



PULSEGUARD

Simone Trapani
Mattia Massara

5B ROB

INTRODUZIONE.....	3
COMPONENTI PRINCIPALI.....	4
<i>AD8232.....</i>	<i>4</i>
<i>MAX30102.....</i>	<i>5</i>
<i>TFT LCD TOUCH.....</i>	<i>6</i>
<i>ESP32.....</i>	<i>7</i>
STRUMENTISTICA.....	8
COMPONENTI E PROGRAMMI.....	8
<i>Strumentistica/Materiali :.....</i>	<i>8</i>
<i>Componenti PCB :.....</i>	<i>8</i>
<i>Software/Programmi :.....</i>	<i>9</i>
IL CIRCUITO STAMPATO.....	10
<i>PULSEGUARD FRONT PCB.....</i>	<i>11</i>
<i>PULSEGUARD BACK PCB.....</i>	<i>11</i>
REALIZZAZIONE.....	12
1. <i>MONTAGGIO.....</i>	<i>12</i>
2. <i>PROGRAMMAZIONE ESP32.....</i>	<i>14</i>
3. <i>PROGRAMMAZIONE MAX30102.....</i>	<i>15</i>
4. <i>PROGRAMMAZIONE TFT LCD 3.5 INCH.....</i>	<i>16</i>
PULSEGUARD BOX.....	23
SITO INTERNET.....	25
CONCLUSIONI.....	33

INTRODUZIONE

Ai giorni d'oggi, le discipline di medicina ed elettronica cooperano spesso insieme per soddisfare le esigenze di tutti coloro che soffrono di una malattia, o che semplicemente hanno il bisogno costante di monitorare la propria salute.

L'idea alla base del nostro progetto è coniugare la tecnologia dei nostri giorni, con la salute, sviluppando soluzioni innovative che possano contribuire al monitoraggio costante della salute del paziente.

Le passioni dei due inventori si legano nel progetto "PulseGuard", un dispositivo all'avanguardia in grado di monitorare costantemente la salute del nostro paziente tramite misure di Elettrocardiogramma e Saturazione Sanguigna.

Ideato per essere semplice, intuitivo e di dimensioni ridotte, è affiancato dall'omonimo sito internet, ove è possibile reperire tutte le informazioni utili al fine di utilizzarlo, e in caso di necessità, il modulo di contatto diretto, permetterà di comunicare direttamente con un operatore via e-mail.

Il progetto presentato è solo l'inizio di un lungo cammino.

PulseGuard continuerà ad essere migliorato per arrivare ad essere un dispositivo utilizzabile in caso di bisogno ; il prossimo step sarà quello di identificare ogni singolo dispositivo creato con un nome specifico, posto dal produttore su ogni scatola, e mettere in diretta comunicazione ognuno di essi con un centro medico, in modo tale da monitorare con costanza i pazienti che ne stanno facendo uso, e intervenire, se necessario.

COMPONENTI PRINCIPALI

Gli ideatori di PulseGuard, dopo una ricerca approfondita e dopo aver riflettuto su quale fosse la strada migliore da percorrere per ottenere un lavoro di qualità, hanno scelto come componenti principali l' AD8232, un sensore MAX30102 e un display Touch Screen TFT LCD da 3.5 inch, il tutto programmato poi con il microcontrollore ESP32.

AD8232

L'AD8232 è un circuito integrato prodotto da Analog Devices, progettato per il condizionamento del segnale bipotenziale, in particolare per applicazioni di tipo ECG a derivazione singola. L'elettrocardiogramma a derivazione singola, a differenza di quello a 12 derivazioni, non permette il completo monitoraggio del cuore, ma è comunque ottimo per un monitoraggio costante e continuo.

Esso è stato ideato per l'utilizzo in dispositivi portatili o indossabili, grazie ad un bassissimo consumo energetico e alle dimensioni particolarmente ridotte (3.5cm * 3.0cm).

Il suo compito è quello di prelevare, amplificare e filtrare il segnale proveniente da elettrodi superficiali applicati sul corpo umano, tipicamente attraverso un sistema a 3 elettrodi. Ciò permette di ottenere un segnale pulito e stabile, pronto per essere digitalizzato tramite un convertitore analogico-digitale (ADC) o elaborato direttamente da un microcontrollore.

Internamente è composto da :

- amplificatore da strumentazione che rappresenta il cuore del circuito, la cui funzione è quella di amplificare differenzialmente il segnale proveniente dagli elettrodi.



- filtri attivi passa-basso e passa-alto, che consentono la soppressione del rumore ad alta e bassa frequenza, creando una banda passante perfetta per i segnali ECG , tra 0,5 Hz e 40Hz .
- Lead-off detection, ovvero un circuito dedicato alla verifica della corretta connessione degli elettrodi al paziente. In caso di disconnessione, il sistema segnala l'anomalia attraverso un uscita digitale.
- reference-buffer di uscita che fornisce un'interfaccia a bassa impedenza tra il segnale amplificato e l'ADC del microcontrollore, minimizzando la distorsione del segnale

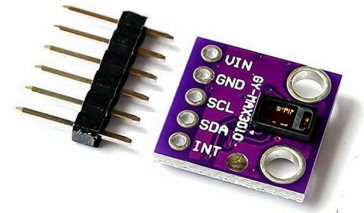
MAX30102

Il MAX30102 è un sensore progettato per il monitoraggio ottico di parametri fisiologici come la frequenza cardiaca e la saturazione di ossigeno nel sangue (SpO2). Questo sensore è principalmente destinato ad applicazioni indossabili, portatili o medicali a basso consumo, dove è necessario effettuare misure non invasive e in tempo reale.

Le misurazioni di vengono prese grazie alla tecnologia fotopletismografia, che consiste nell'illuminare la pelle con un LED infrarosso e luce rossa, e nel misurare la quantità di luce riflessa o assorbita dal tessuto sottostante. Poiché il sangue assorbe la luce in modo diverso a seconda della lunghezza d'onda, il sensore può rilevare variazioni nella quantità di sangue presente nei capillari, legate al battito cardiaco e al livello di ossigenazione dell'emoglobina.

Internamente è composto da:

- LED a luce rossa (660 nm) : utilizzato per la rilevazione della SpO2
- LED a infrarosso (880 nm) : utilizzato per la rilevazione del battito cardiaco e per la misurazione dell'assorbimento ottico
- Fotodiodo : rileva la luce riflessa dalla pelle e converte il segnale ottico in uno elettrico
- Amplificatore e filtro analogico : servono semplicemente per condizionare il segnale ottico
- ADC : converte il segnale da analogico a digitale
- I2C : utile al fine di comunicare direttamente con ESP32



TFT LCD TOUCH

Il display TFT LCD Touch Screen è utilizzato per realizzare interfacce utente grafiche interattive.

