Abstract

Oxygen saturation might be one parameter for judging, if severe measures (e.g. artificial breathing) have to be taken against ones covid19 illness. Measuring Oxygen Level, Heartbeat and Temperature is one main function of this prototype. Such devices and system are commercial available but expensive.

Goal of this project is to create a cheap and easy to build device for measuring oxygen level, heartbeat and temperature over a long time. Components used should easily be purchasable on common resources, used software should be mostly free.

This device should help to establish a long time monitoring of sic persons. The area of application are medical institutions, where several covid19 victims are treated and must be monitored.
The complete system can consists of a central unit and mobile sensor devices.

Introduction / Background

Measuring Oxygen levels with a small sensor on an optical basis is a common technology. See https://en.wikipedia.org/wiki/Pulse_oximetry for more details and optical background.
In brief, two LEDs emit light on different wavelengths. Both wavelengths are choosen in a way, that the reflection depends on the amount of stored oxygen in hemoglobin. Light of one LED is well reflected by hemoglobin enriched with oxygen, the other LED light is well reflected by hemoglobin without or with less oxygen. The ratio of both light amplitudes describes the oxygen saturation in blood.

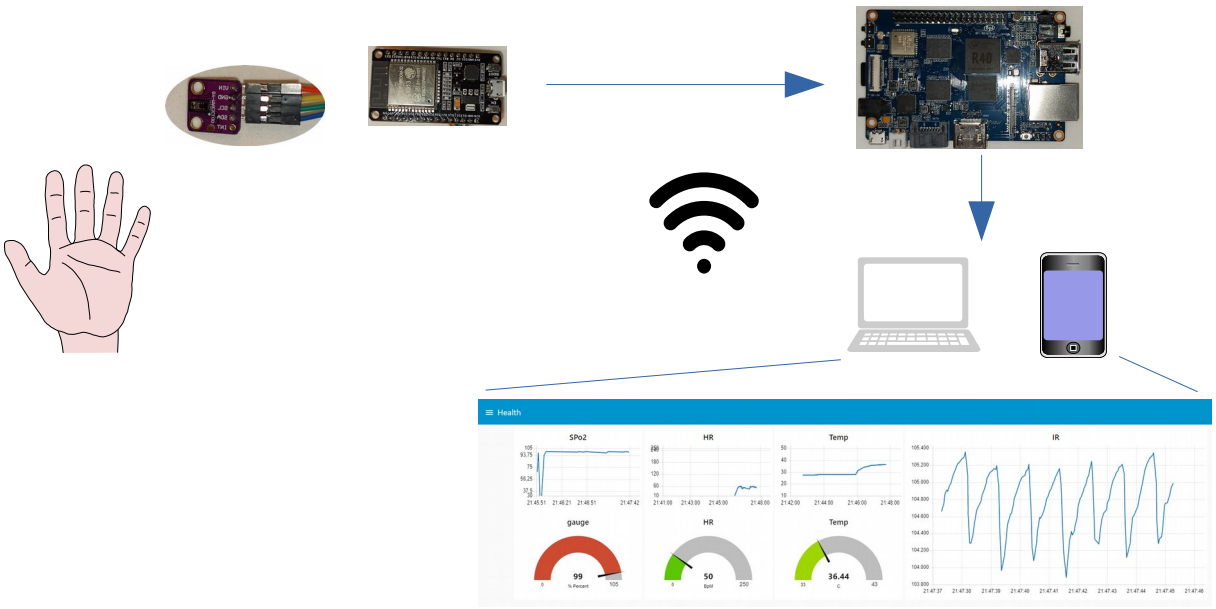
System Architecture

The described prototype uses a small sensor mounted on an arduino shield (in my case a pulseoxy sensor MAX30102). Data collection is done by a small micro controller, which can by powered with a battery (in our case a ESP32 development board).

A small program runs on the micro controller, which collects data samples of the sensor (reads light levels) and calculates the measured heartbeat rate, oxygen level and temperature. The gathered data is send over wifi using the MQTT protocol.

