

IMP- Mikroprocesorové a vestavěné systémy
Š- ESP32: Měření srdečního tepu
[digitální senzor]

Obsah

Úvod	3
Návrh - úvod do problému	3
Vývojové prostredie	3
Použité knižnice	3
Zapojenie.....	3
Implementácia – popis riešenia	4
Použitie	5
Testovanie	5
Test č. 1 – Prst je na senzore – prebieha meranie a zobrazenie údajov	6
Test č.2 – Prst nie je na senzore – vyvolanie upozorňovacej hlášky na displeji.....	6
Záver	7
Autoevaluácia	7
Literatúra	7

Úvod

V projekte bolo cieľom realizovať systém pre meranie srdečného tepu pomocou senzoru MAX30102 a vývojového kitu na báze SoC ESP32 s periferným zariadením, čiže OLED displejom pripojeného cez SPI rozhranie. Vývojová doska pre daný projekt bola *Wemos D1 R32*[1]. Aby som využil potenciál senzoru, doplnil som si ako cieľ projektu funkcionality miery okysličenia krvi.

Návrh- úvod do problému

Spočiatku bolo esenciálne si preštudovať nielen možnosti vývojového prostredia, ale aj spôsoby zapojenia či samotné výpočty kľúčové pre naše funkcionality.

Vývojové prostredie

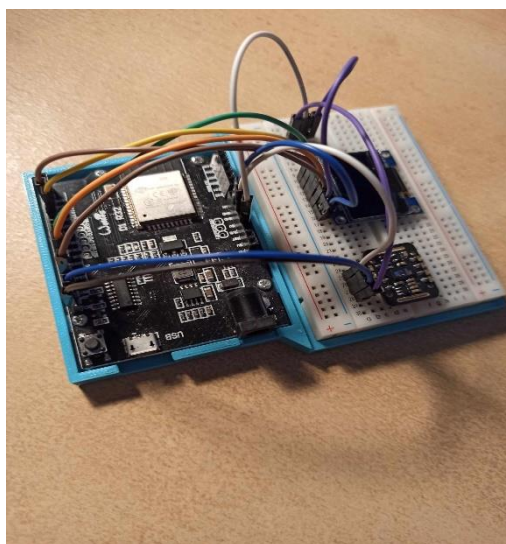
Ako vývojové prostredie som spočiatku využíval iba *Arduino IDE 2.2.1*. Z dôvodu lepšej štruktúry odovzdávaných súborov som zvolil *Visual Studio Code*, rozšírenie *PlatformIO*. Toto rozšírenie som nakoniec využil na finálne nahrávanie kódu do vývojového kitu.

Použité knižnice

- **<Wire.h>** - pre komunikáciu s I2C zariadeniami
- **<Adafruit_SSD1306.h>** - pre monochrome OLED 128x64
- **MAX30105.h** – pre získanie pulzu a proximity pomocou senzoru MAX30102
- **heartRate.h** – pre výpočet srdečného tepu na základe *Heart Rate Detection (PBA algoritmu)*

Zapojenie

Po naštudovaní poskytnutých materiálov k projektu som na základe zdroja[1], kde je popísaná schéma zapojenia dosky *Wemos D1 R32* a zdroja[2] s popisom zapojenia senzoru MAX30102 najskôr zapojil túto komponentu. Následne pomocou popisu zapojenia [3] som zapojil aj OLED displej cez SPI rozhranie, aby som zistil funkcionality aj tohto komponentu. Na obrázku č.1 môžete vidieť konečné zapojenie spomínaných komponentov. Tabuľka č. 1 ukazuje detailné zapojenie portov.



Obrázok č. 1: Zapojenie OLED displeja a senzora MAX30102

Senzor MAX30102	OLED Displej	Vývojový kit Wemos D1 R32
GND	-	GND
SCL	-	GPIO22 (SCL)
SDA	-	GPIO21 (SDA)
VIN	-	3V3
-	GND	GND
-	VCC	3v3
-	D0	GPIO18
-	D1	GPIO23
-	RES	GPIO17
-	DC	GPIO16
-	CS	GPIO05

Tabuľka č. 1: Detailné zapojenie portov OLED displeja a senzoru MAX30102

Implementácia – popis riešenia

Projekt som implementoval vo variante jazyka C/C++, Arduino. Pomocou knižníc popísaných v sekcii „Použité knižnice“ som následne implementoval hlavné funkcie *void* a *loop*. Následne bolo vhodné z dôvodu správnej dekompozície a granularity implementovať inicializačné funkcie *initializingBoth* a *particleSettingUp*, výpočtové funkcie *heartRateComp* a *computingSpO2* a nakoniec funkcie pre výpis na displej *showWarning* a *showOnDisplay*.