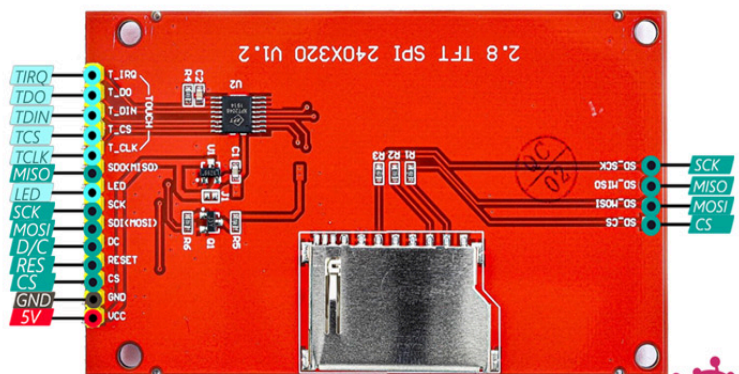
Esso è un tipo di schermo a cristalli liquidi in cui ogni pixel è controllato da transistor a film sottile, consentendo un controllo individuale dell'intensità luminosa e del colore con alta precisione. A differenza dei semplici dispositivi LCD, sono migliori in termini di contrasto, velocità di risposta, risoluzione e luminosità, ma il loro vero punto forte è la presenza di uno strato touch capacitivo o resistivo, che consente all'utente di interagire direttamente con lo schermo.



Il display è composto da :

- Strato TFT : ogni pixel è pilotato da un transistor individuale che ne regola l'attivazione
- Pannello LCD : responsabile della generazione delle immagini
- Retroilluminazione : fornisce l'illuminazione necessaria per rendere visibile il contenuto dello schermo, ed è costituito da led bianchi
- Controller grafico : serve ad interpretare i segnali digitali provenienti dall'ESP32 e li converte nei segnali di controllo del pannello LCD
- Driver Touch : interpreta i segnali provenienti dallo strato Touch e li converte in coordinate X, Y da trasmettere al sistema host.

PINOUT PER I COLLEGAMENTI



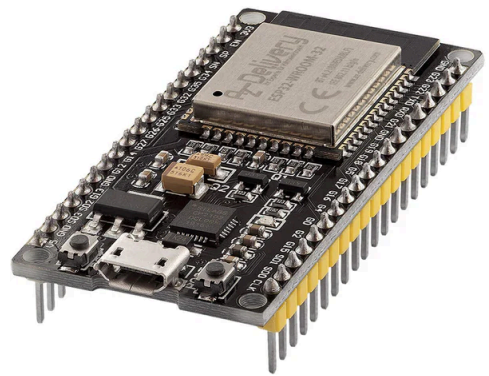
La trasmissione dati, nel nostro progetto, avviene tra ESP32 e il display stesso. Ogni qualvolta l'utente tocca il display o con l'apposita pennina, o con un dito, il sistema di Touch controller rileva il tocco e calcola la posizione in coordinate X , Y ; successivamente l'ESP32, tramite il bus I2C rileva e legge le coordinate, le quali vengono poi usate dal firmware per aggiornare l'interfaccia grafica o gestire eventi, per esempio pulsanti, slider ecc

ESP32

Il ESP32 è un microcontrollore sviluppato da Espressif Systems, progettato per applicazioni che richiedono connettività wireless, elaborazione in tempo reale, e basso consumo energetico.

I componenti principali dell'hardware sono :

- CPU: Dual-core o single-core (a seconda della variante), architettura Xtensa LX6.
- RAM: 520 KB SRAM interna.
- ROM: 448 KB per bootloader e funzioni di sistema.
- Flash: tipicamente da 4 MB a 16 MB esterni, accessibili via SPI.
- GPIO: Fino a 34 linee digitali multifunzione configurabili.
- ADC/DAC: Fino a 18 canali ADC a 12 bit; 2 DAC a 8 bit.
- PWM: Supporto per controllo motori, dimmer, buzzer.
- SPI, I²C, UART, I²S, CAN: Interfacce seriali multiple per comunicazione con periferiche.



Diversi sono i motivi che hanno portato all'utilizzo di questo microcontrollore. Innanzitutto, è compatibile con diverse piattaforme di sviluppo , il team di MSresearch ha optato per l'ambiente software di Arduino IDE ; la compatibilità elettrica con periferiche a 3.3V e l'uso di antenne integrato o esterne per Wi-Fi/bluetooth.

Dopo aver valutato e messo a confronto anche altri microcontrollori, abbiamo optato per scegliere proprio ESP32, così da cimentarci nell'utilizzo di un dispositivo ancora a noi sconosciuto a livello scolastico, ma che ci permetterà, anche in futuro, di portare avanti il nostro progetto

e apportare aggiornamenti molto importanti, come quelli elencati nell'introduzione.

STRUMENTISTICA COMPONENTI E PROGRAMMI

Strumentistica/Materiali :

- ❖ Oscilloscopio
- ❖ Saldatore + Pompa di Aria Calda
- ❖ Stampante 3D
- ❖ Multimetro Digitale
- ❖ Filamento PLA rosso e bianco
- ❖ Pasta Saldante + Stagno
- ❖ Treccia in Rame
- ❖ Pinzette per saldatura SMD
- ❖ Pressa
- ❖ Lente con Luce per la saldatura
- ❖ Ventola Aspira Fumi
- ❖ Flussante per saldatura SMD
- ❖ Alcol Isopropilico
- ❖ Spazzolino per la pulizia della scheda
- ❖ Treppiede per la registrazione dei video
- ❖ SmartPhone
- ❖ Microscopio con Cam

Componenti PCB :

- ❖ Condensatori SMD :
 - 1.5 nF (n.1)
 - 1 nF (n.1)
 - 10 nF (n.1)
 - 0.33 uF (n.2)
 - 0.1 uF (n.2)
 - 1 uF (n.2)

