
Università degli Studi di Padova – Dipartimento di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria Biomedica

Relazione per la prova finale
**«Regolazione in real-time dell'intensità
luminosa per misure
fotopletismografiche»**

Relatrice: Prof.ssa Giada Giorgi
Padova, 20/07/2026

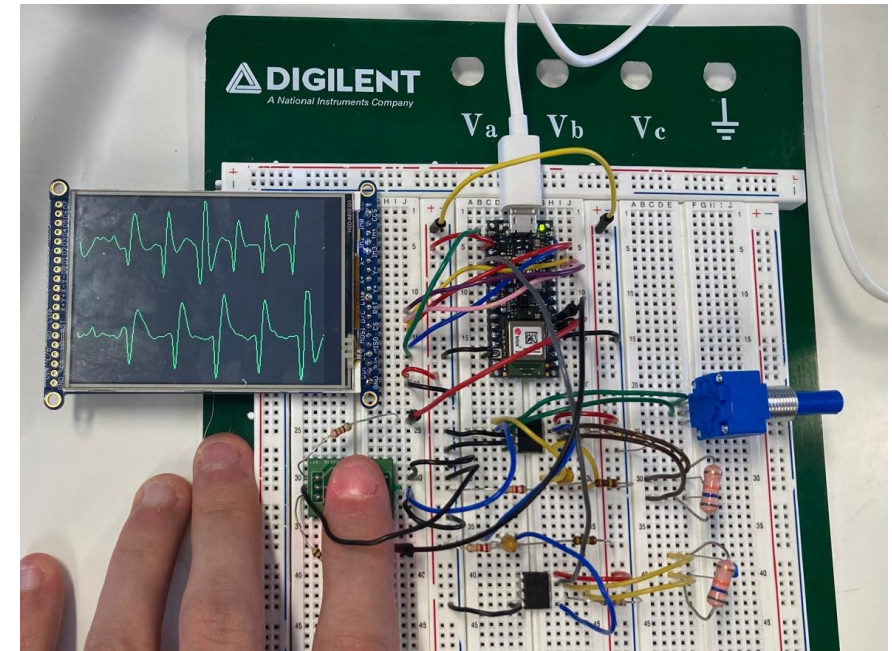
Laureando: Matteo Pinaffo
Matricola n.: 2110268

BACKGROUND

Che cos'è un sensore PPG?

Un sensore PPG è un sensore ottico, in grado di misurare variazioni volumetriche del sangue in tessuti periferici. Permette di misurare:

- **battito cardiaco;**
- **saturazione dell'ossigeno (SpO_2)**
- **curva pletismografica;**
- ...



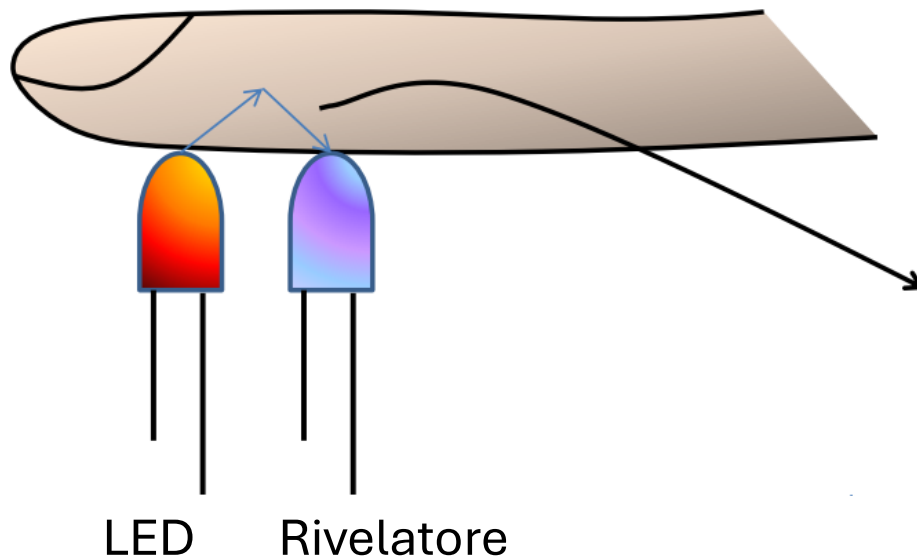
BACKGROUND

Come funziona un sensore PPG?

Un sensore PPG principalmente è composto da:

- una **sorgente luminosa**, che illumina il tessuto con una lunghezza d'onda prestabilita;
- un **rivelatore**, che è in grado di convertire un segnale luminoso in una differenza di potenziale;

