Course of
Embedded Systems for E-Health

LM-32
DIGITAL HEALTH AND BIOINFORMATIC ENGINEERING
a.a. 2022/2023

Project Description Document

T.O.H.R.:
Temperature, Oxygen and Heart Rate measuring device

Authors	Capobianco Federica – Fraenza Valeria – Pastore Palumbo Francesco Pio
Team n.	2
Test Date	15 June2023
Report delivery date	12 June 2023
Document version	0.0

ABSTRACT

T.O.H.R is an advanced health monitoring system designed to measure and track vital physiological parameters in the human body: heart rate, body temperature, and blood oxygen levels. Inspired by the continuous rhythm of the human heartbeat, this innovative system provides real-time insights into the body's vital signs.

INDEX

1. SOLUTION

T.O.H.R. is an integrated health tracking system equipped with specific sensors and intelligent algorithms with which can detect anomalies or irregularities in heart rate, temperature, or oxygen levels, promptly alerting.

In particular the heart rate is accurately captured using MAX32664 sensors that detect and analyse subtle changes in blood flow, providing instant updates on heart rate variations. Simultaneously, it precisely estimates the oxygen saturation level in the blood, allowing for comprehensive oxygen monitoring. In addition there is an high-precision temperature sensor, KY-028, that measures the body's temperature, offering insights into any fluctuations or abnormalities.

The application empowers individuals to take control of their well-being, enabling early detection of potential health issues and facilitating timely interventions. By providing continuous and accurate monitoring of heart rate, body temperature, and oxygen levels, it offers a comprehensive understanding of vital signs, promoting overall well-being and peace of mind.

2. FUTURE PERSPECTIVE

Our solution presents as a valid measurement and analysis tool that is possible to make portable or wearable since it can easily be battery powered and made of easily discretizable components so that we can contain the expenses. Our solution is in fact non-prone to electrical induction issues and allows us to project a single sided pcb with limited dimensions which is really cheap to produce and easy to assemble making it also competitive for the market.

3. DESIGN CHOICES

CLOCK CONFIGURATION

The clock system provides the timing reference for the entire microcontroller, including its core, peripherals, and external interfaces. We configured the clock starting by selecting the internal clock source (HSI), then setting the frequency for the system to 16MHz. This will determine the ooverall speed of the microcontroller. After that we configure the AHB prescaler to 4 and the next others to 1, in order to have the same frequency set to 16MHz for all the different peripherals.

TIMERS

We have four timers handled by our routines:

- TIM10: A general purpose timer that just counts up to 15 seconds. It is used to gather about 150 reads in a buffer from the max sensor that have to be further elaborated. It is set to 15 seconds for ease of use, however, for a correct estimation it has to be about 30-60 seconds to perform further analysis on hr variability and be reliable.
  In order to work with one pulse in 15 seconds we used the following formula to set an acceptable value of the period:

$$\text{UpdateEvent} = \frac{\text{Timer}_{\text{clock}}}{(\text{Prescaler} + 1)(\text{Period} + 1)}$$

  We set the prescaler to 15999 and the counter period to 14985, came out multiplying the clock period, 999, for 15; in this way we are working with an UpdateEvent of 15 seconds. In a real scenario, the patient might have to wait slightly more than 15 seconds due to internal clock delays.
  Also it is enabled in interrupt mode.
- TIM2: It scans the continuous monitoring mode by activating the ADC conversion that synchronizes all the other reads.
  In this timer for a correct behaviour we set the prescaler to 41999 and the counter period to 1999.
  Also it works with the DMA.
- TIM3: It applies a pwm signal on a led. To work correctly the enabled channel is set to PWMGeneration CH1. In this case we set the prescaler to 319, the period to 999 and the pulse value to 499.
- TIM11: Another general purpose timer that changes the pulse of the tim3 almost ten times a second to make the led have the breathing effect. The timer is configured with a prescaler and aperid equal to 159 and 999.
  Also, it is enabled in interrupt mode.

