

## **ESP32: Měření srdečního tepu**

Mikroprocesorové a vestavěné systémy

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
1.1	Zapojení komponent . . . . .	2
1.2	Vlastní způsob spuštění projektu . . . . .	3
<b>2</b>	<b>Popis řešení</b>	<b>3</b>
2.1	Funkce setup . . . . .	3
2.2	Funkce loop . . . . .	3
2.3	Funkce printDisplayInfo . . . . .	4
2.4	Funkce printDisplayNoFinger . . . . .	4
2.5	Funkce webServerInit . . . . .	4
2.6	Funkce handleRoot . . . . .	4
2.7	Funkce handleWeb . . . . .	4
2.8	ROZŠÍŘENÍ - Popis webového serveru . . . . .	5
<b>3</b>	<b>Zhodnocení</b>	<b>5</b>
<b>4</b>	<b>Video prezentující projekt</b>	<b>6</b>
<b>5</b>	<b>Autoevaluace</b>	<b>6</b>

# 1 Úvod

Tento dokument popisuje řešení projektu do kurzu Mikroprocesorové a vestavěné systémy na Fakultě informačních technologií.

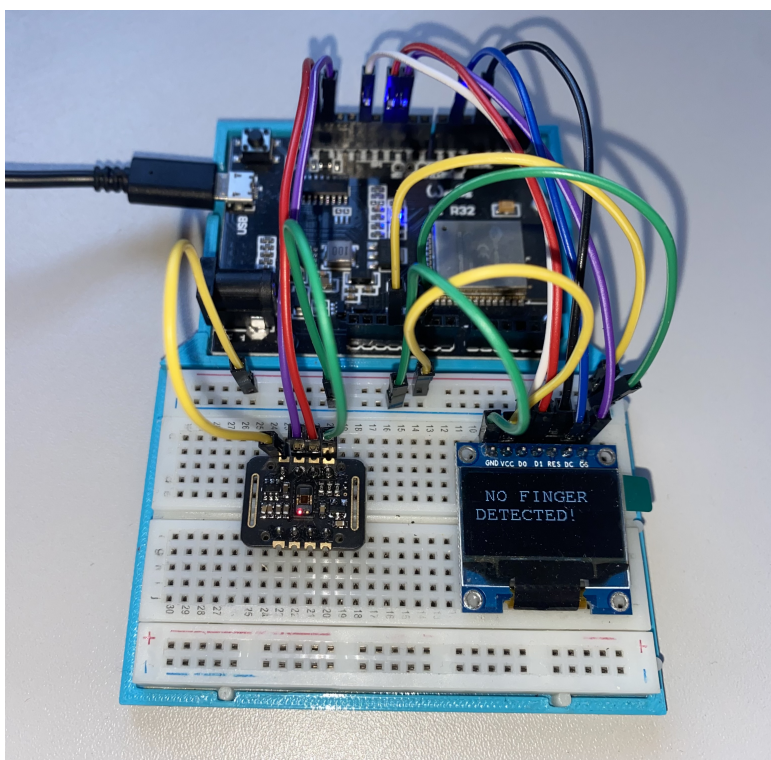
Cílem projektu bylo implementovat měřič srdečního tepu s využitím vývojové platformy na bázi SoC ESP32. Tep je snímán pomocí snímače MAX30102 [3] a aktuální hodnoty tepu jsou zobrazovány na OLED displej ssd1306 [6]. Navíc bylo implementováno i rozšíření, a to takové, že se modul ESP32 využije jako AP, jenž vytvoří vlastní web server, na kterém jsou asynchronně zobrazovány naměřené hodnoty.

Při implementaci bylo využito editoru Visual Studio Code a jeho rozšíření Platformio [4]. Projekt byl vytvořen za použití frameworku Arduino (zvolen z důvodu vyšší kvality knihoven pro naše komponenty) a těchto 3 knihoven:

- **Adafruit\_SSD1306** [2] (Knihovna pro práci s displejem ssd1306)
- **Adafruit-GFX-Library** [1] (Knihovna pro práci s displejem ssd1306)
- **SparkFun\_MAX3010x\_Sensor\_Library** [5] (knihovna pro práci se senzorem MAX30102)

## 1.1 Zapojení komponent

Displej ssd1306 je zapojen pomocí rozhraní SPI a senzor tepu je zapojen pomocí rozhraní I2C. Obě tyto komponenty jsou napájeny pomocí napětí 3.3 V



Obrázek 1: Detail zapojení desky

## 1.2 Vlastní způsob spuštění projektu

Je nutné mít nainstalovaný editor Visual Studio Code s nainstalovaným rozšířením Platformio. Následně se klikne na ikonu rozšíření Platformio v levém panelu VSC, kde zvolíme možnost Quick Access/PIO Home/Open --> New Project. Dále se nám zobrazí dialogové okno pro vytvoření projektu. Zadáme jméno nového projektu, ve výběru „Board“ vybereme Espressif ESP32 Dev Module a možnost „Framework“ ponecháme Arduino a klikneme na tlačítko finish. Následně se nám zobrazí struktura projektu a soubor `platformio.ini`. Do tohoto souboru nakonec dopíšeme řetězec `monitor_speed = 115200`.

