



# Dokumentace

## **ESP32: Měření srdečního tepu [analogový senzor]**

Mikroprocesorové a vestavěné systémy

# Obsah

<b>1 Úvod</b>	<b>1</b>
<b>2 Vývojové prostředí</b>	<b>1</b>
<b>3 Zapojení hardwaru</b>	<b>1</b>
3.1 OLED displej . . . . .	1
3.2 Snímač srdečního tepu PulseSensor . . . . .	1
<b>4 Potíže</b>	<b>2</b>
<b>5 Řešení</b>	<b>2</b>
<b>6 Funkcionalita a testování</b>	<b>3</b>
<b>7 Závěr</b>	<b>3</b>
<b>8 Zdroje</b>	<b>4</b>

# 1 Úvod

Cílem projektu bylo s využitím vývojového kitu Espressif ESP32 [3], snímačem srdečního tepu PulseSensor [2], grafického OLED displeje [1] a vývojového prostředí Arduino zrealizovat systém pro měření srdečního tepu.

Použitý senzor srdečního tepu generuje analogový výstup, který je konvertován na digitální podobu pomocí analogově-digitálního převodníku uvnitř mikrokontroléru ESP32.

## 2 Vývojové prostředí

Projekt byl vyvíjen ve Visual Studio Code ve spojení s PlatformIO, jenž zajišťovalo správu knihoven, nahrávání kódu na hardware a integraci s mikrokontrolérem ESP32.

