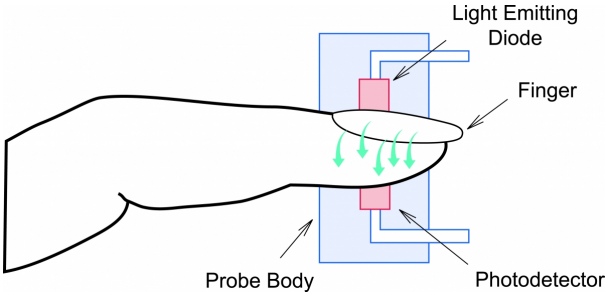
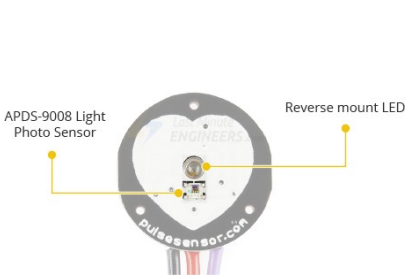
Pulse Oximeter ❤

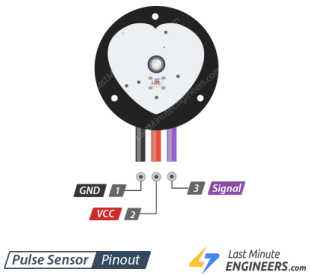
Introduction

This work involves the design of a device that measures the heart rate. The purpose of using a pulse meter is to check how well the heart pumps blood through the body. This device is used to monitor the health of people and their ability to tolerate physical activity. For implementation, an OLED screen, Arduino UNO board, pulse sensor, a Bluetooth data transmission module and a buzzer will be used to emit the sound of the heartbeat.



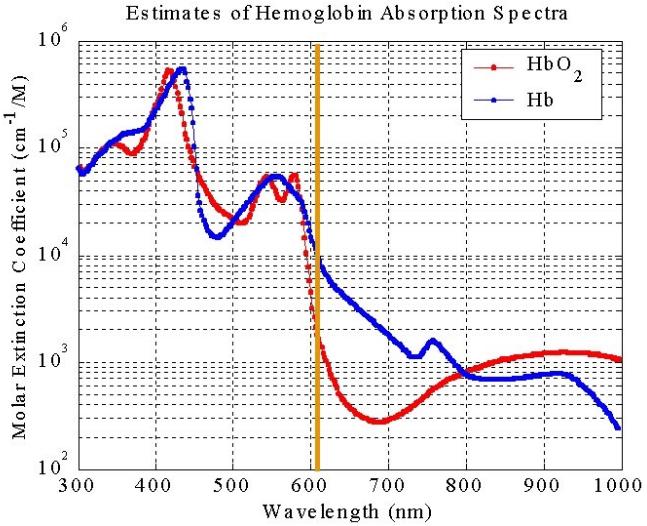
The heart rate sensor is an Arduino-compatible device that can be easily connected and used. It consists of an integrated optical amplification circuit and a noise elimination circuit to ensure accurate pulse measurement.



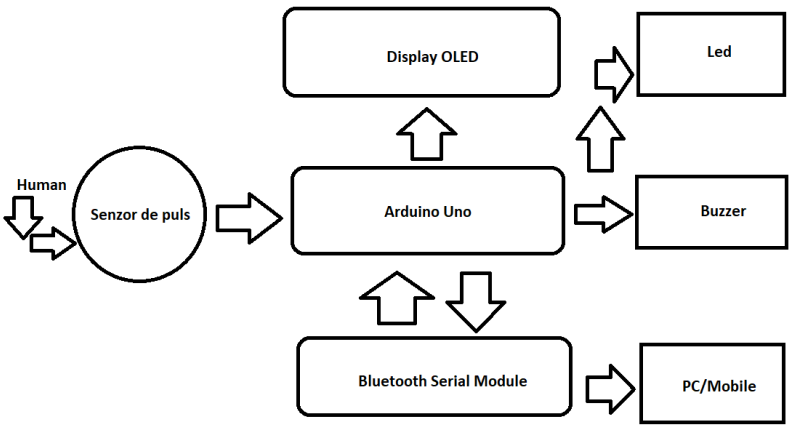


Descriere generală

A common pulse oximeter consists of an electronic processor and a pair of small light-emitting diodes (LEDs) that are directed towards a photodiode through a translucent part of the patient's body, such as the fingertip or earlobe. One LED emits red light at a wavelength of 660 nm, while the other emits infrared light at a wavelength of 940 nm. Oxygenated hemoglobin absorbs more infrared light and allows more red light to pass through, while deoxygenated hemoglobin allows more infrared light to pass through and absorbs more red light.