V úvode je dôležité namapovať porty vzhľadom na zapojenie. Tieto premenné si zadefinujeme pomocou 5 globálnych premenných. Následne si určíme globálne premenné pre výpočet merania tepu a priemeru za minútu *rateSpot*, *lastBeat*, *beatAvg*, *RATE_SIZE*, *rates[RATE_SIZE]* a *beatsPerMinute*. Taktiež boli definované globálne premenné pre výpočet SpO2, filtračné faktory a sledovanie vzoriek, kde bol určený vzorkovací interval *Num* s hodnotou 100, ktorý bude potrebný pri popise funkcionality merania okysličenia krvi.

Na začiatku funkcie *setup* ako prvé nastavím prenosovú rýchlosť 115 200 *Bd*. Následne je potrebné inicializovať obe komponenty (displej a senzor), čo nám zabezpečuje funkcia *initializingBoth*. Ďalej v *setup* nastavujem konfiguráciu pre senzor MAX30102 pomocou funkcie *particleSettingUp*, kde jedným z parametrov je napríklad jasnosť LED.

Vo funkcii *loop* ako prvé definujem hodnoty *ir*, *red*, *fred*, *fir* potrebné pre výpočet úrovne okysličenia krvi, taktiež aj *SpO2*. Následne vstupujem do while cyklu, kde na základe funkcií *getFIFORed* a *getFIFOIR* dostanem vzorku pre postupne *red* a *ir* potrebné v ďalších krokoch. Následne inkrementujeme počítadlo *i* a hodnoty priradíme k hodnotám *fred* a *fir* float hodnoty *red* a *ir*. Tieto všetky premenné budú následne vstupom pre funkciu *computingSpO2*. Pomocou vzťahov

$$avered = avered * frate + (double)red * (1.0 - frate)$$

$$aveir = aveir * frate + (double)ir * (1.0 - frate)$$

zistím priemerné levely IR a red, kde *frate* je hodnota low pass filtru pre IR/red LED hodnoty (v našom prípade 0,95). Tieto hodnoty sa následne ukladajú do premenných *sumredrms* a *sumirrms*, ktoré reprezentujú sumy pre IR/red hodnoty, ktoré sa zapisujú pomocou vzťahov:

$$sumredrms += (fred - avered) * (fred - avered)$$

$$sumirrms += (fir - aveir) * (fir - aveir)$$

Tento proces nastane vždy ak index i aktuálneho vzorku je menší hodnote 100. Po dosiahnutí indexu tejto hodnoty nastane výpočet finálnej hodnoty $ESpO_2$ v % na základe nasledujúcich vzťahov, kde R je ratio potrebné pre výpočet SpO_2 , $FSpO_2$ je filter faktor pre určenie SpO_2 , $ESpO_2$ je finálna hodnota okysličenia krvi v %, ktorá je zobrazená na displeji:

$$\text{double } R = (\text{sqrt}(\text{sumredrms}) / \text{avered}) / (\text{sqrt}(\text{sumirrms}) / \text{aveir})$$

$$SpO_2 = -23.3 * (R - 0.4) + 100$$

$$ESpO_2 = FSpO_2 * ESpO_2 + (1.0 - FSpO_2) * SpO_2$$

Všetky tieto výpočty ako aj priebeh ich konania sú overené podľa zdroja [4].

Meranie srdečného tepu sa začína v časti while cyklu vo funkcii *loop*, kde si najskôr pomocou library funkcie *getIR* zistím okamžitú IR hodnotu, ktorá je prezentovaná pomocou *irValue*, tá je následne predaná ako parameter library funkcie *checkForBeat*, ktorá nám slúži na zachytenie prítomnosti tepu. Ak je tep zachytený, zavolám funkciu *hearRateComp*, kde si najskôr určím hodnotu *delta*, kľúčovú pre správny výpočet úderov srdca za minútu, *beatsPerMinute*. Ten je definovaný pomocou vzťahov

$$\text{long } \text{delta} = \text{millis}() - \text{lastBeat}$$

$$\text{lastBeat} = \text{millis}()$$

$$\text{beatsPerMinute} = 60 / (\text{delta} / 1000.0)$$

kde nám library funkcia *millis* vráti hodnotu doby behu programu na zariadení. Táto hodnota je kľúčová pre výpočet *delta*, ktorý následne pomocou posledného výpočtu použijem pre finálnu hodnotu počet úderov za minútu.