TEMPERATURE THRESHOLDS

Temperature thresholds have to take into account that our measuring spot is the fingertip which is marginally cooler than our armpit. To understand how to set our thresholds we performed multiple measurements with both the fingertip temp measurement system and a common armpit temperature to find the delta. It appears that the temperature on the finger is about 6-10 degrees below the armpit one so the delta is between 12% and 14%. Based on this we decided that the target temperature was

too high when reaching 26 degrees also based on the resources. We also put a lower bound to the temperature that is the ambient one, in this period it is about 21 degrees.

OXIGEN SATURATION THRESHOLDS

We consider the oxygen saturation healthy state when its measure it falls between 95% and100%. There is no universal reference to tell whether is too low but with a saturation below 95% it could be a good idea to see the doctor.

HEART RATE THRESHOLDS

We consider the heart rate healthy state when its measure is it falls between 60 bpm and 100 bpm. These values are a mean considering that they vary based on sex, age height and weight too.

4. HARDWARE ARCHITECTURE

CHIP ARCHITECTURE

The beating heart of our system is the stm32 Nucleo board based on the arm chip Cortex-M. The M in Cortex-M stands for "EMbedded" so the board is particularly suited for embedded systems applications.
Our hardware architecture turns around the I2C peripheral bus to which we connected all of ours

I2C PERIPHERALS

I2C based devices:

- **SSD1306 – OLED display 128x64**
- **MAX32664 – Pulse Oximeter**
- **DS1307 – RTC**

The I2C port used is the first one based on PB6(SCL) and PB7(SDA) pins to avoid conflicts with timers and other interfaces used.

HUART PERIPHERAL

To communicate a larger amount of information we used a simplex connection based on UART transmission in which our board is the master node and a virtual COM terminal is the slave.

ACTUATORS

The actuators used in the project are:

- **Three LEDs(green, red, white):** wired with a 220-ohm resistance to pin and gnd. To determine why a LED (Light Emitting Diode) needs a 220-ohm resistance at 3.3 volts, we need to consider Ohm's Law. Ohm's Law states that the current flowing through a conductor between two points is directly proportional to the voltage across the two points and inversely proportional to the resistance.
  The formula for Ohm's Law is:
  $$V = I \cdot R$$
  where:
  - V is the voltage (in Volts)
  - I is the current (in Amperes)
  - R is the resistance (in Ohms)

  In this case, we have a 3.3V power supply and we want to determine the appropriate resistance for the LED. LEDs typically have a forward voltage drop, which is the voltage required for the LED to operate. Let's assume the forward voltage drop of the LED is approximately 2V (this value varies depending on the specific LED).
  The voltage across the resistor can be calculated by subtracting the forward voltage drop from the power supply voltage:
  $$V = 3.3V - 2V = 1.3V$$
  Now, let's assume we want the LED to have a current of 20mA or 0.02A. We can rearrange Ohm's Law to solve for the resistance:

$$R = \frac{V}{I}$$

Substituting the values, we get:

$$R = \frac{1.3V}{0.02A} = 65Ohms$$

Based on the calculation, a 65Ohm resistor would theoretically be required to limit the current to 20 mA. However, common resistor values are often standardized, and 220 ohms is a common value available in many resistor series. So, using a 220Ohmresistor in this case is a suitable choice to limit the current and protect the LED.

- **A piezo-buzzer:** As a design choice to avoid more software elaboration we directly connected the piezo buzzer between the positive and ground of the red led so that in case of an alert it will be driven automatically.
- **SD1306 – OLED display 128x64:** SSD1306 is a single chip CMOS OLED/PLED driver with controller for graphics display system. It consists of 128 segments and 64 municipalities. Data/Commands are sent from general MCU through the hardware selectable 6800/8000 series compatible Parallel Interface, I2C interface or Serial Peripheral Interface. In our software we have chosen to communicate with the display through the I2C i nterface. The I2C communication interface consists of the slave address bit SA0, the I2C bus data signal SDA (SDAOUT/D2 for output and SDAIN/D1 for input), and the I2C bus clock signal SCL (D0).