Esempio di pulsossimetria a riflessione



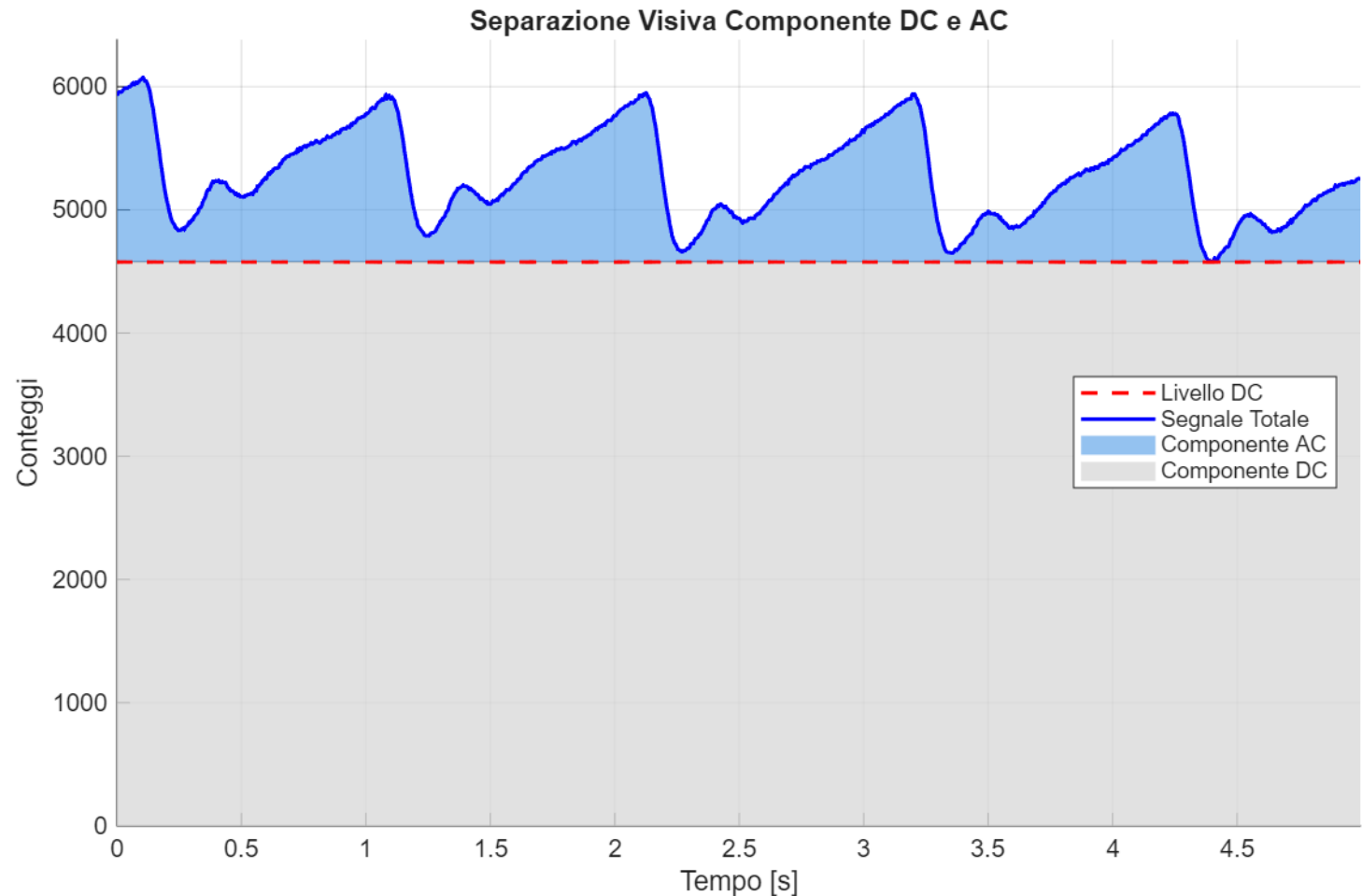
L'intensità luminosa che viene riflessa dal dito è correlata alla variazione del volume del sangue. Il rivelatore registra continuamente tali oscillazioni.

BACKGROUND

Il segnale ottenuto dipende dalle proprietà ottiche del tessuto.

Possiamo distinguere due componenti del segnale:

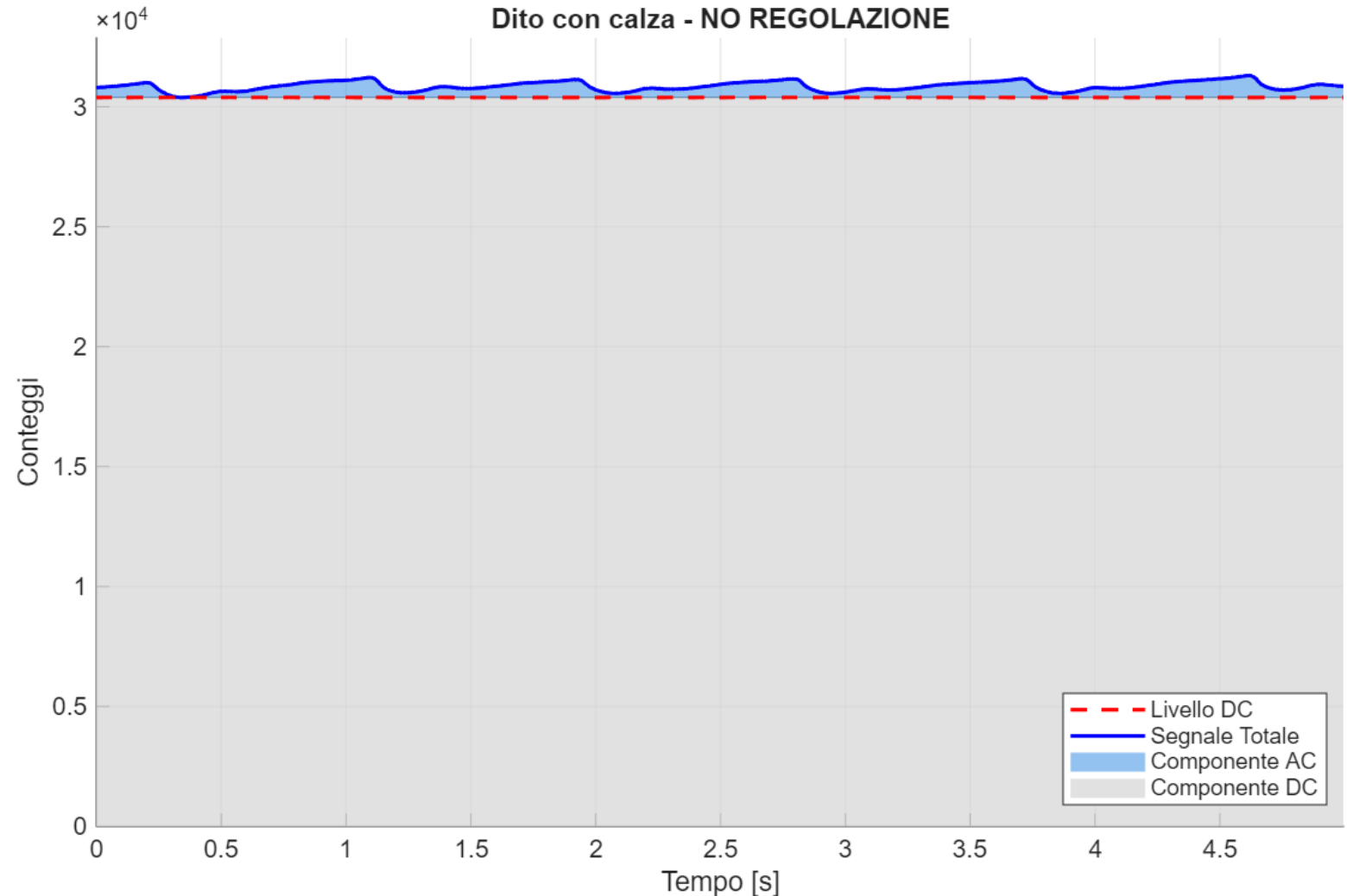
- una **componente costante nel tempo (DC)**, dovuta a epitelio, vasi sanguigni e sangue;
- una **componente variabile nel tempo (AC)**, dovuta alla variazione di volume sanguigno.