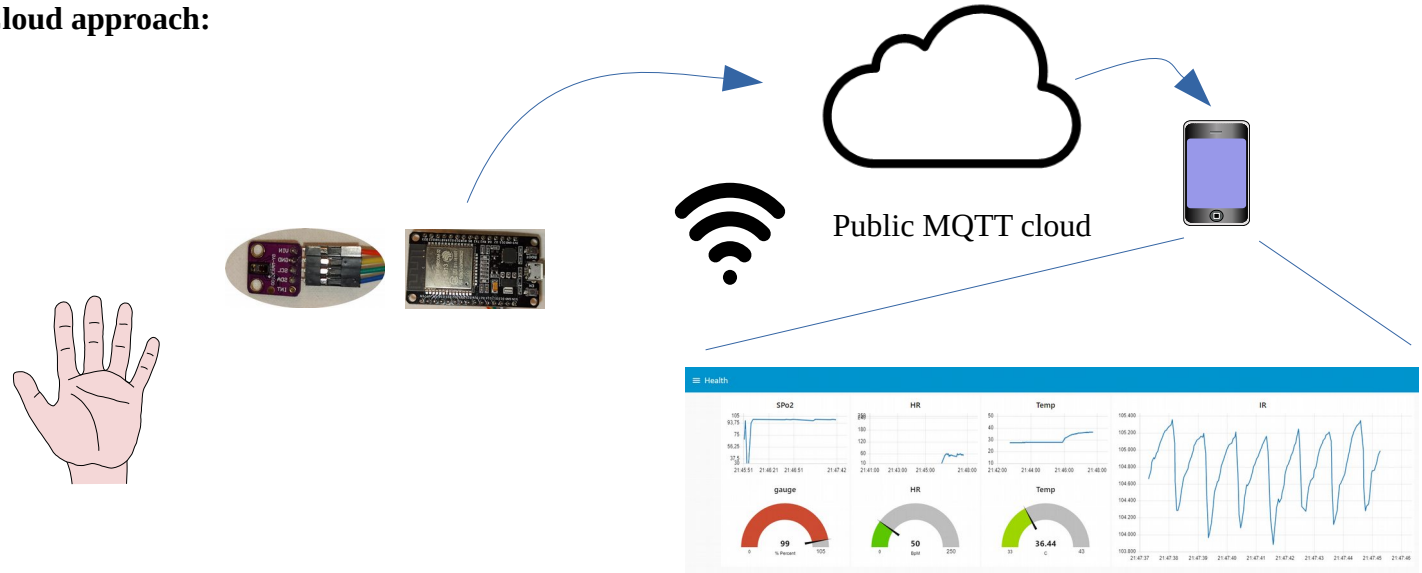
In my case the central unit is single-board computer like a banana pi, or raspberry pi running on a small linux (bananian or raspian). Because we are using MQTT as protocol, we need a broker, which receives the MQTT packages (in our case the mosquito). Visualization is done by node-red and it's dashboard.

If you do not want to use a local MQTT Broker, you can use a public one, and send the data into the cloud. But beware, these are medical data, and it's tricky for data security reasons, to send them into the cloud.



Local approach:

Cloud approach:



Sensor device:

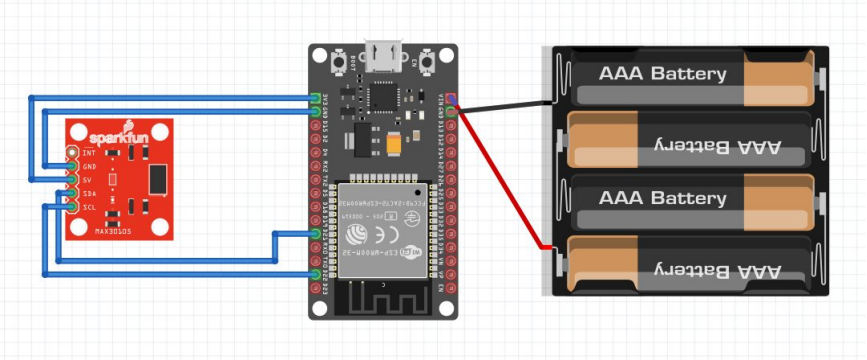


Table: Components / Parts

Sensor: MAX 30102
Datasheet: https://datasheets.maximintegrated.com/en/ds/MAX30102.pdf
Sample Code: https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library

Micro controller: ESP32
https://www.espressif.com/en/products/hardware/esp32/overview
Wikipedia: https://de.wikipedia.org/wiki/ESP32

Development board
Some documentation: https://docs.zerynth.com/latest/official/board.zerynth.doit_esp32/docs/index.html

Software:
Arduino IDE: https://www.arduino.cc/en/main/software
Code: https://github.com/jvoiges/PulseOxy

Central Unit
Banana Pi-M1
Specification: http://www.banana-pi.org/m1.html
Linux Distribution: http://www.lemaker.org/product-bananapi-resource.html
Wikipedia :https://en.wikipedia.org/wiki/Banana_Pi
or Raspberry Pi
https://en.wikipedia.org/wiki/Raspberry_Pi

Software:
MQTT Broker: https://mosquitto.org/
Visualization:
Node Red: https://nodered.org/