Počet úderov za minútu však nie je vždy detekovaný najlepšie a preto je potrebný výpočet priemerného počtu úderov za minútu. Priemerné hodnoty vytváram z posledných 4 načítaných vzorkov. To znamená že po načítaní štvrtého prvku sa suma týchto vzorkov vydolí ich počtom a výsledok sa uloží do premennej *beatAvg*. [5]

V prípade ak nie je prst na senzore, upozorním užívateľa výpisom na displej, čo nám zabezpečuje funkcia *showWarning*. Funkcia *showOnDisplay* nám naopak zobrazuje všetky hodnoty, ktoré som si na začiatku ustanovil ako dôležité zobrazíť, a to hodnotu srdečného tepu *beatsPerMinute*, priemernú hodnotu srdečného tepu *beatAvg* a percentuálnu hodnotu miery okysličenia krvi *ESpO_2* na OLED displeji.

Použitie

Spôsob podrobného použitia a prezentovanie funkcionality je zhrnuté v okomentovanom videu. Užívateľ po spustení zariadenia musí najskôr položiť prst na senzor, následne mu budú zobrazené 3 aktuálne hodnoty srdečného tepu, priemerného srdečného tepu a percentuálna miera okysličenia krvi.

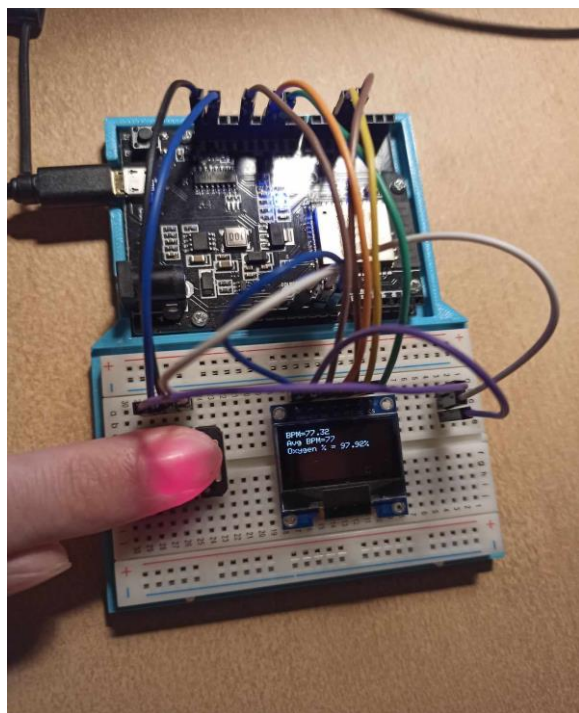
Testovanie

Pre testovanie aplikácie som pripojil vývojový kit Wemos D1 R32 do USB portu môjho počítaču. Následne po uploadnutí kódu do zariadenia sa po priložení prstu na displej začali merať a zobrazovať

potrebné hodnoty. V prípade, ak užívateľ nadvihol prst mimo senzor, tak sa vypísala na displej varovná hláška.

Test č. 1 – Prst je na senzore – prebieha meranie a zobrazenie údajov

Po položení prstu na senzor sa začali zobrazovať namerané a vypočítané potrebné údaje na displej vid'. obrázok č.2 a obrázok č.3.



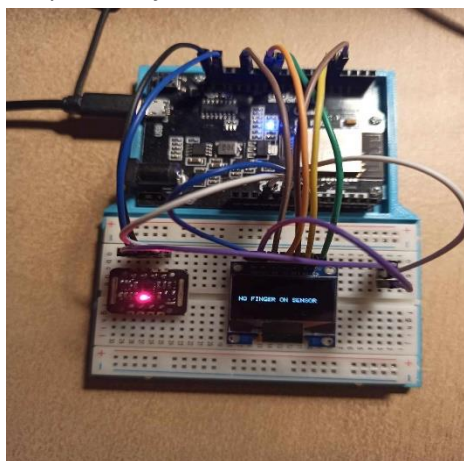
Obrázok č. 2: Zaznamenávanie údajov na displej



Obrázok č. 3: Výsledok na displeji pri teste

Test č.2 – Prst nie je na senzore – vyvolanie upozorňovacej hlášky na displeji

Po dostatočnom oddialení prsta zo senzoru sa nám naozaj zobrazila hláška upozornenia, čo reprezentuje obrázok č.4 a č.5.