PROBLEMI DEI SENSORI PPG

Il segnale registrato non dipende esclusivamente dalla variazione di volume sanguigno del tessuto, ma può essere influenzato da diverse **variabili esterne**, tra cui:

- il **movimento** del paziente
- la **pressione** applicata dal tessuto al sensore;
- la **luce ambientale**;
- la presenza di elementi di disturbo **all'interfaccia cute-sensore**.



SOLUZIONE PENSATA

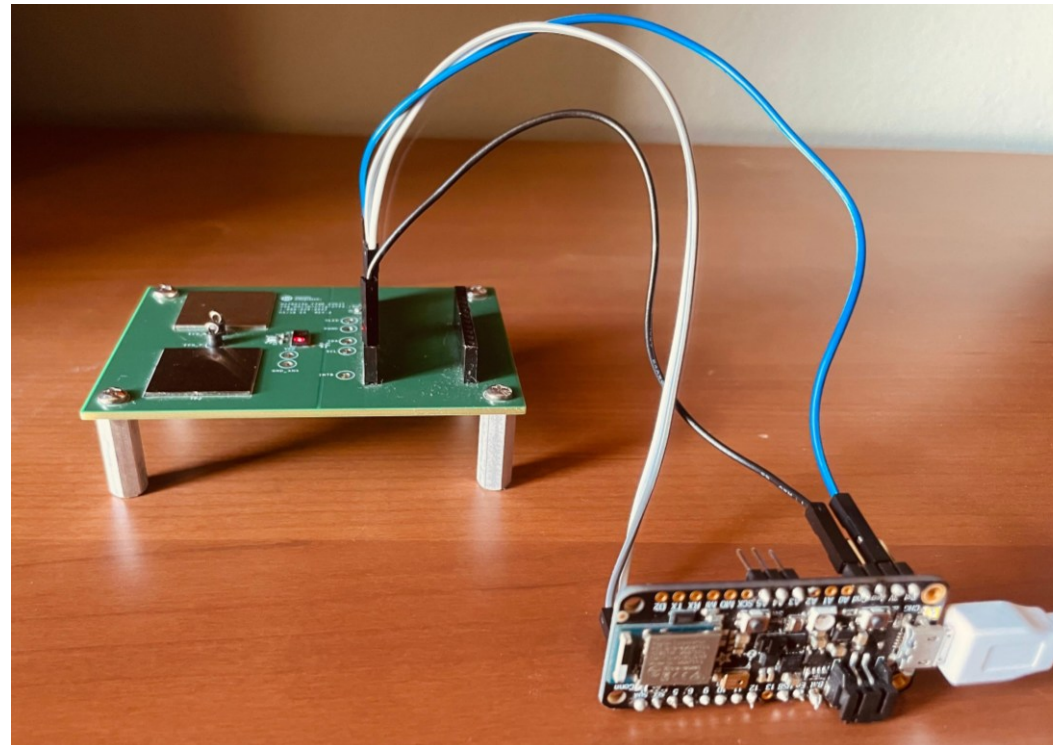
La soluzione pensata è la realizzazione di un meccanismo di feedback:



SETUP SPERIMENTALE

Il setup utilizzato comprende:

- il microcontrollore **Adafruit Feather Bluefruit Sense**, compatibile con l'ambiente di programmazione C++ «Arduino IDE»;
- la scheda **MAX86150**, un modulo sensore integrato per elettrocardiogramma, pulsossimetro e monitoraggio cardiaco.



PROGRAMMA DELL'ALGORITMO DI REGOLAZIONE

```
1 // librerie
2 #include <Wire.h>
3 #include "max86150.h"
4
5 // variabili
6 #define M 50//dimensione buffer
7 #define maxT 100//soglia di innalzamento corrente
8 #define minT 5
9 #define stdAmp 50 //valore di tensione standard
10 #define highAmp 100 //valore di tensione aumentato
11 #define testPin 10
12 uint16_t buffer_memoria[M];
13 int idxBuffer=0;
14
15 int16_t ecgsigned16;
16 uint16_t ppgunsigned16;
17
18 bool regolazione=false;
19
20 MAX86150 max86150Sensor;
21
```

