Mini-Project - Pulse oximeter

Intelligent Sensor 2023/24

Authors: Paolo Bettin, Carlotta De Palo, Tito Vimercati
Supervisor: Lino Marques

08/01/2024

Contents

1 Introduction

The goal of this project was to build a pulse oximeter able to monitor in a non-invasive way a person's peripheral capillary oxygen saturation (SpO2) level. In particular SpO2 is an estimate of the amount of oxygen in capillary blood, which is described as a percentage of the amount of oxy-hemoglobin to total hemoglobin. To estimate these percentages, two LEDs of different wavelengths have been used, in particular a red LED and an infrared LED. Infrared light is mostly absorbed by oxygenated hemoglobin (HbO2) while infrared light is absorbed by deoxygenated hemoglobin (RHb). The LEDs had to emit light alternately so that a single photodiode could detect the variation in light intensity for both wavelengths after crossing the body's tissues and blood. The photoplethysmographic (PPG) signal obtained from the photodiode was constituted of an AC and DC component. The AC component of the signal showed peaks in proximity of the systolic phase (contraction phase) of the cardiac cycle, when "new" blood arrives, so we have high light absorption, while during the diastolic phase we noticed a decrease in light absorption. The DC component is caused by the light absorption of the other blood and tissue components (e.g., venous and capillary blood, bone, water, etc.), independent of the pulsatile variations. In order to calculate the SpO2, we first compute the absorbance ratio of the arterial blood (R):

$$R = \frac{\frac{AC_{red}}{DC_{red}}}{\frac{AC_{ired}}{DC_{ired}}}$$

Because of the alternation of peaks and minimums explained before, we considered the maximums of the photoplethysmographic signals as the AC components, the minimum as the DC components. So the formula can be rewritten in the following way:

$$R = \frac{\frac{max_{red}}{min_{red}}}{\frac{max_{ired}}{min_{ired}}}$$

Then the estimation of the SpO2 is done through the following formula:

$$SpO_2 = aR^2 + bR + C$$

Where a, b and C are constant coefficients determined during the calibration phase.
The main phases we can distinguish in this final part of the work are the following:

1. Building of the hardware
2. Synchronization of red LED, infrared LED and ADC acquisition
3. Signal smoothing and definition of the PPG shape
4. Calibration for SpO2 computation

2 Building of the hardware

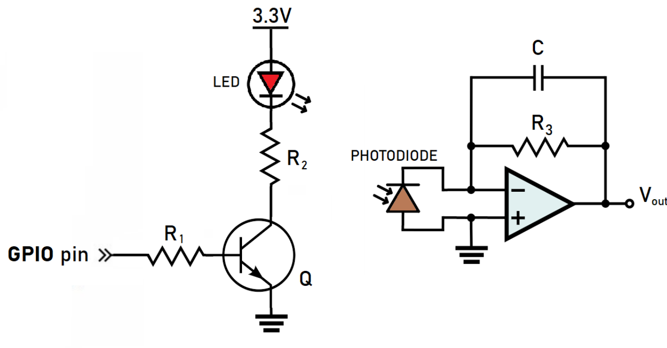


Figure 1: Electric scheme of the apparatus: LED driver scheme on the left, photodiode plus amplification on the right (only one led is shown but the scheme is identical for the other LED).

The hardware components we used are:

1. **STM32L4R5ZI microcontroller**.
2. **Two LEDs**: one in the red wavelengths and one in the infrared wavelengths.
3. **One photodiode**: connected to the ADC component of the STM microcontroller.
4. **Two transistors**: acting as a switch for the LEDs and allowing to power them directly with the 3,3 V power source of the microcontroller.
5. **Resistances**: one is series to the pin of Red LED, one in series to the pin of IR LED, one in sieries to the Red LED, one in series to the IR LED and one for the OPAMP.
6. **Transimpedence Amplifier**: for the amplification of the signal coming from the photodiode. The amplifier has a gain of $-10^6 * I_{IN}$ given by the resistor of 1MΩ.