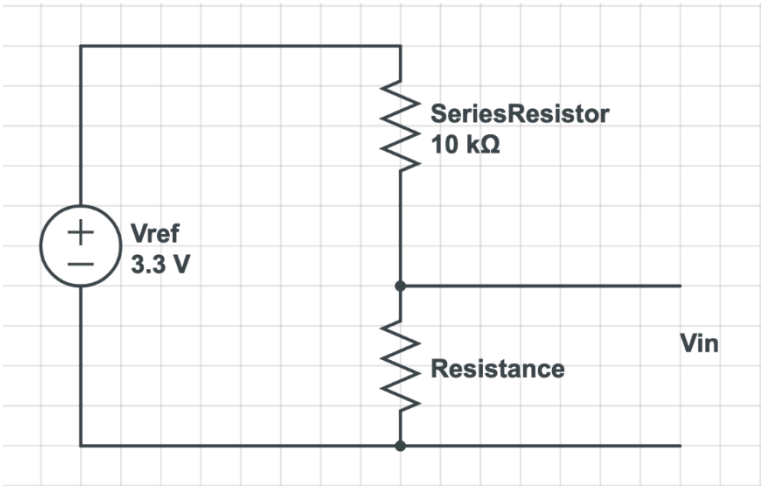
SENSORS

Our sensing part is composed by:

- **Two buttons:** which are simply wired to pull up a pin logical value when pressed and trigger an EXTI routine on the rising edge of the impulse. The buttons are wired in a pull-down configuration with a 10kOhm resistor to avoid using the internal one and have more freedom. Here is how the button works:
  Normally Open (NO) Button: when the button is not pressed, the button contacts are open. In pull-down mode, the pull-down resistor connects the microcontroller's input pin (let's call it PinX) to ground (GND). This establishes a logical low voltage level on PinX.
  Button Pressed: when the button is pressed, it creates a direct connection between the microcontroller's input pin (PinX) and the voltage supply (Vcc) rail. This results in a short circuit across the pull-down resistor.
  Voltage Level on PinX: when the button is pressed, the voltage on PinX becomes close to the supply voltage (Vcc) due to the short circuit. The microcontroller can detect this as a logical high voltage level.
  Regarding the choice of a 10kOhm pull-down resistor, a 10kOhm resistor is a commonly used value that strikes a balance between current flow and power consumption. It limits the current to a small value, typically in the range of a few microamperes, which is negligible for most applications.
  Using a higher resistance value helps reduce power dissipation across the pull-down resistor when the button is pressed, as $P = \frac{V^2}{R}$. This can be advantageous for power-sensitive designs.

In summary, a 10kOhm pull-down resistor in pull-down mode with a button ensures that the microcontroller's input pin has a defined voltage level when the button is not pressed, and it allows the microcontroller to detect when the button is pressed by pulling the voltage level to the supply voltage (Vcc).

- **KY-028:** The sensor has 3 main components on its circuit board. First, the sensor unit at the front of the module which measures the area physically and sends an analog signal to the second unit, the amplifier which amplifies the signal, according to the resistant value of the potentiometer, and sends the signal to the analog output of the module. The third component is a comparator which switches the digital out and the LED if the signal falls under a specific value. We can simplify its circuit schematizing it as a simple voltage divider.
  The most complex design choice has been the estimation of the thermistor module resistance which is a variable resistor varying from 10 and 2M ohm. We did calibrate the sensor turning the potentiometer and trying to match an external thermometer temperature and then hypothized a resistance of about 10kOhm to guarantee we are covering the full range of body temperature values: between 20 and 35 degrees (finger tip).
  The circuit on which the thermistor module breakout is based is the following



Moreover the Steinhart-Hart equation to calculate the temperature is the following:

$$\frac{1}{T} = \frac{1}{T_{25}} + \frac{1}{\beta}\ln\left(\frac{R}{R_{25}}\right)$$

where:

- T_{25} is the nominal temperature set to 25°C
- β is a coefficient set to 3950
- R_{25} is the nominal resistance set to 10kOhm
- R is the Resistance value associated with the breakout to calculate

To find the value of the Resistance we have applied the voltage divider formula to solve the breakout circuit.

$$V_{in} = \frac{V_{ref} \cdot \text{Resistance}}{\text{SeriesResistor} + \text{Resistance}}$$

Based on this formula we performed the following calculations, to solve the circuit and find Resistance.