Do složky `src` v našem nově vytvořeném projektu vložíme zdrojové soubory tohoto projektu `main.cpp` a `index.h`.

Dále je nutné si doinstalovat do projektu potřebné knihovny. To provedeme tak, že se klikne na ikonu rozšíření Platformio v levém panelu VSC, kde zvolíme možnost Quick Access/PIO Home/Libraries. Mělo by se nám zobrazit vyhledávací pole, kde postupně vyhledáme tyto 3 knihovny: Adafruit SSD1306, Adafruit GFX Library, SparkFun MAX3010x Pulse and Proximity Sensor Library. Každou z těchto knihoven je potřeba po vyhledání přidat do nově vytvořeného projektu. Následně již stačí připojit naší ESP32 desku se správně zapojenými komponentami a zvolit možnost Upload and Monitor v levém panelu editoru VSC.

## 2 Popis řešení

Samotné řešení projektu se nachází v souborech `main.cpp` a `index.h`.

Soubor `main.cpp` obsahuje všechny funkce, které řídí měření tepu, zobrazování dat na displej či spuštění webového serveru.

Soubor `index.h` obsahuje strukturu zobrazované webové stránky na web serveru a skript využívající asynchronní komunikace pro zobrazování aktualizovaných dat na webu.

### 2.1 Funkce setup

Na počátku této funkce se vytváří proměnné potřebné k inicializaci snímače tepu. Snímač tepu je inicializován např. s hodnotou intenzity svitu LED diody, či módem svitu (jen červená led/ infračervená + červená led) apod. Vstupní proud diod je nastaven na cca 6,25 mA. Při vyšším proudu již nedokáže snímač kvalitně rozeznat změny ve snímaných hodnotách. Dále se inicializuje displej, na kterém budeme zobrazovat měřené hodnoty a pomocí knihovnických funkcí nastavíme vlastní font, velikost, barvu, kódování zobrazovaného textu a místo pozici na displeji kam budeme chtít vykreslovat údaje.

### 2.2 Funkce loop

Tato funkce je hlavní smyčkou programu, ve které se nachází samotný výpočet uživatelského srdečního tepu. Výpočet funguje tak, že se pomocí knihovnické funkce `checkForBeat` kontroluje, zda byl zachycen úder srdce. Pokud tomu tak je, vypočítá se čas  $\Delta t$  [ms] od posledního zachyceného úderu srdce. Počet úderů srdce za minutu je následně vypočítán tímto vztahem  $BpM = \frac{60}{\Delta t/1000}$ .

Dále se také ze zachycených hodnot `BpM` vypočítá průměrná hodnota uživatelského tepu. Ta se počítá tak, že máme vytvořené pole o velikosti 8, které postupně naplňujeme hodnotami `BpM`. Z těchto hodnot je následně vypočítána průměrná hodnota uživatelského tepu. Dále můžeme z přečtených hodnot

snímače zjistit, zda se uživatel dotýká senzoru či nikoliv. Pokud tedy zjistíme, že je hodnota pod nějakou úroveň, tak voláme funkci `printDisplayNoFinger` 2.4. V opačném případě začneme vypisovat hodnoty na displej voláním funkce `printDisplayInfo` 2.3. Na konci funkce `loop` voláme knihovní funkci `handleClient`, která obslouží klienta, který se chce připojit k našemu web serveru.

## 2.3 Funkce `printDisplayInfo`

Funkce `printDisplayInfo` přijímá jako vstupní parametry počet tepů srdce za minutu a průměrný uživatelský tep. Tyto hodnoty se pomocí knihovních funkcí zobrazují na připojeném OLED displeji. Na počátku funkce se musí displej vymazat pomocí funkce `clearDisplay`.

## 2.4 Funkce `printDisplayNoFinger`

Tato funkce zobrazuje uživateli informaci, že není ke snímači přiložen prst. K zobrazení této informace se využívá funkcí z knihovny `Adafruit_SSD1306` [2].

