duration) {
        digitalWrite(pin, HIGH);
        delayMicroseconds(period / 2);
        digitalWrite(pin, LOW);
        delayMicroseconds(period / 2);
    }
}
```