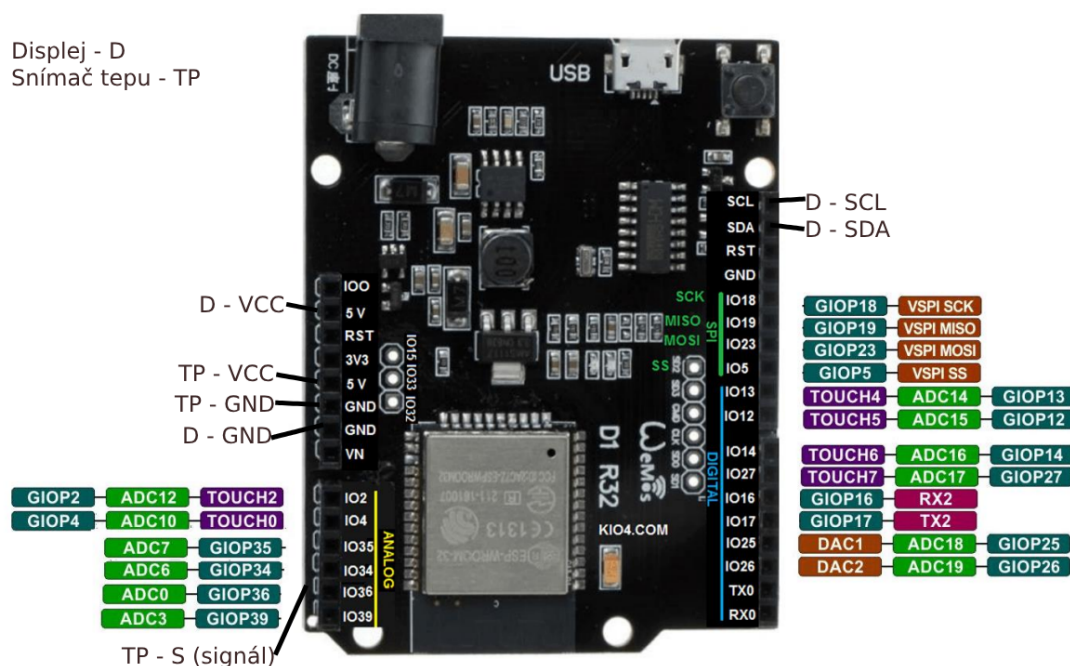
### 3 Zapojení hardwaru

### 3.1 OLED displej

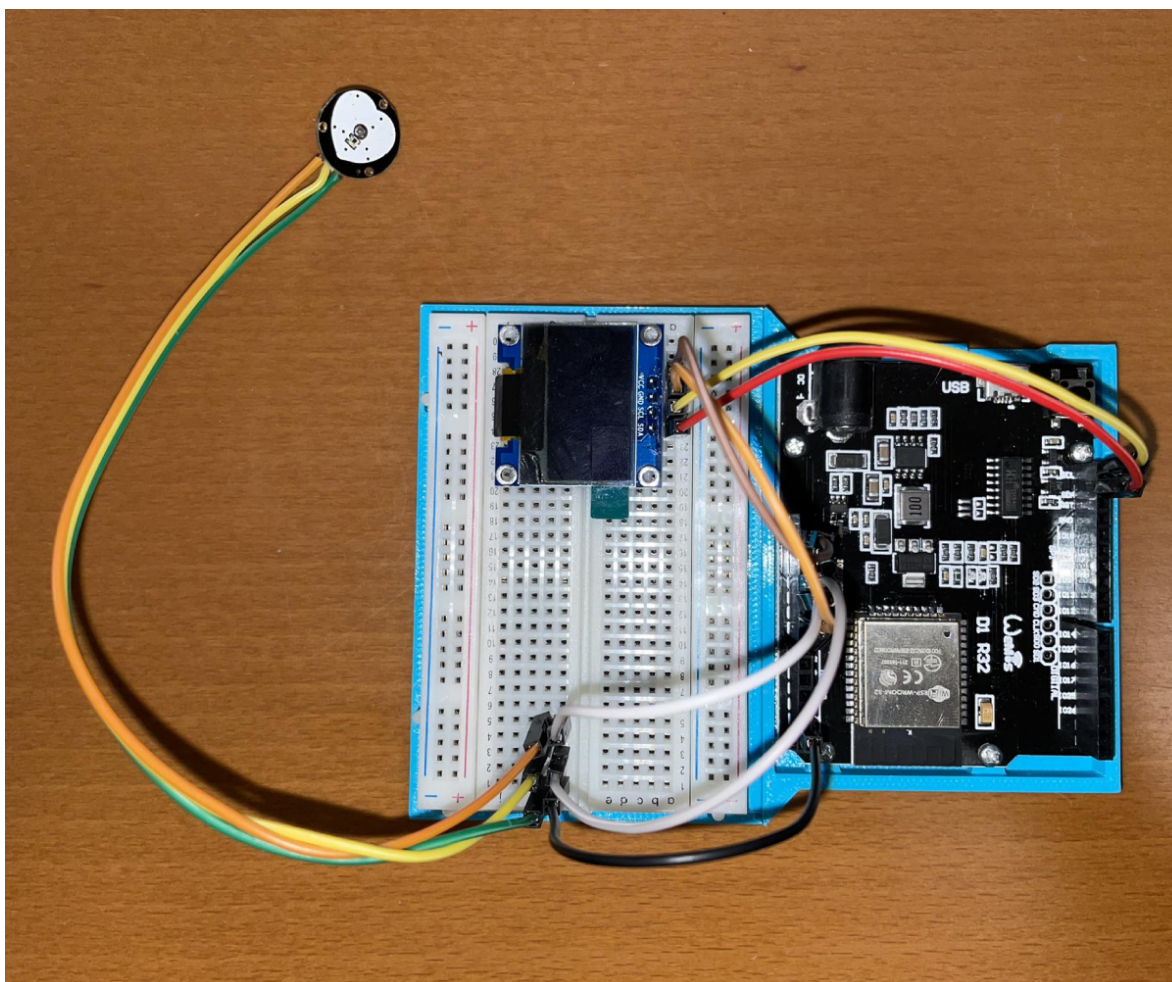
Displej byl zapojen do pinů SCL a SDA. Náapjení probíhá přes 5V (+) a GND (-), viz. obrázek 2.

### 3.2 Snímač srdečního tepu PulseSensor

Snímač byl také napájen přes 5V (+) a GND (-), jeho signál byl zapojen do pinu 36 (ADC0), viz. obrázek 2.



Obrázek 1: Návrh zapojení desky ESP32



Obrázek 2: Realizace zapojení desky ESP32

## 4 Potíže

Zapojení desky a následná konfigurace prostředí bylo bez zaváhání. Problém přišel u samotného snímače teploty, který rád ukazoval hodnoty až nesmyslné. Vždy také chvíli trvalo, než si prst „odchytí“ a začal ukazovat požadované hodnoty.

## 5 Řešení

Program využívá knihoven `Arduino.h`, `Wire.h` a `Adafruit_SSD1306.h` pro jednoduché ovládání displeje, jenž je nainicializován na začátku kódu. Dále je definována proměnná `PULSE_SENSOR_PIN`, která je nastavena na hodnotu 36, podle zapojení zmíněného v kapitole 3, a `NUMBER_OF_VALS` na 500, která bude později použita pro samotné odchytávání teploty.

Program nejprve vstoupí do funkce `setup()`, ve které se nastaví sériová komunikace na 9600 baudů (počet změn signálu za sekundu). Dále je poprvé nastaven displej na port `0x3C` a 2 sekundové zpoždění, aby nastavení proběhlo v pořádku.