## 2.5 Funkce `webServerInit`

V této funkci se nastavuje WiFi mód našeho zařízení. Tento mód nastavíme na AP tímto příkazem: `WiFi.mode(WIFI_AP)`. Dále je nastaveno jméno naší WiFi sítě, které je nastaveno na `XVASIN11-ESP32-AP`. WiFi nemá nastavené žádné heslo pro jednodušší přístup. Dále je zavolána funkce `handleRoot` 2.6, která na web načte HTML strukturu stránky ze souboru `index.h` s asynchronním skriptem, který aktualizuje informace na webu. Nakonec je zavolána funkce `handleWeb`, která zašle aktualizovaná data na web.

## 2.6 Funkce `handleRoot`

Funkce načte ze souboru `index.h` HTML strukturu a tu zašle na server.

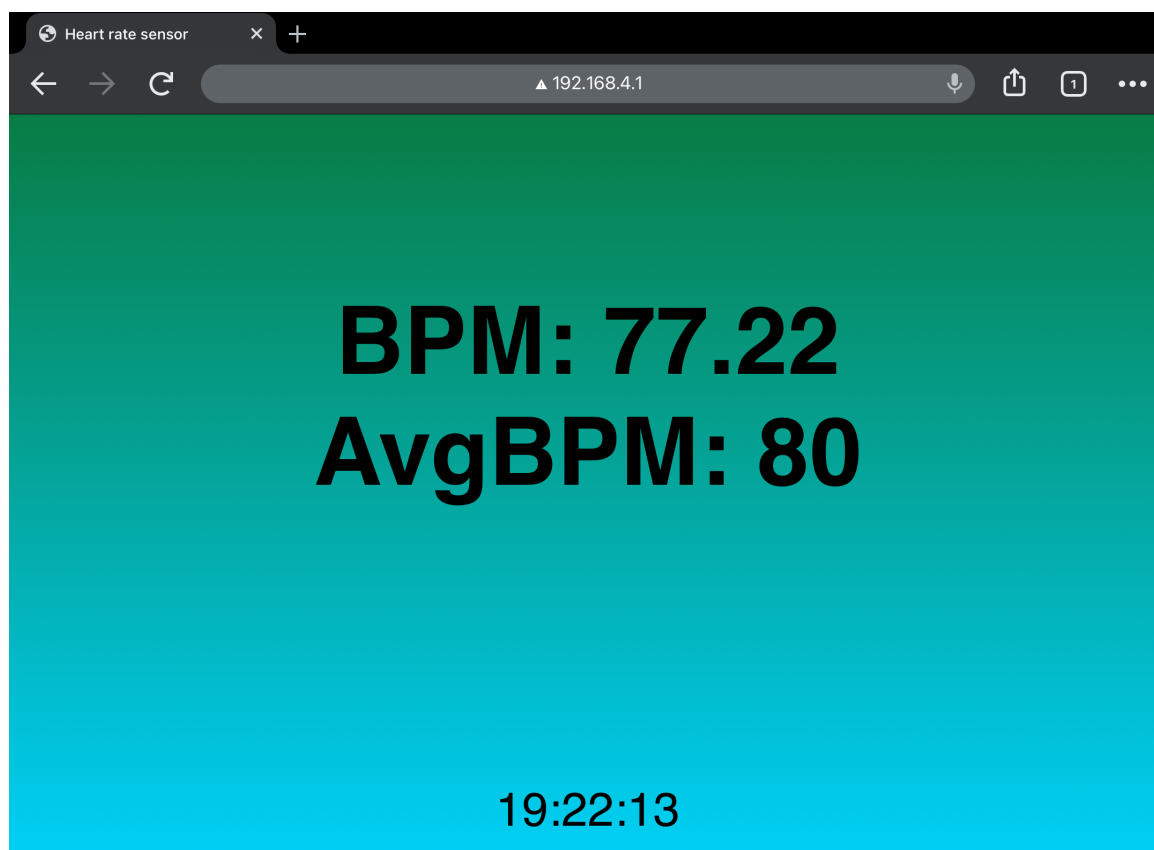
## 2.7 Funkce `handleWeb`

Tato funkce zašle aktualizovaná data o uživatelské tepové frekvenci v text/plain formátu na web server.

## 2.8 ROZŠÍŘENÍ - Popis webového serveru

Po připojení napájení do naší platformy na bázi SoC ESP32 se spustí, nejen měření tepu a displej zobrazující naměřené hodnoty, ale spustí se i webový server umožňující zobrazit si naměřené hodnoty na mobilním zařízení.