$$V_{in}(\text{SeriesResistor}) + V_{in}(\text{Resis tan c e}) = V_{ref}(\text{Resis tan c e})$$

$$\text{Resistance}\left(V_{in} - V_{ref}\right) = -V_{in}(\text{SeriesResistor})$$

$$\text{Resistance} = -\frac{V_{in}(\text{SeriesResistor})}{V_{in} - V_{ref}}$$

- **MAX32664 – Pulse Oximeter:** The MAX32664 is a sensor hub family with embedded firmware and algorithms for wearables. It seamlessly enables customer-desired sensor functionality, including communication with Maxim's optical sensor solutions and delivering raw or calculated data to the outside world. This is achieved while keeping overall system power consumption in check. The MAX32664 Version A supports the MAX30101/ MAX30102 high-sensitivity pulse oximeter and heart rate sensor for wearable health for finger-based applications.
  The sensor hub interface provides a fast-mode, I2C slave interface to a microcontroller host. A second I2C interface is dedicated to communicating with sensors. Based on the version this hub can contain various versions of the MAX sensor. Let's analyze the MAX30101 to understand how it works:
  1. Light Absorption: when light is emitted from the LEDs into the skin, it encounters blood vessels. Oxygenated blood absorbs more red light (around 660 nm) while allowing infrared light (around 940 nm) to pass through. Deoxygenated blood, on the other hand, absorbs more infrared light and allows more red light to pass through. By measuring the intensity of the light received after passing through the blood vessels, the MAX30101 can assess the level of oxygenation in the blood.
  2. Light Reflection: some of the emitted light is reflected back to the photodetectors in the MAX30101 after interacting with the blood vessels. The photodetectors convert this reflected light into electrical signals, which can be processed further.
  3. HR Detection: the variations in the received light signals caused by the pulsatile nature of blood flow are analyzed to detect the heart rate. As the heart beats, blood volume changes in the blood vessels, resulting in corresponding changes in the intensity of the reflected light. By measuring the time intervals between these intensity variations, the MAX30101 can determine the heart rate.
  4. SpO2 Estimation: by comparing the differences in the intensity of the red and infrared signals, the MAX30101 can estimate the oxygen saturation levels in the blood (SpO2). This is based on the principle that oxygenated blood absorbs more red light and allows more infrared light to pass through, while deoxygenated blood absorbs more infrared light and allows more red light to pass through.

  The MAX30101 can precisely measure blood oxygen saturation levels and monitor heart rate by fusing the laws of light absorption and reflection with cutting-edge signal processing techniques. The module's algorithms, calibration, and filtering work to reduce interference, noise, and motion artifacts, allowing for accurate and dependable measurements of HR and SpO2

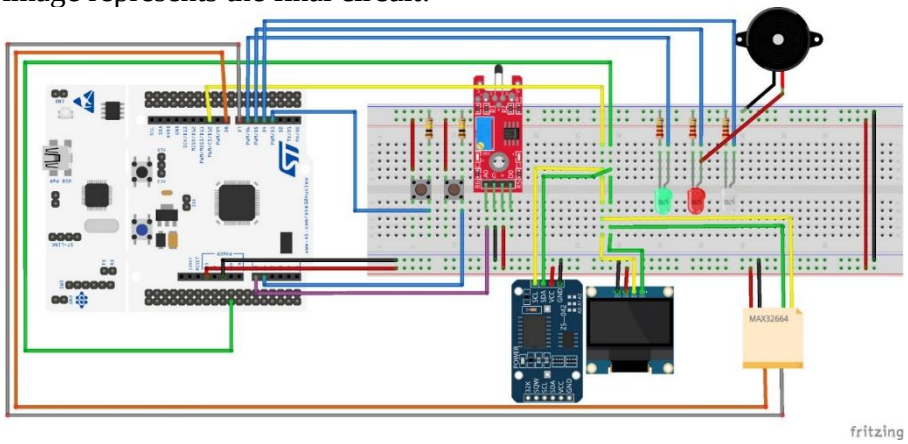
CLOCK

Other than using the internal clock prescaled to 16mHz for power and ease of calculations use we also have an external clock.