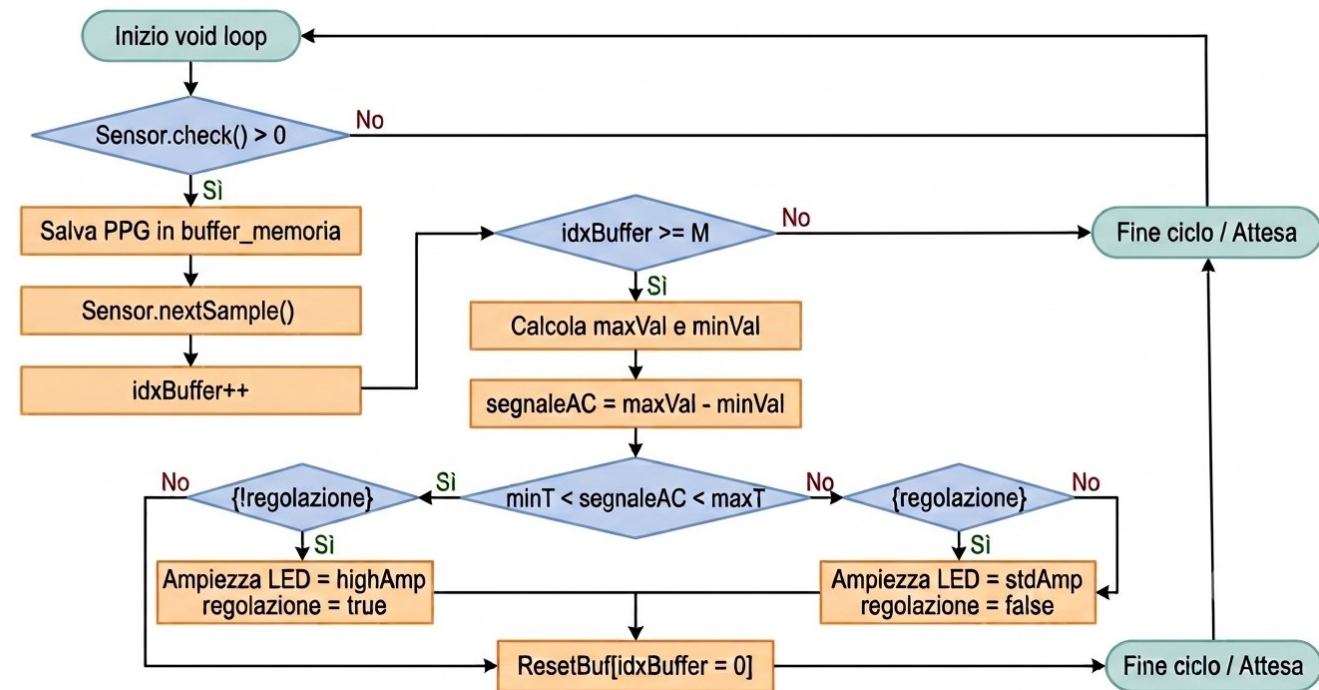
```
22 void setup() {
23     // Configurazione interfaccia seriale
24     Serial.begin(57600);
25     while (!Serial) delay(10);
26
27     // Inizializzazione sensore
28     if (max86150Sensor.begin(Wire, I2C_SPEED_STANDARD) == false) {
29         while (1);
30     }
31     // Configurazione sensore
32     max86150Sensor.setup();
33     max86150Sensor.setPulseAmplitudeRed(stdAmp);
34 }
35
```



PROGRAMMA DELL'ALGORITMO DI REGOLAZIONE



```
40 void loop() {
41   if (max86150Sensor.check() > 0) {
42     buffer_memoria[idxBuffer]=PPG();
43     max86150Sensor.nextSample();
44     idxBuffer++;
45     if (idxBuffer>=M){
46       digitalWrite(testPin, HIGH);
47       uint16_t maxVal=findMax(buffer_memoria);
48       uint16_t minVal=findMin(buffer_memoria);
49       uint16_t segnaleAC= maxVal - minVal ;
50       if(segnaleAC<maxT && segnaleAC>minT){
51         if (!regolazione){
52           max86150Sensor.setPulseAmplitudeRed(highAmp);
53           Serial.println("Ampiezza aumentata!");
54           regolazione=true;
55         }
56       }
57     } else {
58       if (regolazione){
59         max86150Sensor.setPulseAmplitudeRed(stdAmp);
60         Serial.println("Ampiezza diminuita!");
61         regolazione=false;
62       }
63     }
64     idxBuffer=0;
65     digitalWrite(testPin, LOW);
66   }
67 }
68 }
```