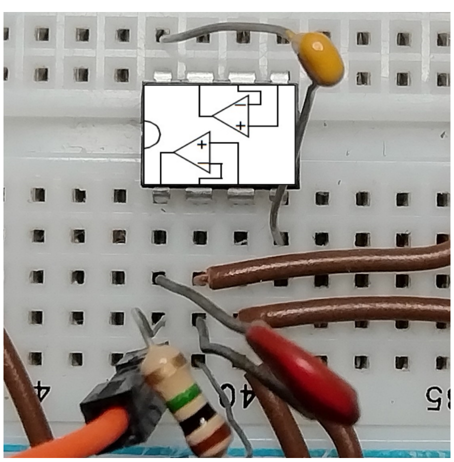


Figure 2: Transimpedance amplifier

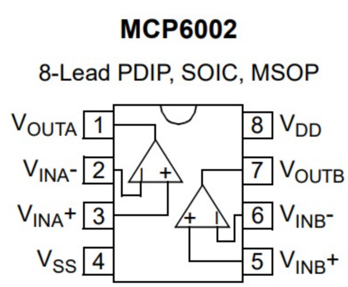


Figure 3: Amplifier schematic

7. **Serial communication**: for visualizing the SpO2 value.
8. **Capacitors:** to mitigate noise between source and ground.

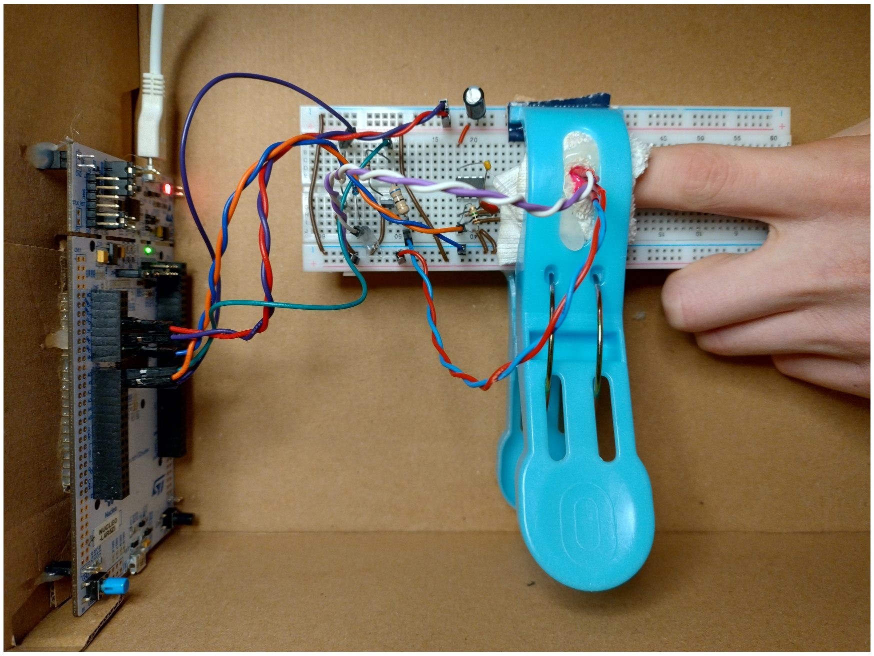


Figure 4: Fully mounted setup picture.

2.1 Noise Mitigation

A big problem we had to face since the beginning was the presence of noise, this because the signals we were working with were very low and we wanted to be sure we had the cleanest signal possible before amplifying it. The following countermeasures have been applied to achieve this goal:

- Shortening of the cables connecting the photodiode to the amplifier to minimize the electromagnetic noise picked up from the environment (since cables act as antennas).

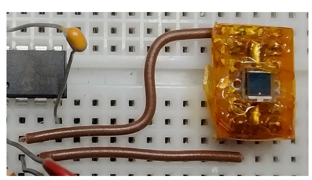


Figure 5: Brown cables are the above mentioned

- The cables connecting the LEDs to the breadboard have been twisted to reduce the inductive coupling and thus the voltage fluctuations in the LEDs signals.