- 10 uF (n.2)
- 22 uF (n.1)
- 100 nF (n.3)
- 100 uF (n.2)
- ❖ Resistori SMD :
 - 10 Mohm (n.6)
 - 1.4 Mohm (n.1)
 - 1 Mohm (n.3)
 - 180 Kohm (n.2)
 - 360 Kohm (n.1)
 - 100 Kohm (n.1)
 - 10 Kohm (n.2)
 - 4.99 Kohm (n.1)
 - 5.1 Kohm (n.2)
 - 1.2 Kohm (n.1)
 - 1 Kohm (n.3)
 - 100 ohm (n.1)
 - 0 ohm (n.3)
- ❖ Induttanze :
 - 1 uH (n.1)
 - 4.7 uH (n.1)
- ❖ ESP32 Wroom
- ❖ Diodi LED_SMD Red, Green
- ❖ Jack di alimentazione 3.5mm
- ❖ Pin 1x14 per display TFT LCD 3.5inch Touch Screen
- ❖ Pin 1x05 per MAX30102
- ❖ Pin 1x05 per Uart (attacco ESP32 per la programmazione)
- ❖ Jack USB-C
- ❖ AD8232
- ❖ MAX30102
- ❖ TFT LCD 3.5 inch Touch Screen

Software/Programmi :

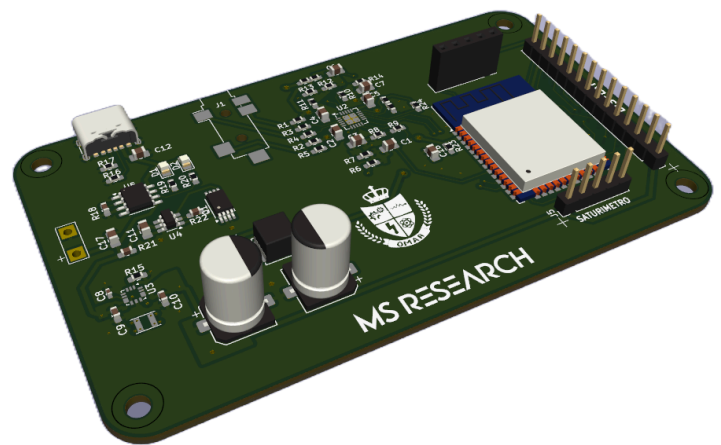
- ❖ OnShape
- ❖ Fusion 360
- ❖ Visual Studio Code
- ❖ Arduino IDE
- ❖ DaVinci

❖ Photoshop

IL CIRCUITO STAMPATO

L'obiettivo principale che avevamo in mente durante la realizzazione della scheda era quello di ottenere un circuito stampato piccolo e compatto, in modo tale da ridurre al minimo anche le dimensioni della PulseGuard Box, l'ultima parte fondamentale da progettare.

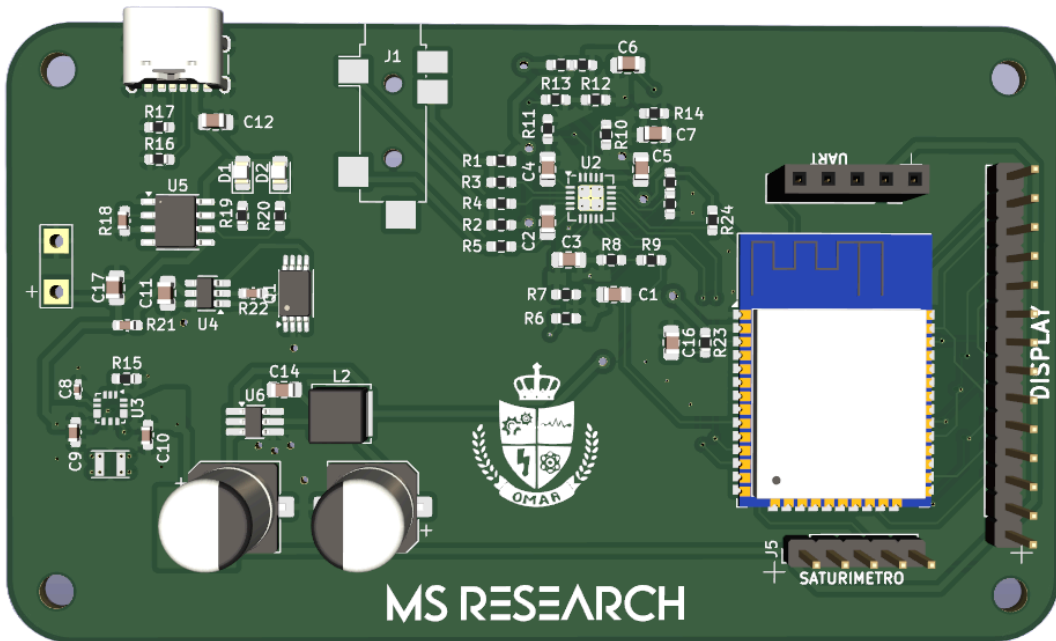
La posizione di ogni singolo componente sulla scheda è stata studiata nel minimo dettaglio per riuscire a diminuire le dimensioni, cercando però di mantenere sempre un'elevata qualità e un ridotto numero di fori passanti. Il voler ottenere una scheda di dimensioni ridotte, minimalista e professionale, ci ha portato a preferire i componenti di tipologia SMD (Surface Mounting Device) a quelli THT, i quali ci avrebbero sicuramente ridotto i tempi di produzione e facilitato l'assemblaggio, ma non ci avrebbero soddisfatto a pieno sotto l'aspetto minimalista e professionale della scheda. PulseGuard, oltre ad essere un dispositivo funzionale e utile alla vita, deve anche essere bello da vedere, e avere un circuito stampato pulito, minimalista e di dimensioni ridotte, curato nei minimi dettagli, era ciò che avevamo in mente sin dall'inizio.



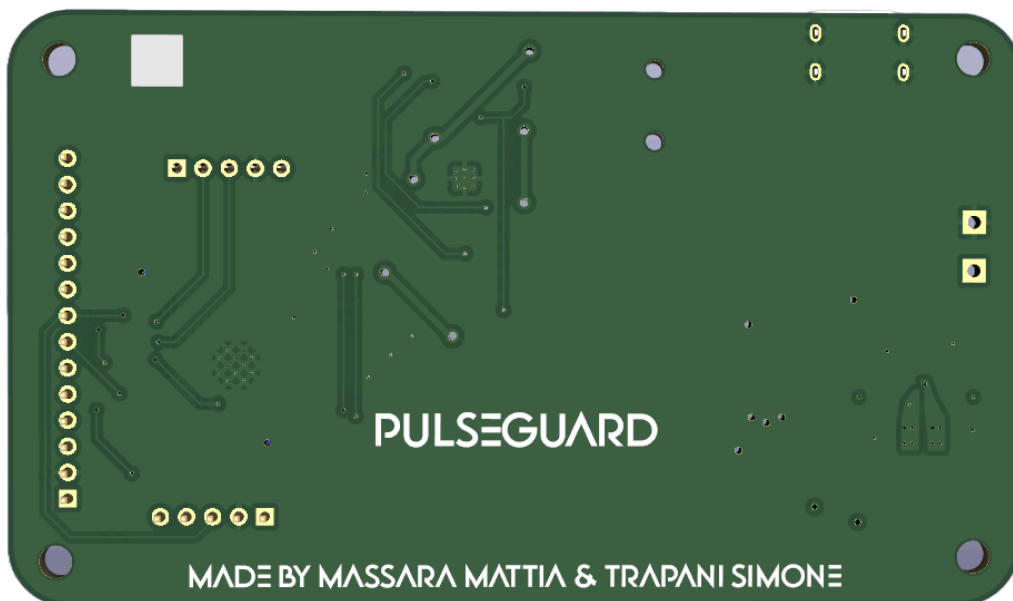
Il circuito stampato è stato realizzato tramite l'utilizzo del Software Kicad, già utilizzato diverse volte in ambito scolastico. Non avendo tutte le impronte necessarie per la realizzazione del PCB, abbiamo provveduto a disegnarne di nuove per diversi componenti.