Aby uživatel mohl číst tyto hodnoty musí se na svém mobilním zařízení připojit na WiFi síť s názvem XVASIN11-ESP32-AP (WiFi je bez hesla) a do prohlížeče zadat adresu **192.168.4.1**. Na této webové adrese bude uživateli zobrazen výstup měření jeho srdečního tepu.



Obrázek 2: Ukázka webového rozhraní AP

## 3 Zhodnocení

Mnohými měření tepu jsem zjistil, že pro získání korektních a stabilních hodnot, je opravdu velmi důležité udržovat konstantní tlak na senzor tepu. Projekt mi jako takový nepřišel nějak moc složitý, proto jsem se rozhodl navíc implementovat zobrazení naměřených hodnot na vlastním web serveru viz 2.8. Ověření funkčnosti probíhalo simultánním měřením srdečního tepu na platformě ESP32 s využitím senzoru MAX30102 a na mobilním telefonu Samsung Galaxy S9, který má také senzor pro měření tepu. Po provedení tohoto paralelního měření byly získané hodnoty v celku podobné, z čehož usuzuji, že mé řešení měření tepu odpovídá realitě.

## 4 Video prezentující projekt

Níže jsou uvedené odkazy k videu prezentující funkční řešení projektu.

Odkaz na přehraní videa na YouTube: <https://youtube.com/shorts/GdGj5rN2Gnw?feature=share>

Odkaz na stažení daného videa z Google Drive:

<https://drive.google.com/file/d/1TznYSvp2Ogxsz00HByiR1h28raW7ji5f/view?usp=sharing>

## 5 Autoevaluace

Celkové hodnocení je dáno tímto vztahem:  $\Sigma = (K_1 + K_2 \cdot F/5) \cdot (E + F + Q + P + D)$   
Hodnoty jednotlivých složek mého projektu:

- Konstanty  $K_1 = 0,25$
- Konstanty  $K_2 = 0,75$
- Přístup  $E = 1$   
Projekt jsem začal implementovat i dokončil s předstihem. Nad rámec zadání jsem implementoval webový server zobrazující naměřená data.
- Funkčnost  $F = 5$   
Cílem projektu bylo implementovat systém pro měření srdečního tepu. Tuto informaci má implementace úspěšně měřit a zobrazuje na připojený OLED displej i na webový server.
- Kvalita  $Q = 3$   
Řešení projektu je pro uživatele velmi přívětivé. Uživateli stačí pouze připojit napájení a téměř okamžitě může začít měřit své hodnoty srdečního tepu. Dále je díky nezabezpečenému AP, velmi jednoduché se na AP připojit. Po přístupu na určenou adresu jsou uživateli okamžitě zobrazeny naměřené hodnoty. Zdrojové kódy jsou okomentovány a rozděleny do funkcí, což zajišťuje jejich přehlednost a čitelnost. Implementace projektu je dekomponována na menší problémy, které jsou řešeny v jednotlivých funkcích (zobrazování dat na displeji/webu apod.).
- Prezentace  $P = 1$   
Video prezentující řešení plně zobrazuje funkčnost podle zadání i implementované rozšíření o webový server zobrazující měřené hodnoty.
- Dokumentace  $D = 4$   
Dokumentace splňuje všechny požadované komponenty jako jsou úvod do problému, popis implementace a zhodnocení.

Po dosazení hodnot do výše uvedeného vzorce nám vyjde celkový počet bodů:

$$\Sigma = (0,25 + 0,75 \cdot 5/5) \cdot (1 + 5 + 3 + 1 + 4) = 14 \text{ bodů}$$

## Použitá literatura

- [1] *Adafruit-GFX-Library*. [vid. 2022-12-13]. Dostupné z: <https://github.com/adafruit/Adafruit-GFX-Library>.
- [2] *Adafruit\_SSD1306*. [vid. 2022-12-13]. Dostupné z: [https://github.com/adafruit/Adafruit\\_SSD1306](https://github.com/adafruit/Adafruit_SSD1306).
- [3] *MAX30102*. [vid. 2022-12-13]. Dostupné z: <https://www.hwkitchen.cz/max30102-snimac-pro-srdecni-tep-a-pulzni-oxymetr/>.
- [4] *Platformio*. [vid. 2022-12-13]. Dostupné z: <https://platformio.org/>.
- [5] *SparkFun\_MAX3010x\_Sensor\_Library*. [vid. 2022-12-13]. Dostupné z: [https://github.com/sparkfun/SparkFun\\_MAX3010x\\_Sensor\\_Library](https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library).
- [6] *Ssd1306*. [vid. 2022-12-13]. Dostupné z: <https://www.hadex.cz/m508a-displej-oled-096-128x64-znaku-7pinu-bily/>.