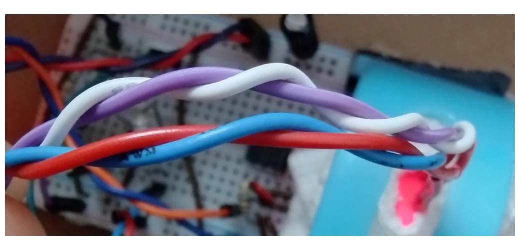


Figure 6: Twisted pairs for both LEDs running to the finger.

- The whole instrumentation have been fitted into a box to minimize the environmental light that could be picked up by the photodiode.
- Some capacities have been added to stabilize the power supply of the amplifier, the output signal of the amplifier and the power supply of the whole circuit.

3 Synchronization of red LED, infrared LED and ADC acquisition

An important issue for the implementation of the pulse oximeter's software was the turning on and off of the two LEDs in an alternative and synchronized way. In addition it was important also to find a way of activating the ADC acquisition of the signal in specific phases of the LEDs' states. The measure of the signal happens:

- during the on-phase of the red LED, corresponding to *state 0*
- during the off-phase of the red LED, corresponding to *state 1*
- during the on-phase of the infrared LED, corresponding to *state 2*
- during the off-phase of the infrared LED, corresponding to *state 3*

Acquiring the signal during both the on and off phases will be important for deleting the environmental light from the signal acquired and allowing signal processing (see "*Signal smoothing and definition of the PPG shape*").
The synchronization have been obtained through the implementation of three PWM signals:

- Yellow: this signal marks the activation, leading to the acquisition, of the ADC.
- Cyan: this signal is the driver for the IR LED
- Violet: this signal is the driver for the RED LED

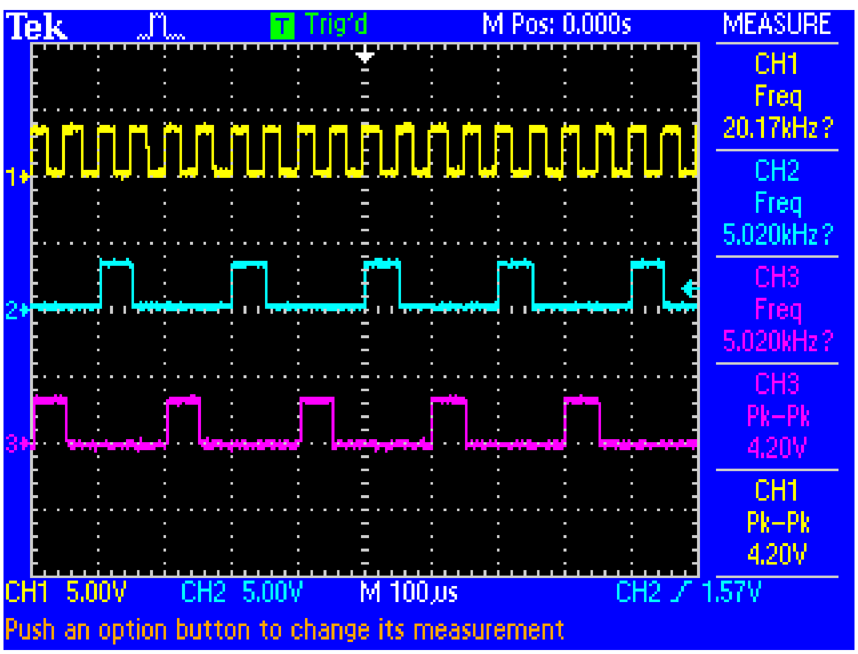
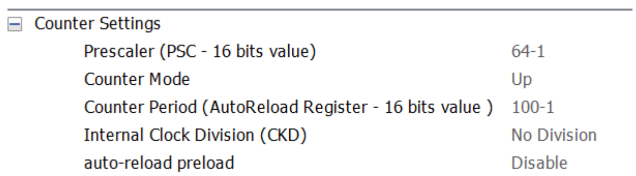


Figure 7: Oscilloscope view of the synchronization signals.