Dopo un lungo periodo di lavoro, siamo riusciti ad ottenere esattamente ciò che avevamo in mente : la scheda elettrica di PulseGuard è di dimensioni notevolmente ridotte (99.00mm * 58.00 mm).

PULSEGUARD FRONT PCB



PULSEGUARD BACK PCB



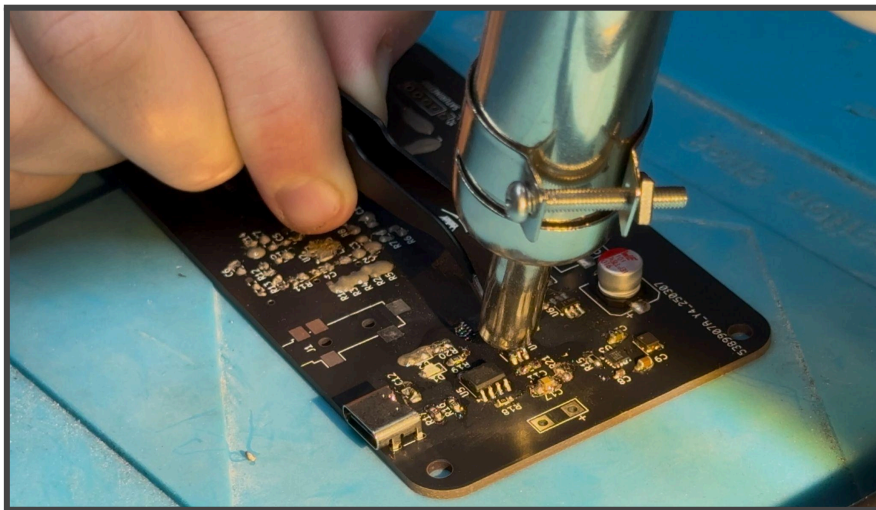
REALIZZAZIONE

Dopo la fase di progettazione, il circuito stampato è stato ordinato tramite il sito [jlcPCB](#), e nei giorni successivi abbiamo ordinato tutti i componenti necessari per procedere con il montaggio effettivo del nostro progetto. L'ordine dei componenti è stato fatto su due diversi siti, [Mouser Electronics](#) e [Aliexpress](#), e dopo diversi giorni di attesa, tutto l'occorrente per l'assemblaggio era a nostra completa disposizione.

1. MONTAGGIO

Venerdì 25 aprile 2025, il nostro progetto prende vita.

Il montaggio ha richiesto diverse ore e molta pazienza. Avendo realizzato una scheda completamente con componenti SMD, molto vicini tra loro e soprattutto di dimensioni notevolmente ridotte, il processo di saldatura ha richiesto tempo e molta collaborazione.

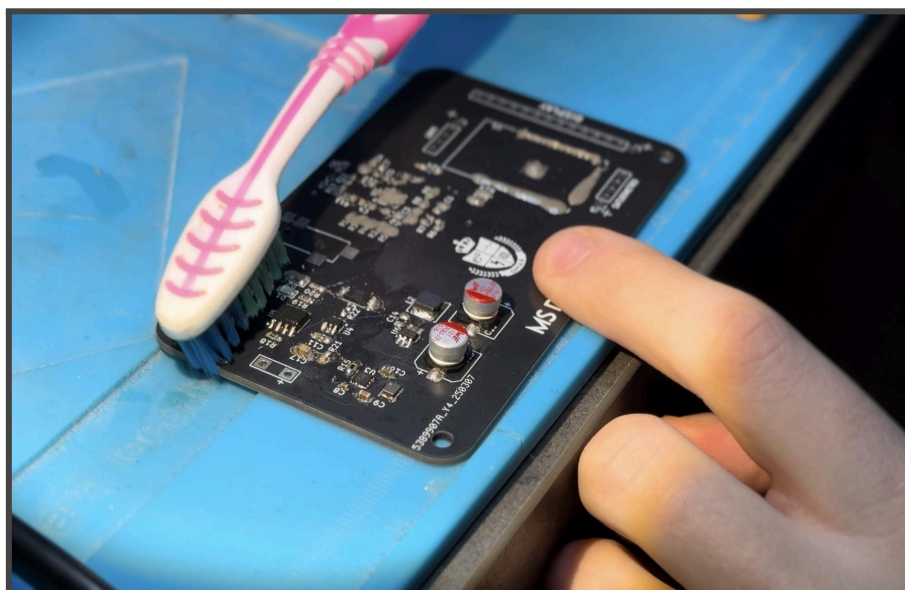


Tutto non finisce con il “semplice” saldare i componenti. Durante il percorso abbiamo riscontrato problemi causati dalla troppa vicinanza dei componenti, ma grazie all'utilizzo della microcamera montata sul microscopio, siamo riusciti ad intravedere ogni minimo errore e rimediare prima del dovuto.



Avere una scheda funzionale è importante, ma ciò che un elettronico guarda oltre al funzionamento, è la cura del proprio lavoro, ed è per questo che abbiamo badato ai minimi dettagli, come la rimozione di ogni singola goccia di stagno in eccesso. Per fare ciò abbiamo utilizzato la treccia di rame, creata appositamente per rimuovere gli eccessi di stagno: appoggiando la treccia sulla piazzuola desiderata e scaldandola con il saldatore, si ha la possibilità di rimuovere lo stagno in eccesso, permettendo così di ottenere una scheda più pulita e curata.