- **DS1307 – RTC:** The DS1307 Serial Real Time Clock is a low power, full BCD clock/calendar plus 56 bytes of non volatile SRAM. Address and data are transferred serially via a 2-wire bi-directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The DS1307has a built-in power sense circuit which detects power failures and automatically switches to the battery supply.
  The communication between the controller and the sensor occurs through I2C.

SCHEMATICS

The following image represents the final circuit:



PINOUT:

- **SSD1306 – OLED display 128x64**
  - SCL: PB6
  - SDA: PB7
- **DS1307 – RTC**
  - SCL: PB6
  - SDA: PB7
- **MAX32664 – Pulse Oximeter**
  - SCL: PB6
  - SDA: PB7
  - RST: PA8
  - MFIO: PA9
- **KY-028 - NTC Thermistor**
  - A0: PA0
- **Led Green**
  - Cathode: PB10
- **Led Red**
  - Cathode: PB4
- **Led White**
  - Cathode: PB5
- **Button 1**
  - PB3
- **Button 2**
  - PA1
- **Buzzer**
  - Connected to the same pin of the led Red: PB4

All devices are connected to the same ground pin and have the same 3.3V power supply.

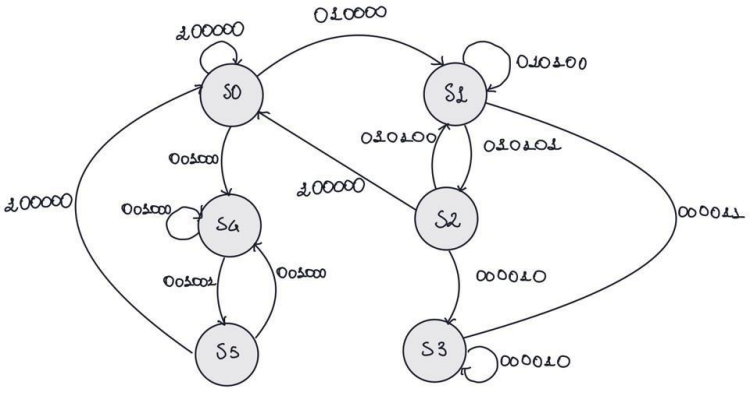
5. SOFTWARE ARCHITECTURE

Our software architecture is based based on the necessity not to create busy waiting. Basically what we did is a polling loop that launches routines based on timers and EXTI interrupts. This choice allows us to have on demand actions triggered by buttons, continuous monitoring and also time intervals driven actions. Moreover we used DMA to perform the ADC conversions.
We have 3 modes:

1. auto mode: in which the controller boots and it is temporized by the tim2 timer that automatically drives the ADC conversion. Using the conversion interrupt to also capure the other sensors values allows us to have synced reads(whitin seconds).
2. Hr/Ox mode: This routine is entered pushing a button linked to an interrupt. This way we deactivate the tim2 interrupt and we only use the max sensor to gather data into a buffer and then analyze them. If the read fails it asks the user to repeat it. If the read results in anomaly values the device launches an alert and then lets the user calm down breathing at the rhythm of a pulsing led. At the end of the breathing the measure starts again to check whether the patient is now fine or not. The time to breathe and to gather data is scanned by a timer that ticks every 15 seconds when active.
3. Temp mode: This routine performs the same actions as the Hr/Ox routine but avoids the breathing part.

6. SOFTWARE PROTOCOLS

To describe the software protocol we used a finite state automaton with six states.



S0	S1	S2	S3	S4	S5
General monitoring loop	Heart Rate and Oxygen percentage monitoring	Heart Rate and Oxygen percentage analysis	Breathig loop	Temperature monitoring	Temperature analysis

Each state represents a different routine that the monitoring system can execute: the routine is accessed by a specific bit configuration where, starting from the left

- the first bit set to 1 is for the continuous acquisition of all the data simultaneously
- the second bit set to 1 is for the acquisition just from the MAX sensor for 15 seconds (period duration)
    - the third bit set to 1 is for the elaboration of the MAX parameters
    - the sixth bit set to 1 is for the period elapse of the timer TIM10
- the fourth bit set to 1 is for the acquisition just from the Temperature sensor for 15 seconds (period duration)
    - the fifth bit set to 1 is for the elaboration of the Temperature parameters