To achieve this synchronization, we utilized two timers (TIMER 3 and TIMER 4) and three callback functions (HAL_ADC_ConvCpltCallback(), HAL_TIM_PeriodElapsedCallback(),HAL_TIM_PWM_PulseFinishedCallback()).

- HAL_TIM_PWM_PulseFinishedCallback() is used every time the yellow signal ends its pulse and it activates the acquisition of the signal by the ADC.
- HAL_ADC_ConvCpltCallback() is called whenever the ADC completes the signal conversion and prepares the data to be read. Inside this callback we accumulate the values of a certain state of our logic as explaned in the next section.
- HAL_TIM_PeriodElapsedCallback() is called when the TIMER 3 that controls the LEDs completes one period of the signal or when the TIMER 4 that controls the ADC acquisition completes its period.

The configuration of the TIMER 3 that controls the LEDs signals is the following:



Counter Settings	
Prescaler (PSC - 16 bits value)	64-1
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits value)	100-1
Internal Clock Division (CKD)	No Division
auto-reload preload	Disable

Figure 8: LED timer 3 settings parameters

Thus, we have a counter that has 100 ticks and each tick is of $1\mu s$ that leads to a period of $100\mu s$. So, every $100\mu s$, inside the callback function, we decide which one of the pin has to be set high/low or if they both have to be turned off, following this logic:

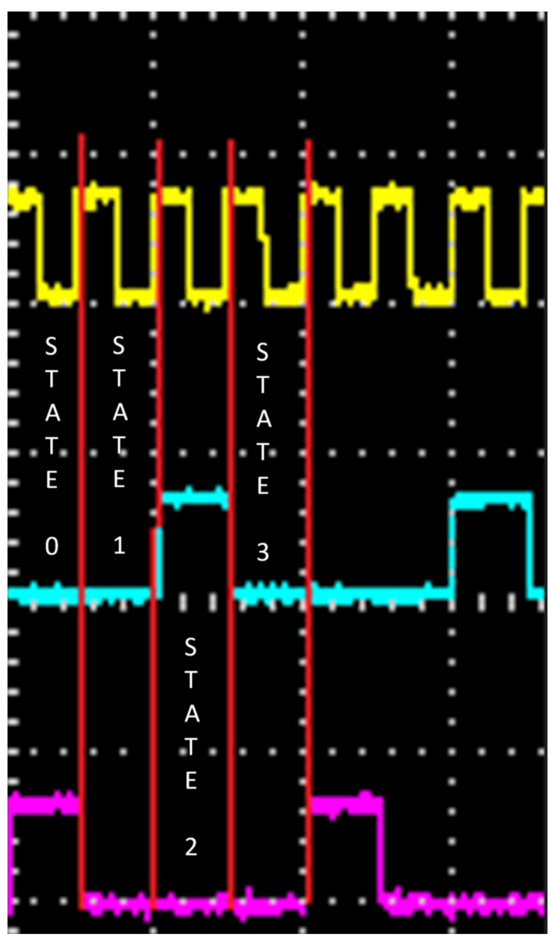


Figure 9: Signals timeline and correlated states.

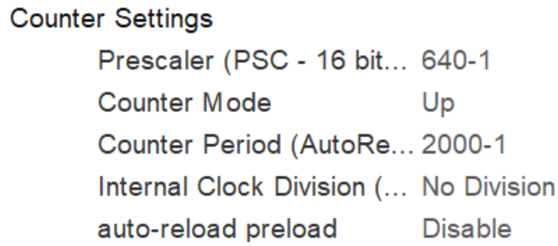


Figure 10: ADC Timer 4 setting parameter

Since the ordinary cardiac frequency is between 1-4 Hz, following the Nyquist sampling theorem, a sampling frequency of 8 Hz is the minimum acceptable to reconstruct the signal. So we chose a sampling frequency of 50 Hz to have a good margin.