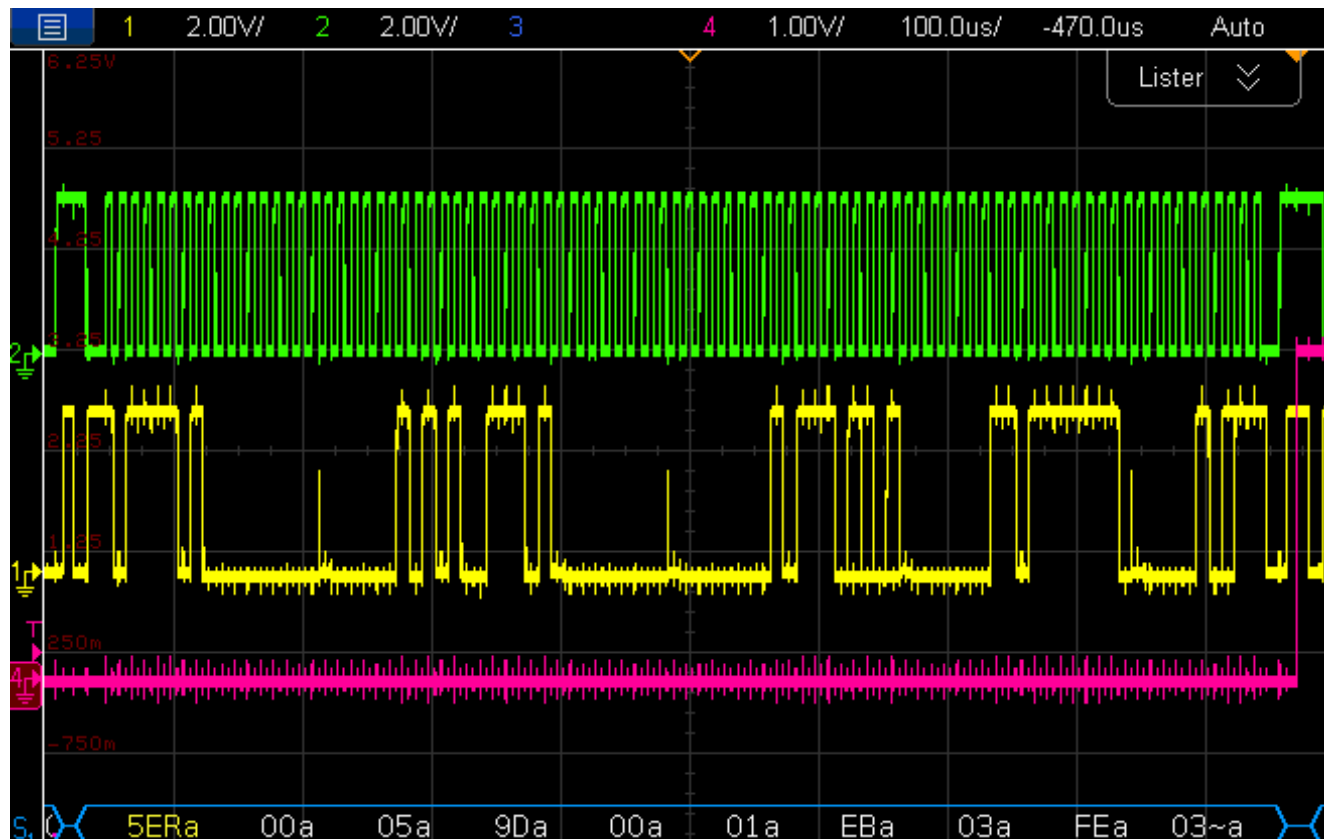


PROGRAMMA DELL'ALGORITMO DI REGOLAZIONE

```
70 // Funzioni
71
72 // Acquisizione campioni PPG
73 uint16_t PPG() {
74     ppgunsigned16 = (uint16_t)(max86150Sensor.getFIFORed() >> 2);
75     Serial.println(ppgunsigned16);
76     return ppgunsigned16;
77 }
78
79 // Restituisce il valore massimo in buffer_memoria[]
80 uint16_t findMax(uint16_t buffer_memoria[]) {
81     uint16_t maxVal = buffer_memoria[0];
82     for (int j = 1; j < M; j++) {
83         if (buffer_memoria[j] > maxVal) maxVal = buffer_memoria[j];
84     }
85     return maxVal;
86 }
87
88 // Restituisce il valore minimo in buffer_memoria[]
89 uint16_t findMin(uint16_t buffer_memoria[]) {
90     uint16_t minVal = buffer_memoria[0];
91     for (int j = 1; j < M; j++) {
92         if (buffer_memoria[j] < minVal) minVal = buffer_memoria[j];
93     }
94     return minVal;
95 }
```

VERIFICA DEI TEMPI I²C

Per verificare l'effettivo funzionamento del programma, è necessario assicurarsi che l'algoritmo di modifica della corrente non causi perdite di dati.



La verifica è stata effettuata collegando un oscilloscopio a:

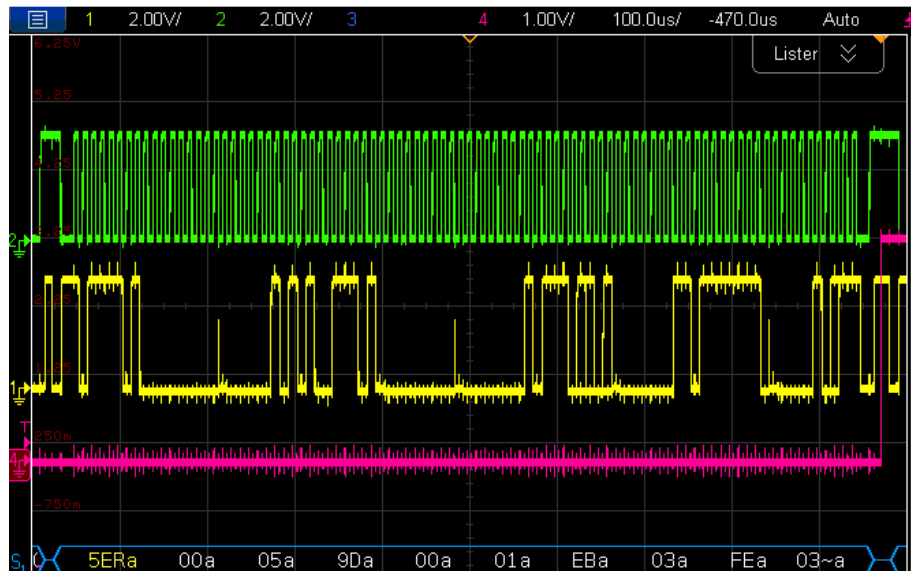
- i **due bus I²C**, in modo da poter visionare la comunicazione tra master e slave in tempo reale;
- un'**uscita digitale del microcontrollore** all'oscilloscopio (l'uscita viene attivata soltanto durante il calcolo della componente AC la regolazione della corrente).

VERIFICA DEI TEMPI I²C

L'indirizzo I2C è 0x5E, seguito dal bit di Write e dal bit di ACK. Dall'oscilloscopio è possibile distinguere:

- il registro **0x04** di Write;
- il registro **0x06** di Read.