Muniti anche di un semplice spazzolino da denti vecchio, abbiamo utilizzato l'alcool isopropilico per pulire l'intero circuito stampato. Esso è un solvente utilizzato nel mondo dell'elettronica principalmente per rimuovere sporco, grasso e ossidazioni senza lasciare alcun residuo.



2.PROGRAMMAZIONE ESP32

```
void setup() {  
  // initialize the serial communication:  
  Serial.begin(9600);  
  pinMode(26, INPUT); // Setup for leads off detection LO +  
  pinMode(27, INPUT); // Setup for leads off detection LO -  
}  
  
void loop() {  
  
  if((digitalRead(26) == 1) || (digitalRead(27) == 1)){  
    Serial.println('!');  
  }  
  else{  
    // send the value of analog input 0:  
    Serial.println(analogRead(33)+2048);  
  }  
  //Wait for a bit to keep serial data from saturating  
  delay(1);  
}
```

3.PROGRAMMAZIONE MAX30102

```
#include <Wire.h>
#include "MAX30105.h"
#include "heartRate.h"

MAX30105 particleSensor;

float beatsPerMinute;
float beatAvg;
long lastBeat = 0;

void setup() {
  Serial.begin(115200);
  delay(1000);

  Wire.begin(21, 22); // SDA, SCL

  if (!particleSensor.begin(Wire, I2C_SPEED_STANDARD)) {
    Serial.println("MAX30102 non trovato!");
    while (1);
  }

  particleSensor.setup();
  particleSensor.setPulseAmplitudeRed(0x2F); // Aumenta intensità
  LED
  particleSensor.setPulseAmplitudeIR(0x2F);

  Serial.println("Sensore inizializzato. Metti il dito sul sensore.");
}

void loop() {
  long irValue = particleSensor.getIR();

  Serial.print("IR: ");
  Serial.print(irValue);

  if (checkForBeat(irValue)) {
    long delta = millis() - lastBeat;
    lastBeat = millis();
  }
}
```

```
float bpm = 60 / (delta / 1000.0);
if (bpm < 255 && bpm > 30) {
    beatsPerMinute = bpm;
    beatAvg = (beatAvg * 0.9) + (bpm * 0.1);
}

Serial.print("  BPM: ");
Serial.print(beatsPerMinute);
Serial.print("  Avg BPM: ");
Serial.println(beatAvg);
} else {
    Serial.println("  (Nessun battito rilevato)");
}

delay(100);
}
```

4. PROGRAMMAZIONE TFT LCD 3.5 INCH

```
#include <TFT_eSPI.h> // Hardware-specific library
#include <Wire.h>
#include "MAX30105.h"
#include "heartRate.h"
//*****FONTS*****//
#include "Modern_Age10pt7b.h"
#include "Modern_Age20.h"
#include "Modern_Age14.h"
#include "Modern_Age34.h"
#include "Roboto_Bold24.h"
#include "Roboto_Bold18.h"

#define ModernAge10 &Modern_Age10pt7b
#define ModernAge20 &Modern_Age20pt7b
#define ModernAge14 &Modern_Age14pt7b
#define ModernAge34 &Modern_Age34pt7b

#define Roboto18 &Roboto_Bold18pt7b
#define Roboto24 &Roboto_Bold24pt7b
//*****//
```

```

#define BACKLIGHT_PIN 22
#define running 1
#define stop 0
#define D_W 480 // Larghezza del display
#define D_H 320 // Altezza del display

TFT_eSPI tft = TFT_eSPI(); // crea l'oggetto display
#define TOUCH_CALIBRATION 1963, 358, 1589, 193, 1
int selection = 0; // Indica la selezione della pagina

uint16_t white = TFT_WHITE;
uint16_t black = TFT_BLACK;

//*****//
bool firstTimeHome = true; // Definizione flag per determinare se è
gia stata eseguita la funzione MENU
bool firstTimeECG = true; // Definizione flag per determinare se è
gia stata eseguita la funzione MENU
bool firstTimeSpO2 = true;

//*****//

MAX30105 particleSensor;

#define RATE_SIZE 8
byte rates[RATE_SIZE]; //Array of heart rates
byte rateSpot = 0;
long lastBeat = 0; //Time at which the last beat occurred
int SpO2 = 0;
int bpmAVG = 0;
float bpm;
static long lastIR = 0;
static long lastRed = 0;
static unsigned long lastUpdate = 0;

//*****//
void setup() {
  Serial.begin(115200); //inizializzazione comunicazione seriale
  //*****/ACCENSIONE
BACKLIGHT/*****
  pinMode(BACKLIGHT_PIN, OUTPUT); // selezione pin
retroilluminazione
  digitalWrite(BACKLIGHT_PIN, HIGH); // accensione retroilluminazione