4 Signal smoothing and definition of the PPG shape

The phase of signal processing aimed to:

1. Deleting the noise component due to the environmental light during the data acquisition, consequently smoothing the signals.
2. Finding maxima and minima of the signals obtained.
3. Calculating the absorption ratio R.

4.1 Smoothing the signal

For each red and infrared acquisition, we summed together the on-phase's measurement in a variable *sum_value* and we subtracted the off-phase's acquisition value to the same variable. The range of time in which we did this was delimited by a counting variable *sum_count* that was incremented at each OnPhase/OffPhase period. As the counting value desired was reached, the *sum_value* obtained was divided by the *sum_count*. The result was saved in a *mean_value* variable. This operation is equivalent to applying a low pass filter to the original signal, allowing to delete the environmental light component (subtraction of off-phase's measurements) and smoothing the signal coming from the on-phase's measurements. The low pass filter effect is regulated by the *sum_count* to reach.
We also saved the *mean_value* calculated in another array called *mean_buffer* having a maximum size of *BUFFERSIZE*. We waited the array to be completely compiled, then we summed all the values in the *mean_buffer* array and divided the result by *BUFFERSIZE*. Doing so we obtained the mean of all the measurements saved. The final filtered signal consisted in the values contained in the array *mean_buffer* decreased by the mean just calculated. This operation allowed the signal to be always centered on the horizontal axis. At the end of this process, we obtained two filtered and separate signals: one for the red LED and one for the infrared LED.

4.2 Finding maxima and minima

Again, we did the same operations for red and infrared signals in parallel. The main challenge here was finding an approach that allowed us to search for global maxima and minima, and not for local ones. This problem arises because the pulsation of the heart has at least two different peaks, one for the systolic phase and one for the diastolic phase plus the peaks due to the noise. In order to calculate the R parameter we need to distinguish the global maxima and minima. We considered two possible solutions.
1) 3-components arrays The idea behind this approach was to save the values obtained during the previous approach in a three-components array, which could be scanned in search of local maxima and minima of the signal. At each cycle, we first translate the array so that the second component goes in the first position

of the array, the third one in the second, and the third is updated with the latest signal value. These are examples of lines of code:

$$array[0] = array[1]$$

$$array[1] = array[2]$$

$$array[2] = newvalue$$

Then, we verify if the central value (index=1) of the array is greater than both the value in position 0 and in position 2. If it is, that value is saved into another array containing the maxima found. The same operations are done for the research of minima, but the condition to check is if the central value is lower than both the other two.
For ensuring that the maximum and minimum used for the SpO2 calculation are global and not local, we find the greatest value contained in the maxima array, and the lowest value in the minima array. Those are the maximum and minimum value that will be used for the SpO2 computation.

2) 60-cells windows This approach finds directly the global maximum and minimum of the signal inside a window of given time range. We select a 60 cells array in which we save the obtained samples of the signal. This array will thus contain at least full period of the heartbeat signal and we will be sure that the found maximum is not a local maximum. Indeed a window of 60 samples allows us to include all the peaks. The value 60 has been chosen because, in the worst case scenario, we may have all the peaks within 1 second (60 bpm, equivalent to 1 Hz). Since we generate a sample every 20 milliseconds, we need a minimum of 50 cells to cover the entire signal. To provide a margin of safety and ensure that we capture all the peaks, we selected 60. After saving 60 samples of the signal, we start computing the maximum and minimum of the signal in the iterative way, by scanning all the array.