Všechno získávání hodnot probíhá v nekonečném cyklu, který po dobu 10 sekund (500/20ms zpoždění) získává hodnoty ze senzoru (přes funkci `analogRead()`), kterou uložíme do pole `values`. Dále se v tomto cyklu také odečítá celkový čas z měření, který se postupně zobrazuje na displeji v hodnotách 9 (začátek měření) až 0 (konec měření).

Pro dosažení lepších výsledků jsem se rozhodl naimplementovat jednoduchý mediánový filtr, který prochází již zmíněné hodnoty, porovná je, seřadí vzestupně a nakonec vrátí hodnotu uprostřed pole. Tato metoda mi přinesla lepší výsledky než počítání průměru, jelikož tam nejsou zahrnovány extrémní výkyvy senzoru.

Po aplikování mediánového filtru program porovnává tuto hodnotu, zvýšenou o 40 (toto číslo fungovalo nejspolehlivěji), s hodnotami naměřenými z `analogRead()`. Při každém nález vyšší hodnoty inkrementuji proměnnou `anomaly` o 1.

V závěru cyklu se hodnota `anomaly` vynásobí 6, abychom dosáhli hodnoty za minutu. Dále jsem nastavil spodní hranici tepu na 40 a horní na 200, aby byl program ještě o trochu spolehlivější.

## 6 Funkcionalita a testování

Displej na začátku zobrazuje 0, poté naposledy naměřenou hodnotu společně s odpočtem právě probíhajícího měření.

Testování probíhalo při celé implementaci programu. Tep byl porovnáván s hodnotami, které by měly odpovídat mému fyzickému stavu. Avšak hodnoty se mohou lišit vzhledem k momentálnímu stavu jedince, jenž může být ovlivněn například stresem, únavou, množstvím užitého kofeinu, atd.

- Klidový stav: 50 - 100
- Fyzická námaha: 100 - 150

U konečných testů byl zkoušen klidový tep (tep se snižoval s délkou klidového stavu) a tep po fyzické námaze. Tyto testy jsem zdokumentoval pomocí videa a nahrál na Google Drive.

### Odkazy na testování:

[Snižující se klidový tep](#)

[Tep po fyzické námaze](#)

Problém s odkazy? Použijte [Zdroje](#).

## 7 Závěr

Projekt byl svou náplní jeden z těch zábavnějších. Jeho řešení je za správné funkčnosti senzoru odpovídající. V některých případech ale trvá, než si senzor sedne a začne snímat adekvátní hodnoty.

## 8 Zdroje

Při realizaci projektu bylo čerpáno z těchto zdrojů: [1] [2] [3] [4] [5]

Při problémech s hypertextovým odkazem u testování, použijte:

[https://drive.google.com/file/d/1o8epXSymkZcHvcDapgJTr4hES4HnGgzo/view?usp=drive\\_link](https://drive.google.com/file/d/1o8epXSymkZcHvcDapgJTr4hES4HnGgzo/view?usp=drive_link)

[https://drive.google.com/file/d/1SJeyl9q001pvIM8aHSfGlxWqP7FoRAWI/view?usp=drive\\_link](https://drive.google.com/file/d/1SJeyl9q001pvIM8aHSfGlxWqP7FoRAWI/view?usp=drive_link)

## Odkazy

- [1] Hadex. *Displej OLED 0,96", 128x64 znaků, IIC/I2C, 4piny, bílý*. <https://www.hadex.cz/m508b-displej-oled-096-128x64-znaku-iici2c-4piny-bily/>.
- [2] HW Kitchen. *Snímač pro tep srdce*. <https://www.hwkitchen.cz/snimac-pro-tep-srdce/>.
- [3] Ing. Václav Šimek. *ESP32*. [https://www.fit.vutbr.cz/~simekv/IMP\\_projekt\\_board\\_ESP32\\_Wemos\\_D1\\_R32.pdf](https://www.fit.vutbr.cz/~simekv/IMP_projekt_board_ESP32_Wemos_D1_R32.pdf).
- [4] Neznámý. *Monitor the Heart Rate using Pulse Sensor and Arduino*. <https://lastminuteengineers.com/pulse-sensor-arduino-tutorial/>.
- [5] technolabcreation. *Heartbeat Sensor using Arduino (Heart Rate Monitor)*. <https://www.technolabcreation.com/heartbeat-sensor-using-arduino-heart-rate-monitor/>.