```

```
//*****/INIZIALIZZAZIONE
DISPLAY/*****
tft.init();
tft.setRotation(1);          // rotazione orizzontale
tft.fillScreen(TFT_BLACK);  // Sfondo nero

//*****

if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C
port, 400kHz speed
{
  Serial.println("MAX30105 not found.");
  while (1)
    ;
}

particleSensor.setup(0x1F, 4, 2, 100, 411, 4096);
particleSensor.setPulseAmplitudeGreen(0x00); //Turn off Green LED
}

void loop() {

  uint16_t rawx, rawy;
  if (tft.getTouch(&rawx, &rawy)) {
    uint16_t x, y;
    y = map(rawx, 24, 478, 0, 320);
    x = map(rawy, 308, 0, 0, 480);
    Serial.printf("x: %i      ", x);

    Serial.printf("y: %i      \n ", y);
    if (x >= 0 && x <= 40 && y >= 0 && y <= 40) selection = 0;
  }
  switch (selection) {

    case 0:
      home_page();
      break;
    case 1:
      ECG_page();
      break;
    case 2:
      SpO2_page();
```

```
        break;
    }
}

void home_page() {
    firstTimeECG = true;
    firstTimeSpO2 = true;
    if (firstTimeHome) {
        tft.fillScreen(TFT_RED);

        // Titolo piccolo
        tft.setTextColor(white, TFT_RED);
        tft.setTextDatum(MC_DATUM); // Centro orizzontale
        tft.setFreeFont(ModernAge14);
        tft.drawString("MSRESEARCH PRESENTS", 240, 30);

        // Testo "PULSEGUARD"
        tft.setFreeFont(ModernAge34);
        tft.drawString("PULSEGUARD", 240, 120);

        // Box ECG
        drawBoxWithText(80, 200, 120, 60, "ECG", black, white);

        // Box O2
        drawBoxWithText(280, 200, 120, 60, "SpO2", black, white);
        firstTimeHome = false;
    }
    uint16_t rawx, rawy;
    uint16_t x, y;
    if (tft.getTouch(&rawx, &rawy)) {

        y = map(rawx, 24, 478, 0, 320);
        x = map(rawy, 308, 0, 0, 480);
        Serial.printf("x: %i    ", x);

        Serial.printf("y: %i    \n ", y);
    }
    if (x >= 60 && x <= 195 && y >= 200 && y <= 275) selection = 1;
    if (x >= 285 && x <= 445 && y >= 200 && y <= 275) selection = 2;
}
```

```
void ECG_page() {
  firstTimeHome = true;
  if (firstTimeECG) {
    tft.fillScreen(white);
    drawGrid();
    drawHouse(10, 10);
    tft.setFreeFont(ModernAge14);
    tft.setTextColor(TFT_RED);
    tft.setTextDatum(MC_DATUM);
    tft.drawString("ECG", 240, 10);
    firstTimeECG = false;
  }
}

void SpO2_page() {
  firstTimeHome = true;
  if (firstTimeSpO2) {
    tft.fillScreen(white);
    drawHouse(10, 10);
    tft.setFreeFont(ModernAge20);
    tft.setTextColor(TFT_RED);
    tft.drawString("%SpO2", 150, 70);
    tft.drawString("bpm", 350, 70);
    firstTimeSpO2 = false;
  }

  long irValue = particleSensor.getIR();
  long redValue = particleSensor.getRed();

  if (checkForBeat(irValue)) {
    long delta = millis() - lastBeat;
    lastBeat = millis();
    Serial.println("Beat detected");
    bpm = 60 / (delta / 1000.0);
    Serial.println(String(bpm));
    if (bpm > 30 && bpm < 200) {
      rates[rateSpot++] = (byte)bpm;
      rateSpot %= RATE_SIZE;

      bpmAVG = 0;
      for (byte x = 0; x < RATE_SIZE; x++)
        bpmAVG += rates[x];
    }
  }
}
```

```
bpmAVG /= RATE_SIZE;

Serial.print("BPM: ");
Serial.println(bpmAVG);
}
}
// Aggiorna ogni 500 ms
if (millis() - lastUpdate >= 500) {
    lastUpdate = millis();

    if (irValue > 50000 && redValue > 30000) {
        // Stima SpO2
        float ratio = (float)(redValue - lastRed) / redValue;
        ratio /= (float)(irValue - lastIR) / irValue;
        SpO2 = 110 - 25 * ratio;

        if (SpO2 > 100) SpO2 = 100;
        if (SpO2 < 70) SpO2 = 70;

        lastIR = irValue;
        lastRed = redValue;

        // Mostra valori
        tft.setFont(ModernAge14);
        tft.setTextColor(TFT_RED);

        drawBoxWithText(100, 150, 100, 50, String(SpO2), black, white);
        drawBoxWithText(300, 150, 100, 50, String(bpmAVG), black, white);

    } else if (irValue < 20000 && redValue < 10000) {
        tft.setFont(ModernAge14);
        tft.setTextColor(TFT_RED);
        drawBoxWithText(100, 150, 100, 50, "--", black, white);
        drawBoxWithText(300, 150, 100, 50, "--", black, white);
    }
    // Nessun dito rilevato
}
}

void drawHouse(int x, int y) {
    // Corpo casa (larghezza 20, altezza 14)
    tft.fillRect(x, y + 6, 20, 14, black); // Il corpo parte da y+6 per
    lasciare spazio al tetto
}
```

```
// Tetto (triangolo) - altezza 6 pixel
tft.fillTriangle(
  x - 2, y + 6, // punta sinistra
  x + 10, y - 4, // vertice sopra (punto più alto della casa)
  x + 22, y + 6, // punta destra
  black);

// Porta (4x7, centrata orizzontalmente)
tft.fillRect(x + 8, y + 13, 4, 7, white); // y+13 perché il corpo
parte da y+6
}

void drawBoxWithText(int x, int y, int w, int h, String label, uint16_t
textColor, uint16_t bgColor) {
  tft.fillRoundRect(x, y, w, h, 10, bgColor);
  tft.drawRoundRect(x, y, w, h, 10, textColor);
  tft.setTextColor(textColor, bgColor);
  tft.setTextDatum(MC_DATUM);
  tft.setFont(Roboto18); // Font grande per i pulsanti
  tft.drawString(label, x + w / 2, y + h / 2);
}

void drawGrid() {
  tft.setTextColor(TFT_BLACK, TFT_WHITE);

  // Linee verticali ogni 20px
  for (int x = 40; x <= 440; x += 10) {
    tft.drawLine(x, 30, x, 290, TFT_RED);
  }

  // Linee orizzontali ogni 20px
  for (int y = 30; y <= 290; y += 10) {
    tft.drawLine(40, y, 440, y, TFT_RED);
  }
}
```

PULSEGUARD BOX

Ormai siamo quasi al termine del nostro percorso, è arrivato il momento di pensare anche all'estetica del nostro progetto.

Sin dall'inizio abbiamo messo in chiaro di curare molto l'estetica di PulseGuard ; la Box rappresenta infatti il bigliettino da visita del nostro progetto.

Dopo aver sperimentato diversi software per il disegno 3D, abbiamo trovato per noi l'opzione migliore : Onshape, un open source che ti da la possibilità di realizzare disegni 3D e molto altro.