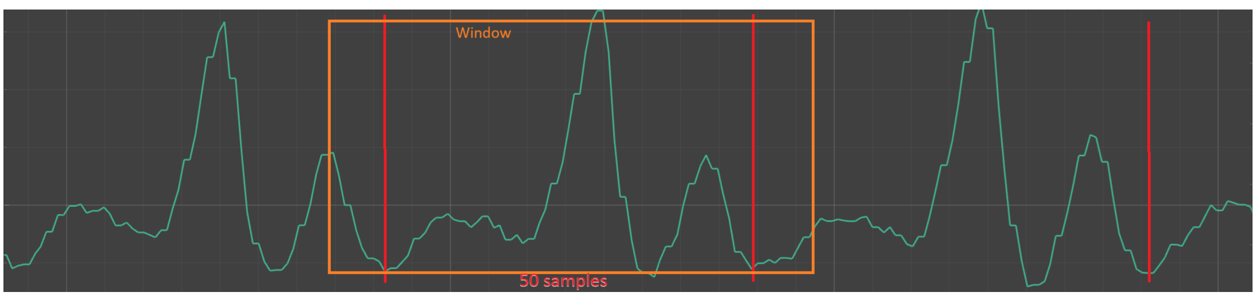


Figure 11: Signal acquisition window.

We decided to use and test the second approach, since it seemed effective and of easy computation. Nevertheless we realize that saving all the samples every time isn't the most efficient solution, mostly in terms of storage. However it was our intention to keep the program as easy as possible, so that we had time to explore satisfactorily all the phases of this work.

4.3 Absorption ratio

The global minimum and maximum were then used for calculating the absorption ratio R:

$$R = \frac{\frac{max_{red}}{min_{red}}}{\frac{max_{ired}}{min_{ired}}}$$

5 Calibration for SpO2 computation

The calculation of SpO2 was then computed through the formula:

$$SpO_2 = bR + C$$

We didn't consider the quadratic component since we weren't able to reproduce a proper calibration process. As we explained in the previous report, the calibration of pulse oximeters is usually done regulating the SpO2 in a controlled manner through the use of a gas mask. We tried to decrease the oxigenation through spending some time in apnea or doing physical exercises. None of these methods worked to decrease significantly the oxigenation, even with the reference device. So we just acquired the SpO2 in different people at normal condition. The parameters b and C were chosen by building the calibration curve using the measurements with our device of the input R and the measurements with the reference device of the desired output SpO2. The parameters obtained were:

$$b = -0.2047$$

$$C = 1.1977$$

6 Testing

We measured the oxygen saturation levels of individuals with both our device and the reference device in various conditions.

Figure 12: Testing phase setting. The box is opened just for illustration.

We used also an external software ("Better serial plotter") for the visualization of the acquired and filtered signal. The first proof of the good working of our system was the observation of the smoothness of the signal, which appears very similar to a classic PPG signal.

Figure 13: The first signal comes from IR light detection and the second from red light.

Initially, during a resting state, all participants exhibited a saturation level of 99%. Secondly, despite efforts to reduce saturation, such as through physical exercise, we were unable to achieve values lower than 98%. A professional equipment is needed to investigate farther the accuracy of the device for lower SpO2 values.

Tester/Measurements	Reference	Tester 1	Tester 2	Tester 3
Measurement n°1	99	99.2	99.3	98.9
Measurement n°2	98	98.5	98.7	98.5

Table 1: Results table

Another issue that arose during the measurements was related to the environmental temperature. After a period in a cold environment and the consequent decrease in body temperature of the subjects, the sensor experienced a loss in the capability of acquiring the signal. This problem was observed not only in our test equipment but also in the reference oximeter we used for comparison.
We also tested the device in different conditions of external illumination. Thanks to the deleting of noise through software and to the addition of a box, the different conditions of external light didn't represent a problem in the acquisition's accuracy.

7 Conclusions

We went through all the phases for the building of a sensor device: from choosing the components and building the hardware, to filtering and processing the signal acquired, to calibrating the device and testing the results obtained. We learned how to use the software STM32cubeIDE and functionalities like the Interrupt Service Routines (ISRs). Although we didn't have the means to fully test our device, we can conclude that we managed to achieve the objective of this project, since the shape of the signals acquired (visible through the external software) and the numerical results obtained were significantly close to those of a classic pulse oximeter. In the end, the project was successfully carried out.

8 Future development

The work could be farther develop for example by testing the other cited approach for finding the global maxima and minima, and by finding smaller support materials to make our device more easly wearable.