The processor calculates the ratio between the measurements of red and infrared light, and then utilizes a lookup table based on the Beer-Lambert law to convert it into SpO2 (oxygen saturation).



Hardware Design

The electrical diagram:

Components

- Arduino UNO R3 CH340
- OLED Display 0.91” I2C IIC Blue
- LED 5mm
- Active buzzer 5v
- Pulse sensor XD-58C

- HC-05 Bluetooth serial-module
- Transistor BC547

The pulse signal displayed in the serial monitor:

Software Design

Initially, I defined the necessary pins, sensors, buzzer, and OLED display for the project. These components are essential for its functionality. In the setup and loop functions of the Arduino code, I implemented the core features of the project, which involved reading data from the sensors.

I used the pulse sensor to measure heartbeats accurately. Whenever a heartbeat was detected, I activated both the LED and buzzer, providing a visual and audible indication. Furthermore, I calculated the Beats Per Minute (BPM) value based on the pulse readings and displayed it on the OLED screen. This allowed for real-time monitoring of my heart rate.

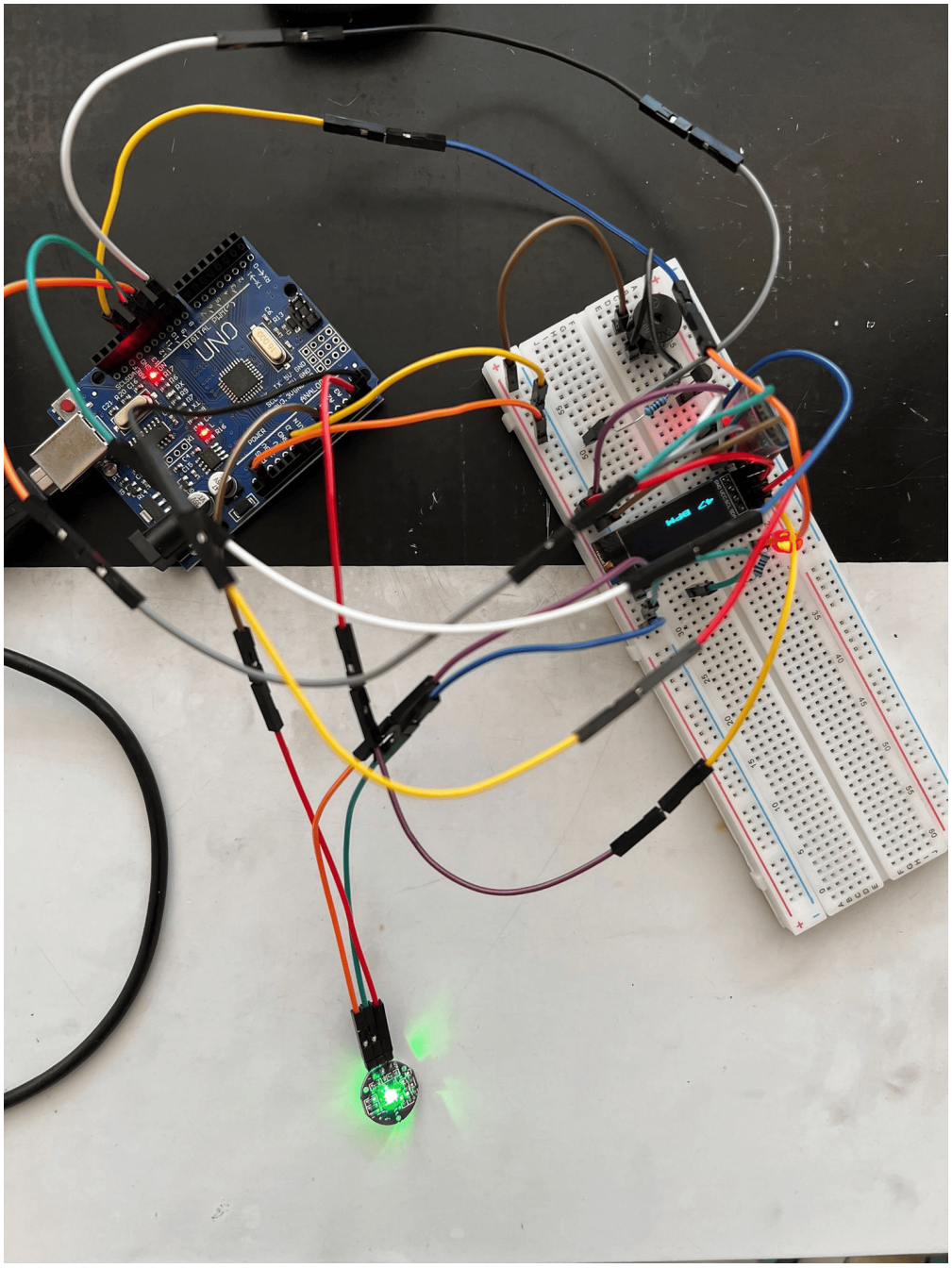
To enhance the project's capabilities, I incorporated a Bluetooth module, specifically the HC-05, which enabled communication between the Arduino and my mobile phone. Using the BTHC05.h library, I transmitted data, including the BPM value, wirelessly from the Arduino to my phone via Bluetooth. This allowed me to monitor my heart rate remotely and potentially log the data for further analysis.