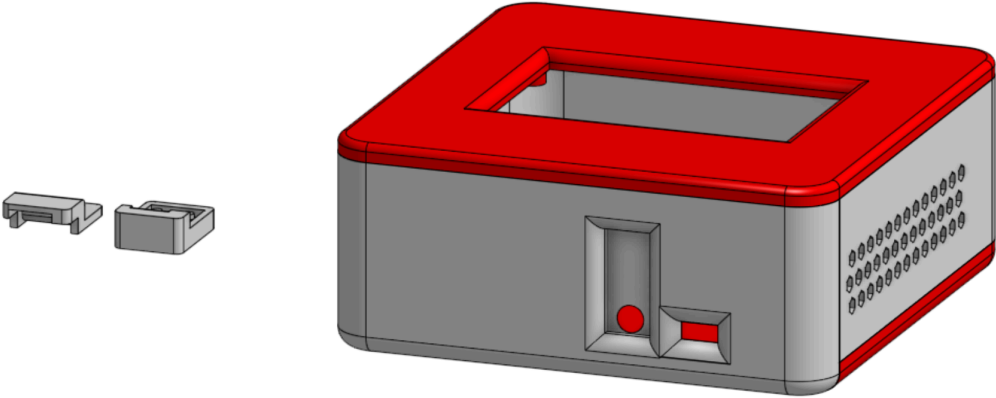
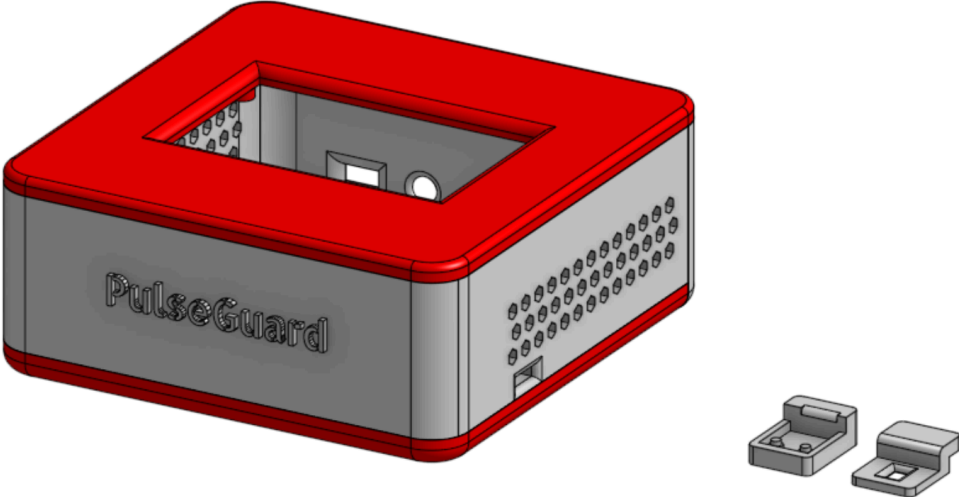
I colori base della PulseGuard Box sono il rosso e il bianco, scelti appositamente perché per noi rappresentano il perfetto connubio tra semplicità e bellezza.

Il design della scatola è stato studiato su misura basandoci sulle dimensioni della scheda elettronica, tenendo conto anche delle diverse aperture da realizzare, come l'attacco degli elettrodi, la porta usb-c e anche il supporto per la misurazione della SpO2.

Sui lati della Box è stato progettato un sistema di fori ad alveare affinché il calore prodotto possa fuoriuscire, evitando quindi un possibile danneggiamento della fragile componentistica dopo svariati utilizzi.

I risultati delle misurazioni verranno poi visualizzati tutti grazie al TFT LCD da 3.5 inch, posizionato sul coperchio della scatola, anch'esso progettato su misura affinché potesse contenere il display.

Internamente, grazie a due strutture ad incastro realizzate su misura, abbiamo la possibilità di fissare sia la scheda, che la batteria.



Di seguito è raffigurata la PulseGuard Box stampata in 3D, con filamenti in PLA di colore bianco e rosso.



SITO INTERNET

Un progetto come PulseGuard, non poteva non essere accompagnato dall'omonimo sito internet.

Creare un sito internet da per noi la possibilità a tutti i futuri utilizzatori di PulseGuard, di poter scoprire di più sulla storia del progetto, di acquisire nuove nozioni leggendo i file sull'elettrocardiogramma e sulla monitorizzazione della saturazione sanguigna, e di poterci contattare direttamente per avere maggiori informazioni.

Il sito web è stato realizzato con Visual Studio Code nell'arco dei mesi del secondo quadrimestre, quando abbiamo iniziato a studiare indipendentemente il linguaggio HTML, CSS e JavaScript. La palette di colori del sito riprende i colori della PulseGuard Box.

Di seguito sarà riportata solo una piccola parte dell' HTML del sito di PulseGuard, in maniera tale da dare un piccolo spunto a tutti coloro che volessero approcciarsi al mondo del Web Design.

```
<!DOCTYPE html>
<!-- HOMEPAGE di PULSEGUARD -->

<html lang="it">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title> PulseGuard.MSresearch </title>
  <link rel="stylesheet" href="style.css">
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.9.1/gsap.min.js"
defer></script>
  <script src="javascript.js"></script>

</head>

<body class="light-theme">
  <header>
    <nav>
      <h1 class="titolo"> PulseGuard </h1>
      <ul>
        <li><a href="html.html">Home</a></li>
        <li><a href="#nascitaprogetto"> Nascita del Progetto </a></li>
        <li><a href="#servizi"> Servizi </a></li>
        <li><a href="#contatti"> Contatti </a></li>
        <li><a href="#tesina"> Tesina </a></li>
        <li><button id="themeToggle">☾</button></li>
      </ul>
    </nav>
  </header>

<main>
```

```

<section id="nascitaprogetto" class="animated">
  <h1 class ="sezione" style="text-align: center;"> Nascita del
Progetto </h1>
  <p>
    <h3 class="TestoImg">
      <section class = animated2 style="margin:0 auto;
text-align: center; width:90%">
        La nostra storia inizia con un sogno: quello di creare un
dispositivo che potesse monitorare la salute dei pazienti in modo
continuo e preciso. <br>
        Dopo mesi di ricerca e sviluppo, siamo orgogliosi di
presentare PulseGuard, il nostro dispositivo innovativo per il
monitoraggio della salute.
        Con PulseGuard, i pazienti possono sentirsi al sicuro e
protetti, sapendo che la loro salute è sempre sotto controllo.
      </h3>
    </p>

    <br>

    <table style="margin: 0 auto; width: 85%;">
      <tr>
        <td>
          
        </td>
        <td>
          <h3 class="TestoImg" style="margin-left:20px;"> <i>
PulseGuard prende vita ufficialmente nel 2025,quando ci siamo riuniti
per la prima
              volta come team di lavoro MSresearch per
assemblare effettivamente il nostro dispositivo e iniziare a testarlo.
            </i> </h3>
        </td>
      </tr>
    </table>

    <table style="margin: 0 auto; width: 85%;">
      <tr>
        <td>
          <h3 class="TestoImg" style="margin-left:20px;">

```

```

        <i> Ogni aspetto del nostro progetto è stato curato
nei minimi dettagli per poter
        ottenere le migliori prestazioni possibili; ci
siamo presi cura di ogni minimo
        dettaglio tecnico ed estetico, come la pulizia
della scheda interna! </i></h3>
    </td>
    <td>
        
    </td>
</tr>
</table>

<table style="margin: 0 auto; width: 85%;">
    <tr>
        <td>
            
        </td>
        <td>
            <h3 class="TestoImg" style="margin-left:20px;">
                <i> Le fasi di assemblaggio della scheda hanno
richiesto diverse ore di lavoro, e di collaborazione.
                Nonostante diversi ostacoli durante il percorso,
siamo riusciti a rimanere molto soddisfatti del
prodotto finale, raffigurato in fondo a questa
pagina. </i></h3>
            </td>
        </tr>
    </table>

<!--
<center>
    <div class="video-container">
        <video controls autoplay muted loop>
            <source src="HTML\FINALE.mp4" type="video/mp4">

        </video>
    </div>
</center> -->
</section>