Obrázok č.4 – Upozornenie na displeji



Obrázok č.5 – Priblíženie upozornenia

Záver

Ako bolo vidno počas testovania, overenie funkčnosti aplikácie bolo dosiahnuté s obohatením o zobrazenie miery okysličenia krvi. Vznikla nám teda aplikácia, kde OLED displej komunikujúci cez rozhranie SPI a senzor obe napojené na vývojovú dosku Wemos D1 R32 vykonávajú funkcionality zobrazenie srdečného tepu za minútu, priemerného srdečného tepu za minútu a percentuálnu mieru okysličenia krvi. Tieto všetky údaje sa zobrazujú na OLED displeji.

Odkaz na demonštračné video: <https://drive.google.com/file/d/178wGRnf-VXFOlBafqhQGPT5gx7SDj8XH/view?usp=sharing>

Autoevaluácia

- E – Dovolím si povedať, že na projekte som začal pracovať včas, bolo to týždeň pred odovzdaním a vo finálnych častiach som sa snažil dekomponovať výsledný kód na prehľadnejšie časti, aby bol môj kód čitateľnejší a reprezentatívnejší
- F – Okrem základnej funkcionality a to merania tepu a zistenie priemerného tepu, som nakoniec doplnil funkcionality o meranie miery okysličenia krvi, keďže už predčasne som mal dosiahnuté vlastnosti funkcionality hotové.
- Q – Osobne si myslím a dúfam, že je to aj z videa jasné, že užívateľovi je to dostatočne príjemné a jednoduché pre pochopenie a ovládanie riešenia, takisto prehľadnosť zdrojových kódov už som spomenul v predošlom bode a samotná dekompozícia prebehla tak, aby jednotlivé operácie boli čo najprehľadnejšie vo finálnom kóde
- P – Na prezentovanie som sa poctivo pripravoval, dopredu som si určil scenár, na základe ktorého som chcel osloviť pozerajúceho človeka a zároveň mu predviesť, čoho je schopné finálne riešenie.
- D – Na záver dokumentácia. Úprimne si myslím, že túto dokumentáciu som vypracoval s veľkým úsilím o čo najlepší výsledok, keďže je dôležité, aby na základe nej bolo vidno celý priebeh riešenia. Samotný proces riešenia som sa snažil opísať tak podrobne a dôležito, ale zároveň výstižne, ako to len išlo, o čom zodpovedá aj jeho rozsah. Čo sa týka zhrnutia neistôt, tak je samozrejme dôležité povedať, že senzor nie je vždy dokonalý, ale keď sa jedná o dostatočne rozsiahlu vzorku meraných hodnôt, tak tieto hodnoty sa približovali skutočným možným hodnotám.

Na úplný záver by som chcel dodať, že tento celé riešenie tohto projektu vrátane dokumentácie a všetkých zložiek vyššie popísaných som sa snažil. Jediné, čomu som možno nevenoval v dokumentácii väčší prejav, boli neistoty a možné chyby pri meraní, ale tie som každopádne zmienil tu. Ak by som sa mal ohodnotiť týchto zložiek, tak by som súhlasil s ohodnotením 13.5 bodov.

Literatúra

[1] Šimek, V.: IMP_projekt_board_ESP32_Wemos_D1_R32.pdf. [online], [vid. 2023-12-11]. Dostupné z: https://www.fit.vutbr.cz/~simekv/IMP_projekt_board_ESP32_Wemos_D1_R32.pdf

[2] Dostupné z: <https://microcontrollerslab.com/esp32-heart-rate-pulse-oximeter-max30102/>

[3] Teja, R. : [vid. 2021-02-24]. Dostupné z: <https://www.electronicshub.org/esp32-oled-display/>

[4] Dostupné z: <https://tinkerfoundry.blogspot.com/2023/05/esp32-based-pulse-oximeter-using.html>

[5] Dostupné z: <https://surtrtech.com/2019/02/05/measure-heart-pulses-bpm-and-oxygen-saturation-spo2-with-max30102-and-arduino/>