To pass from one state to another some elaborations are performed. They are split between timed and conditioned ones:
The timed loops stay set for 15 seconds;
The conditioned loops are triggered by EXTI routines or failed measurements/thresholds reached.
We considered a measurement buffer acceptable when the number of null values is below 20% for HR(because the hub associates each hr value with its own confidence) and below 30% for blood oxygen saturation.
When a buffer is considered acceptable an average calculation is performed. To evaluate whether the average is representative of the buffer values we calculate the absolute uncertainty with the following formula:

$$\text{Uncertainty}_{\text{abs}} = \frac{\text{Max} - \text{Min}}{2}$$

To obtain the percentage uncertainty, we computed the relative uncertainty by dividing our absolute uncertainty for the average value, and then evaluated in percentage:

$$\text{Uncertainty}_{\text{rel}} = \frac{\text{Uncertainty}_{\text{abs}}}{\text{mean}}$$

$$\text{Uncertainty}_{\text{perc}} = \text{Uncertainty}_{\text{rel}} \cdot 100$$

The more the absolute uncertainty is near to our average the lower will be the percentage uncertainty so the more accurate are the measurements considering the fact that our patient is resting and it shall not have many fluctuations in heart rate and blood oxygen concentration.

7. INFORMATIONS ABOUT POWER DISSIPATION

SSD1306 – OLED display 128x64

- VDD = VCC = 3.3V
- V_{max} = OLED driving output voltage = 15V
- I_{common} = Common maximum sink current = 15mA
- $I_{segment}$ = Segment maximum source current = 100μA

To calculate the power consumption, we'll consider the maximum values.
First, we'll calculate the power consumption for the common sink current:

$$P_{common} = VDD \cdot I_{common} = 3.3V \cdot 0.015A = 0.0495W \text{ or } 49.5mW$$

Next, we'll calculate the power consumption for the segment source current:

$$P_{segment} = V_{max} \cdot I_{segment} = 3.3V \cdot 0.00001A = 0.000033W \text{ or } 33\mu W$$

Now, let's calculate the total power consumption by summing up the power consumed by the common sink and segment source however $P_{segment}$ is so small not to be considered in this order:

$$P_{tot} = P_{common} + P_{segment} = 49.5mW + 0 = 49.5mW$$

DS1307 – RTC

- I_{active} = Active Supply Current = maximum 1.5mA
- $I_{leakageTyp}$ = VBAT Leakage Current = typical 5nA
- $I_{leakageMax}$ = VBAT Leakage Current = maximum 50nA
- V_{RTC} = Operating Voltage = 3.3V

First, let's calculate the power consumption when the RTC is in the active state:

$$P_{active} = V_{RTC} \cdot I_{active} = 3.3V \cdot 0.0015A = 0.00495W \text{ or } 4.95mW$$

Next, we'll calculate the power consumption due to VBAT leakage when the RTC is in a non-active state:

$$P_{leakageTyp} = V_{RTC} \cdot I_{leakageTyp} = 3.3V \cdot 5nA = 0.00001545W \text{ or } 15.45\mu W$$

$$P_{leakageMax} = V_{RTC} \cdot I_{leakageMax} = 3.3V \cdot 50nA = 0.000165W \text{ or } 165\mu W$$

The power consumption of the RTC module will vary depending on its operating state. When the RTC is active, the power consumption is approximately 4.95mW. In a non- active state with VBAT leakage, the power consumption can be as low as 15.45µW (typical) or as high as 165µW (maximum).
We do not take into account the leakage because it is one order smaller so it is considered irrelevant.