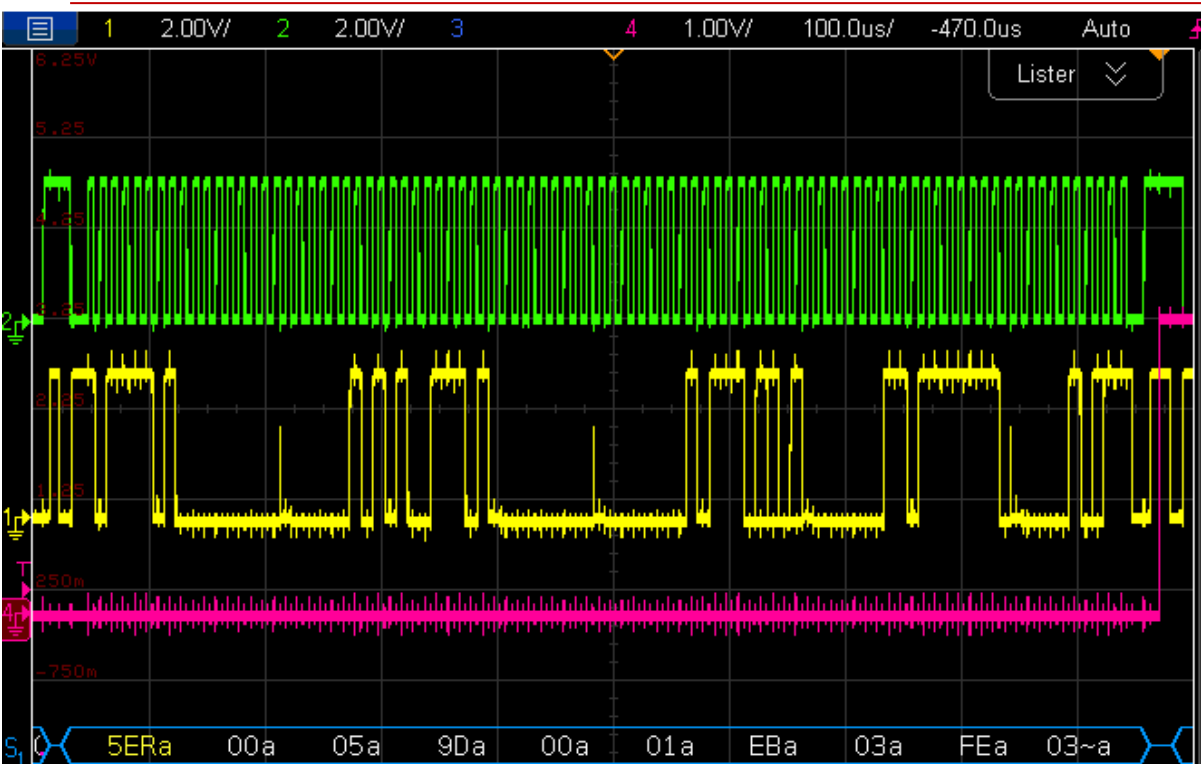
Entrambi i puntatori vengono incrementati automaticamente ad ogni operazione di lettura dei dati dalla memoria FIFO.



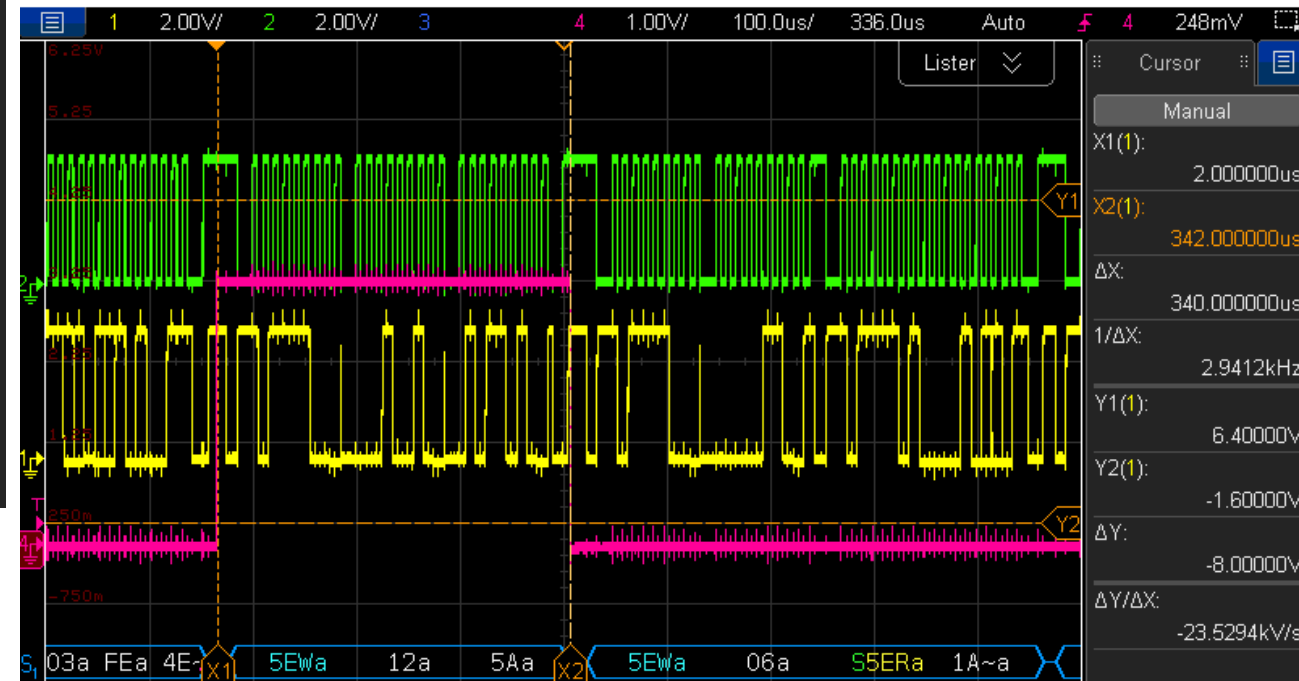
Nel caso in cui le letture dei due registri non coincidano, viene poi letto il registro **0x07**, cioè il registro dei dati FIFO.

Quando la condizione per la modifica della corrente nel sensore viene soddisfatta, il microcontrollore accede al buffer **0x12**, ove scrive il valore binario di corrente da impostare.