```

```

<section>
  <h1 class="sezione"> La Scheda Elettrica di PulseGuard </h1>
  <center>
    <table>
      <td style="margin:0 auto; width : 1%" >
        
      </td>

      <td style="margin:0 auto; width : 1%" >
        
      </td>
    </table>

    <br> <br>

    <h3 class="sezioni" style="margin:0 auto; width: 70%; ">
      La scheda di PulseGuard, il cuore del nostro dispositivo, è
      stata progettata con cura per garantire prestazioni ottimali e
      affidabilità.

      Realizzata interamente con componenti SMD, è stata progettata
      per essere il più piccola possibile, senza compromettere
      le prestazioni. La scheda è dotata di un microcontrollore
      ESP32, e utilizza i sensori
      <a target= "_blank" href =
      https://www.alldatasheet.com/datasheet-pdf/pdf/879178/MAXIM/MAX30100.ht
      ml"> <i> <u> MAX30100 </a></i> </u>
      e
      <a target= "_blank" href =
      https://www.alldatasheetit.com/html-pdf/544648/AD/AD8232/23/1/AD8232.ht
      ml"> <i> <u> AD8232 </a></i> </u>

    </h3>
  </center>
</section>

<section id="servizi" class="animated">
  <h1 class="sezione"> Servizi </h1>

```

```

    <h3 class="sezioni" style="text-align: center ; margin:0 auto;
width:70%;">
    PulseGuard offre una gamma di servizi per garantire il
monitoraggio continuo della salute.
    I nostri servizi includono il monitoraggio della frequenza
cardiaca e della pressione sanguigna in tempo reale.
    Con PulseGuard, i pazienti possono sentirsi al sicuro e
protetti, sapendo che la loro salute è sempre sotto controllo.
    </h3>

    <table style="margin: 0 auto; margin-top:25px;">
        <tr>
            <td style="margin:0 auto; width:60%">
                <a href="Ecg.html"> <button type="submit">
Elettrocardiogramma </button> </a>
            </td>
            <td style="margin:0 auto; width:60%">
                <a href="SpO2.html"> <button type="submit">
Saturazione Sanguinia </button> </a>
            </td>
        </tr>
    </table>

</section>

<section id="contatti">
<h1 class="sezione"> Contatta MSresearch </h1>
    <section>

        <h3 class="sezioni" style="text-align:center; width:70%;
margin:0 auto;">
        Per qualsiasi domanda o richiesta di informazioni, non esitare
a contattarci tramite il modulo qui sotto.
        Siamo qui per aiutarti e rispondere a tutte le tue domande
riguardo PulseGuard e i nostri servizi.
        </h3>

        <form id="contattoForm" style="margin:0 auto; width: 70%;
margin-top:45px;">
            <input type="text" id="nome" name="from_name" placeholder="Il tuo
nome" required>

```

```

    <input type="email" id="email" name="from_email" placeholder="La
tua email" required>
    <input type="text" id="oggetto" name="subject"
placeholder="Oggetto" required>
    <textarea id="messaggio" name="message" rows="5" placeholder="Il
tuo messaggio" required></textarea>

    <button type="submit">Invia</button>
    <p id="formRisposta"></p>
</form>

<h2 class="sezioni" style="text-align:center;">
    Salva il nostro indirizzo email per contattarci direttamente:
<br>

    <center>
    <table>
    <tr>
    <td>
        
    </td>
    <td>
        <h3 class="TestoImg"><a target="_blank"
href="mailto:contact.msresearch@gmail.com?subject=oggettomail&body=corp
omail" >
            contact.msresearch@gmail.com </a> </h3>
    </td>
    </tr>
</table>
</center>

</section>

<!-- COLLEGAMENTO CON GOOGLE MAPS CON IL SITO HTML; NEL NOSTRO
CASO COLLEGHIAMO LA POSIZIONE DEL LUOGO DI NASCITA DEL PROGETTO -->
<section>
    <h1 class="sezione"> Dove nasce PulseGuard </h1>
    <center>
    <iframe class="map-wrapper" width="620px" height="470px"

```

```
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d22345.873460
69217!2d8.622814161433181!3d45.565729093705045!2m3!1f0!2f0!3f0!3m2!1i10
24!2i768!4f13.1!3m3!1m2!1s0x47865df191f8ebc7%3A0x76851a8d0a6fae09!2s280
43%20Bellinzago%20Novarese%20NO!5e0!3m2!1sit!2sit!4v1747252979219!5m2!1
sit!2sit"
        loading="lazy"
referrerpolicy="no-referrer-when-downgrade">
        </center>

    </iframe>
</section>
<!------->
</section>
</main>
</body>
</html>
```

I codici CSS e JavaScript non verranno allegati a causa dell'eccessiva lunghezza e a causa dell'innumerabile tempo impiegato per la loro realizzazione.

CONCLUSIONI

30 maggio 2025, il nostro percorso termina.

Dopo circa nove mesi, presentiamo il nostro progetto : PulseGuard.

Ci riteniamo molto soddisfatti di tutto questo percorso, svolto interamente insieme, dalle prime idee fino al primo collaudo.

Abbiamo dovuto affrontare diversi problemi, ma ogni sforzo è stato ricompensato dal risultato finale : ogni obiettivo che ci siamo prefissati all'inizio dell'anno è stato raggiunto se non superato.

Siamo riusciti a realizzare un dispositivo funzionale, compatto e semplice da utilizzare, accompagnato dall'omonimo sito internet e da un video raffigurante l'intero percorso fatto in questo anno scolastico.

PulseGuard ci ha dato la possibilità di unire le nostre passioni, le nostre conoscenze e ci ha fatto capire veramente quali sono le nostre potenzialità per affrontare il futuro in maniera decisa.

Gli sforzi fatti durante tutti questi mesi e i suggerimenti ricevuti per rendere il nostro dispositivo ancora più all'avanguardia, ci porteranno sicuramente a migliorare il nostro progetto e non abbandonarlo ad un semplice ricordo di un lungo e complesso triennio di scuola superiore.

Trapani Simone e Massara Mattia