MAX32664 – Pulse Oximeter

- I_{led} = LED current = 20mA

To calculate the power consumption for the LED, we'll use the formula:

$$P = V \cdot I$$

The voltage across the LED will depend on the specific characteristics of the LED and the circuitry used to drive it.
Typically, for a red LED, V_{LED} = the forward voltage drop, can range from approximately 1.8V to 2.2V. Let's assume a value of 2V for this calculation.

$$P_{LED} = V_{LED} \cdot I_{LED} = 2V \cdot 0.02A = 0.04W \text{ or } 40mW$$

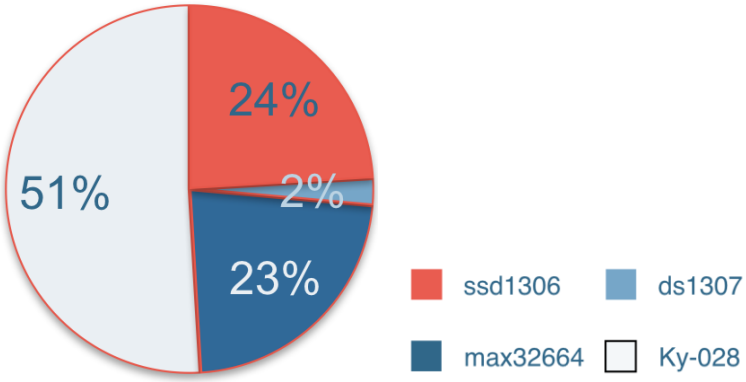
Therefore, the estimated power consumption for the red LED, driven by the MAX32664 microcontroller with a current of 20mA, is 0.04W or 40mW.
To estimate the full power consumption we did measure by hand the current using a multimeter:

$$I_{measured} = 24mA$$
$$P_{inst} = I_{measured} \cdot V_{ref} = 24mA \cdot 3.3V = 79.2mW$$
$$P_{inst} = P_{LED} + P_{inst} = 40mW + 7.9mW = 48mWW$$

KY-028 - NTC Thermistor
The thermistor used has a breakout board that masks the actual resistance, in the hypothesis that R=10kOhm we can calculate the current flowing using Ohm's law:

$$I = \frac{V_{ref}}{R} = \frac{3.3V}{10kOhm} = 33mA$$
$$P_{inst} = I \cdot V_{ref} = 33mA \cdot 3.3V = 108mW$$

Total instantaneous power consumption



Given the fact that all the sensors above are active at 100% usage all the time we can simply sum the power:

$$P_{instTot} = P_{ssd1306} + P_{ds1307} + P_{max32664} + P_{thermistor} =$$
$$= 49.5mW + 4.95mW + 48mW + 108mW = 210.45mW$$

In the hypothesis that the operating time is 1h we have a power consumption of 211.95mW/h.

$$BatteryLife = \frac{BatteryCapacity}{LoadCurrent}\left(\frac{mAh}{mA}\right)$$

- Load = 63.77mA

If the sensors only are provided with an external battery with 850mAh capacity at 3.3v the battery life should be: 13 Hrs 32 Min, however we are not keeping into account the consumption of the board.

8. REFERENCES

TEMPERATURE THRESHOLDS
"Normal body temperature is considered to be 37°C; ... Among normal individuals, mean daily temperature can differ by 0.5°C": https://www.ncbi.nlm.nih.gov/books/NBK331/
"The temperature between the subjects was homogeneously selected to be within a range of 18–26°C": https://www.frontiersin.org/articles/10.3389/fnhum.2019.00066/full#B85

OXIGEN SATURATION THRESHOLDS
"There is no set standard of oxygen saturation where hypoxemia occurs. The generally accepted standard is that a normal resting oxygen saturation of less than 95% is considered abnormal.": https://www.ncbi.nlm.nih.gov/books/NBK525974/

HEART RATE THRESHOLDS
"The normal resting heart rate (when not exercising) for people age 15 and up is 60 to 100 beats per minute (bpm). ": Normal Resting Heart Rate By Age (Chart) – Forbes Health

SSD1306 DATASHEET: https://pdf1.alldatasheet.com/datasheet-pdf/view/1179026/ETC2/SSD1306.html

DS1307 DATASHEET: https://pdf1.alldatasheet.com/datasheet-pdf/view/58481/DALLAS/DS1307.html

MAX32664 DATASHEET: https://cdn.sparkfun.com/assets/4/3/c/2/b/MAX32664.pdf

KY-028 DATASHEET: https://datasheetspdf.com/pdf-file/1402039/Joy-IT/KY-028/1