VERIFICA DEI TEMPI I²C



Lettura della memoria FIFO

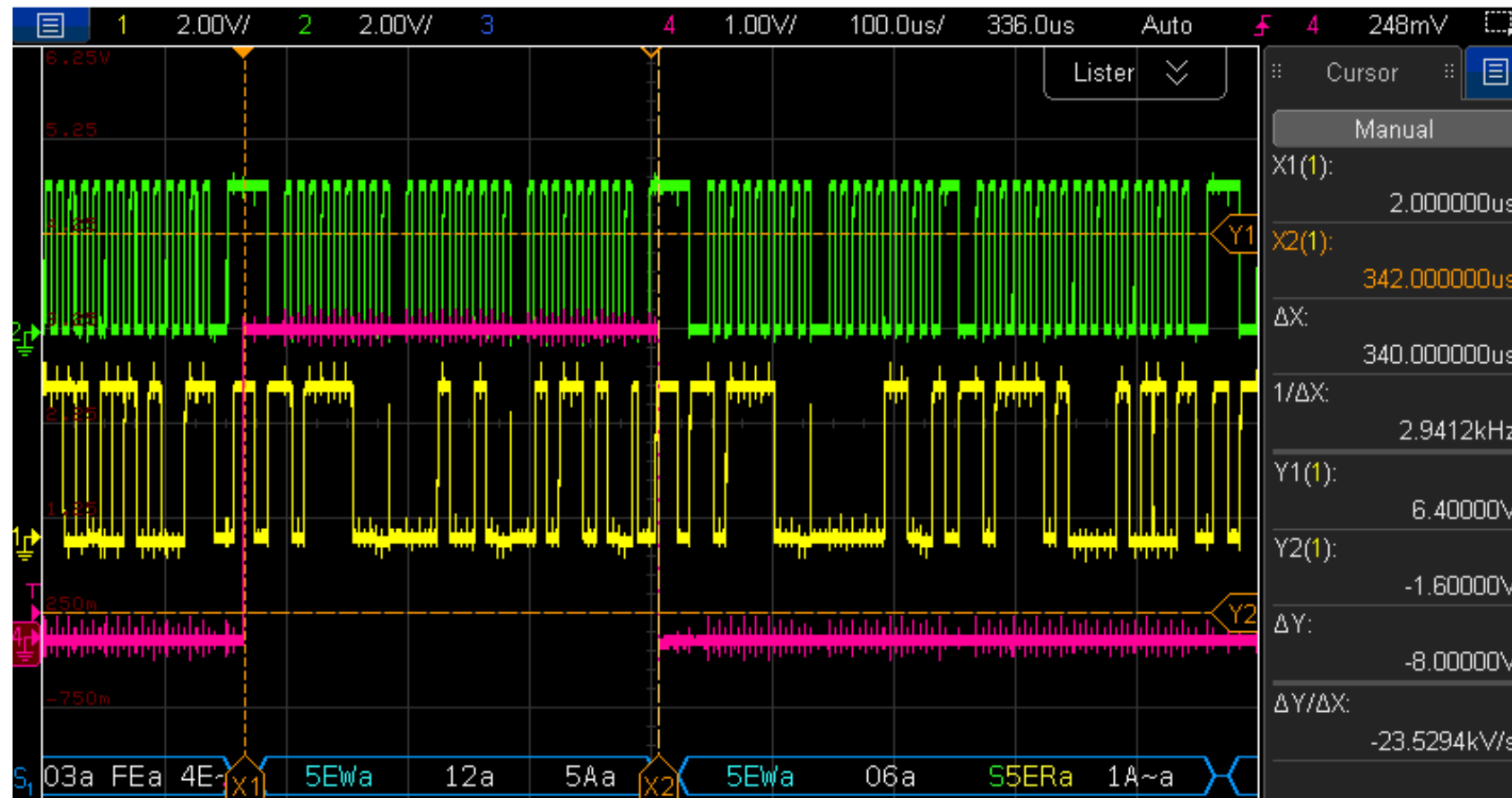


Regolazione
corrente

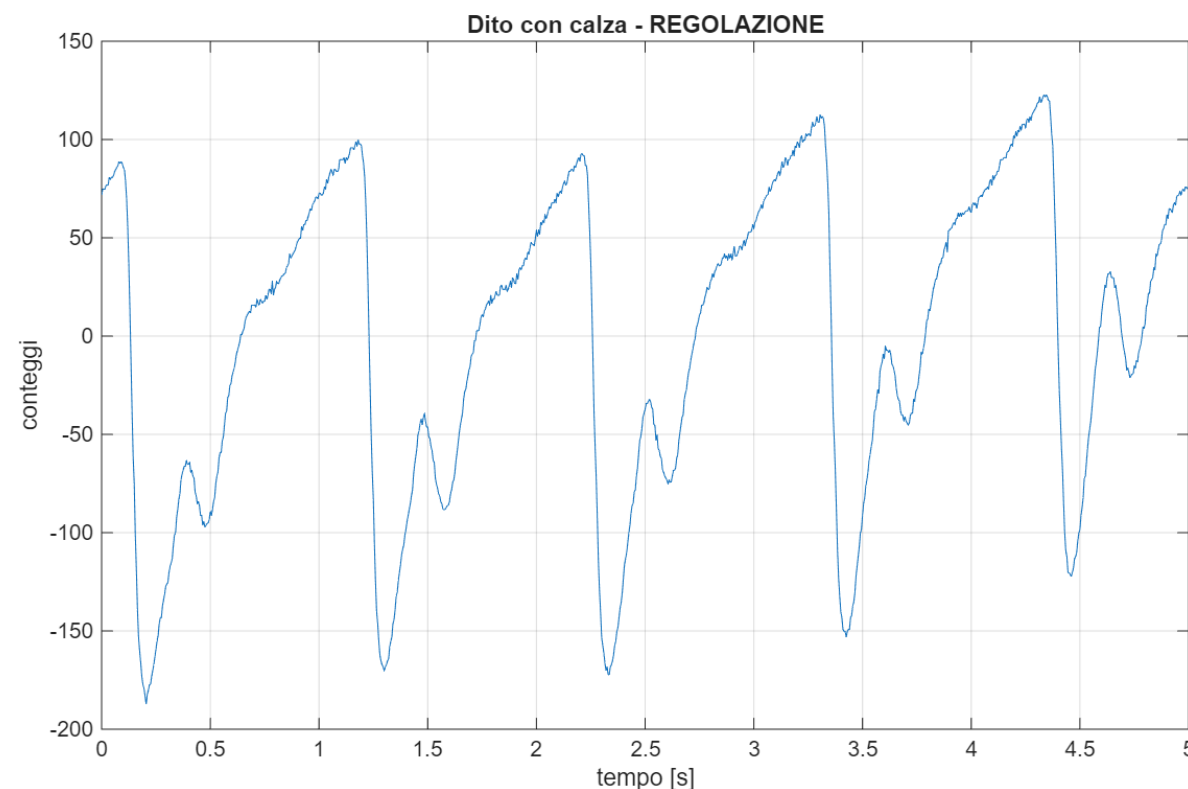
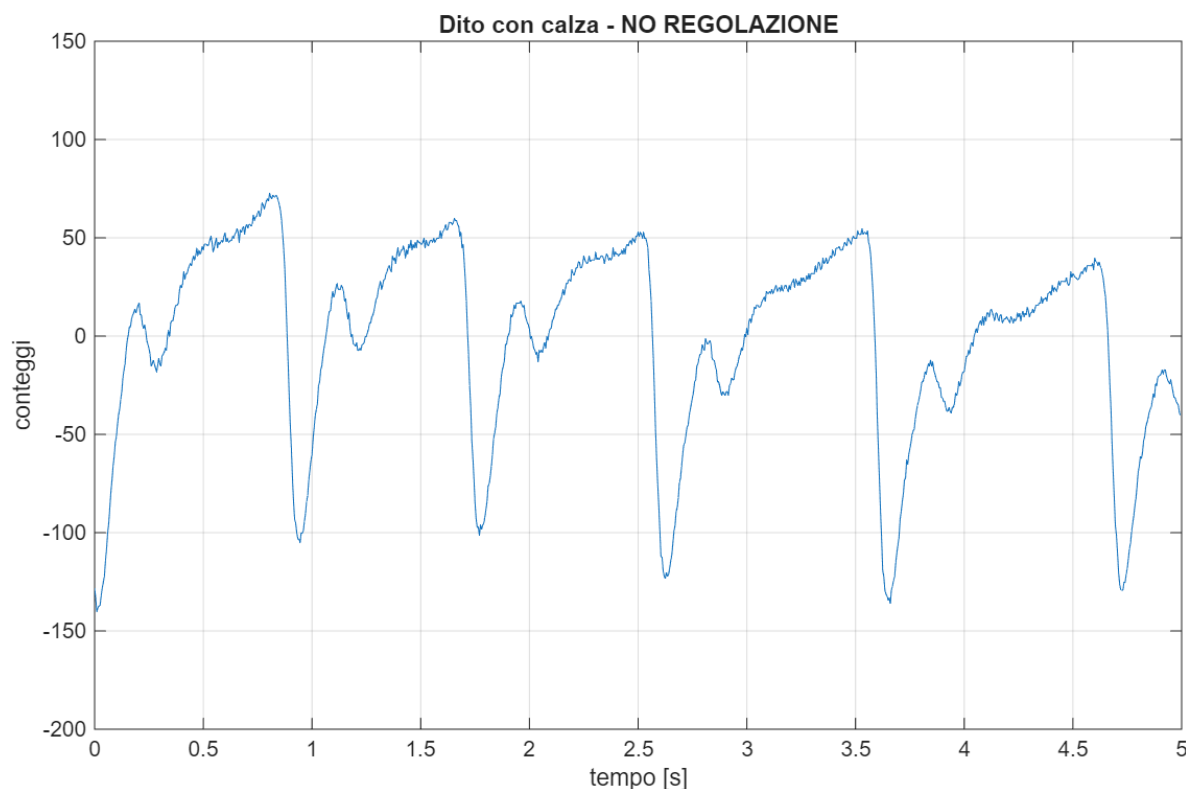
Controllo stato FIFO

VERIFICA DEI TEMPI I²C

La fase di regolazione della corrente dura $342\ \mu\text{s}$: l'intervallo di tempo è sufficientemente breve da non interferire con la memoria FIFO.



Confronto tra la componente AC **senza regolazione** ($I=50$ mA) e **con regolazione** ($I=100$ mA):



Rispetto al primo grafico, nel segnale ottenuto con i 100 mA l'ampiezza picco-picco **raddoppia**.
Posso quindi ipotizzare che esiste una **proporzionalità diretta** tra corrente di alimentazione del diodo e intensità luminosa misurata dal PPG.

CONCLUSIONI

Un **semplice programma** di regolazione è sufficiente ad aumentare in modo consistente l'ampiezza del segnale utile senza dover alterare in altro modo né il sensore, né la posizione del dito.

I limiti del programma sono due:

- il miglioramento del rapporto segnale-rumore è **vincolato al valore massimo di corrente** erogabile al LED;