Throughout the development process, I utilized the Arduino IDE as my primary software environment. It provided a convenient platform for me to write and upload code to the Arduino board.

The key libraries I used in this project were as follows:

- BTHC05.h: This library facilitated seamless communication with the HC-05 Bluetooth module, enabling data transmission between the Arduino and my connected mobile device via Bluetooth.
- Wire.h: I used this library for I2C communication, which is a common protocol for interacting with various devices, such as the OLED display used in this project. It enabled straightforward communication between the Arduino and the display module.
- Adafruit_SSD1306.h: This library provided a set of functions that allowed me to easily control the Adafruit SSD1306 OLED display module. It simplified the process of displaying relevant information, including the calculated BPM value, on the OLED screen.
- SPI.h: I employed this library for Serial Peripheral Interface (SPI) communication, which is commonly used for interfacing with devices like SD cards, sensors, and displays. It facilitated the communication between the Arduino and any SPI-compatible components used in the project.

Final Result



Concluzii

Upon completion of this project, I developed a thorough comprehension of how a pulse oximeter operates. I actively utilized Arduino to write the required code and employed the `Serial.println` function to visualize the signal output on the serial monitor. Additionally, I utilized Fritzing to accurately construct the circuit diagram. Furthermore, I gained valuable knowledge on effectively filtering and amplifying biomedical signals using an operational amplifier and a high-pass filter. Moreover, I successfully implemented signal filtering directly within the program code.

Pulse oximeters play a crucial role in medical settings, given their non-invasive nature, affordability, and reasonably high level of accuracy. When measuring heart rate and oxygen saturation, it is advisable to use a pre-calibrated sensor, as each device possesses distinct parameters that can vary based on its specific design and construction.

Download

You can access the code by following this link. marmandiu_darius-valentin_cod_part.zip

Bibliografie/Resurse

1. IoT MQTT-based Heart Rate Monitor using ESP8266 and Arduino: https://how2electronics.com/iot-mqtt-based-heart-rate-monitor-using-esp8266-arduino/
2. Pulse Oximetry - Wiki: https://wikicro.icu/wiki/pulse_oximetry
3. Pulse Sensor with OLED Arduino: https://how2electronics.com/pulse-sensor-with-oled-arduino/
4. Pulse Sensor Arduino Tutorial: https://lastminuteengineers.com/pulse-sensor-arduino-tutorial/
5. How to Display Images on OLED Using Arduino: https://www.instructables.com/How-to-Display-Images-on-OLED-Using-Arduino/
6. Measure Heart Rate and SpO2 with MAX30102: https://create.arduino.cc/projecthub/SurtrTech/measure-heart-rate-and-spo2-with-max30102-c2b4d8?ref=similar&ref_id=372673&offset=0

Export to PDF