BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER SYSTEMS
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

HEARTH RATE SENSOR BASED ON ESP32

SENZOR TEPU ZALOŽENÝ NA ESP32

TERM PROJECT
SEMESTRÁLNÍ PROJEKT

AUTHOR **SAMUEL DOBROŇ**
AUTOR PRÁCE

BRNO 2022

Contents

Chapter 1

Introduction

The aim of this project is design and implementation of hearth rate and oxygen saturation monitor. Hearth rate frequency (measured in beats per minute) as well as oxygen saturation needs to be shown on display.
Used components were:

- `MAX30102` sensor of hearth rate and oxygen saturation [3]
- `SSD1306` OLED display [2]
- `ESP32` microcontroller [4]

From software's point of view, the heart is `Espressif IoT Development Framework`[1] `(ESP-IDF)`.

Chapter 2

Circuit diagram, components

To easier manipulation, every components is placed on breadboard but for illustration purposes, components are connected directly.

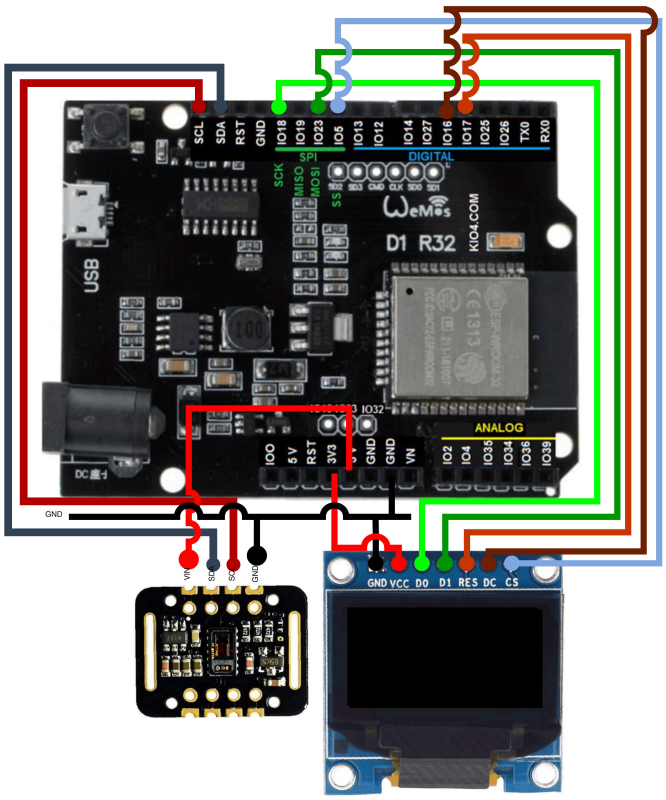

Figure 2.1: Circuit diagram

2.1 `MAX30102` - pulse oximeter

Module uses I^2C[1] protocol for communication with outside components.
Input voltage pin of oximeter is connected to $5V$ pin on `WeMoS` board. For grounding,

[1] More on https://en.wikipedia.org/wiki/I^2C

common ground both for display and oximeter is used. Since this sensor uses I^2C for communication, pins SDA and SCL are connected to the same pins on microcontroller - SDA and SCL.

2.2 SSD1306 OLED display

Display support both I^2C and SPI. But since I^2C is already used for MAX30102 - pulse oximeter SPI needs to be used instead. Display requires $3.3V$ as input voltage, common ground both for oximeter and diplay is used. D0 pin is connected to IO18 pin on microcontroller and D1 is connected to pin IO23. Data/command (DC) pin is connected to pin IO16 on the microcontroller side, as well as chip select CS pin is connected to IO5. Reset pin RST is connected to IO17.

Chapter 3

Implementation

Project was inspired by [5], implementation of I^2C communication with pulse oximeter is taken from there as well as algorhitm of proration to beats per minutes. Some parts of this "library" was refactored. Communication over `SPI` with text rendering is not implemented from scratch, standard but adjusted `ESP-IDF` library is used, available on https://github.com/nopnop2002/esp-idf-ssd1306. Some of unnecessary parts were removed. Rest of the implementation is done in `main/main.c` file.

After machine boot `app_main` function is executed. It initializes both display and pulse oximeter. Afterwards 2 tasks "parallel" are created - `max30102_task` and `draw_data_task`. `draw_data_task` based on global variable `finger_on_sensor` prints "Put your finger on the sensor" message. If the finger is placed on sensor, actual hearth rate and oxygen saturation is printed on the display. Those 2 values are updated by `max30102_task` task and minimal allowed time between updates is 0.5 *seconds*.

3.0.1 Flashing

Program needs to be compiled and flashed to microcontroller. Just use `idf` and `idf.py flash monitor` command that builds everything (if necessary) and flashes it into `ESP32`.

Chapter 4

Video demonstration

Video demonstration of project could be found on
https://www.youtube.com/watch?v=KGk2CRs4Ns8.

Chapter 5

Conclusion

Project implements all the parts of assignment, it even implements displaying of oxygen saturation. Results are not "profesionally" validated, I just compared it with my Apple Watch and results were close to each other.

5.0.1 Proposed evaluation

- Functionality 5 pts - project implements all required functionality, there is even displaying of oxygen saturation
- Quality of code 2 pts - project is not implemented from scratch
- Presentation 1 pts
- Documentation 3 pts - scope is not that big
- Approach 1 pts
- Total 12 pts

Bibliography

[1] *ESP-IDF Programming Guide* [[online]]. [seen on 2022-11-24]. Available at: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/esp-idf-en-v5.1-dev-2186-g454aeb3a48-esp32.pdf.

[2] *128 x 64 Dot Matrix OLED/PLED Segment/Common Driver with Controller* [[online]]. Rev. 24. April 2008. [seen on 2022-11-30]. Available at: https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf.

[3] INTEGRATED maxim. *High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health* [[online]]. Rev. 05. November 2018. [seen on 2022-11-25]. Available at: https://www.analog.com/media/en/technical-documentation/data-sheets/max30102.pdf.

[4] SYSTEMS, E. *ESP32 - Technical Reference Manual* [[online]]. Rev. 15. March 2020. [seen on 2022-11-25]. Available at: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf.

[5] WARREN, M. *ESP Heart Rate Monitor* [[online]]. Rev. 02. May 2020. [seen on 2022-11-20]. Available at: https://hackaday.io/project/164155-esp-heart-rate-monitor#j-discussions-title.