- la **soglia di innalzamento** della corrente è stata determinata in modo **empirico**.

CONCLUSIONI

Nel momento in cui gli smartwatch saranno sufficientemente affidabili da poter essere utilizzati come dispositivi medici, giocheranno un ruolo fondamentale nella telemedicina.



Sarà dunque fondamentale perfezionare il funzionamento dei sensori PPG, ad oggi l'unica tecnologia nel suo ambito in grado di garantire versatilità, costi bassi e comodità.

Slide 2: immagine saturimetro <https://it.wikipedia.org/wiki/Pulsiossimetro>

Le immagini nelle **slide 6, 9 e 17** sono generate con AI

- «Progettazione di un sistema per l'acquisizione simultanea di segnali ECG e PPG » G. Giorgi, G. Borin
- «La Fotopletismografia (PPG) come Strumento Non Invasivo per il Monitoraggio Continuo dell'Ipertensione: Metodologie di Acquisizione, Applicazioni cliniche e Prospettive Future» M. Veronese, S. Monego
- «Photoplethysmographic sensors, potential and limitations: Is it time for regulation? A comprehensive review»
Francesco Scardulla, Gloria Cosoli, Susanna Spinsante, Angelica Poli, Grazia Iadarola, Riccardo Pernice, Alessandro Busacca, Salvatore Pasta, Lorenzo Scalise, Leonardo D'Acquisto