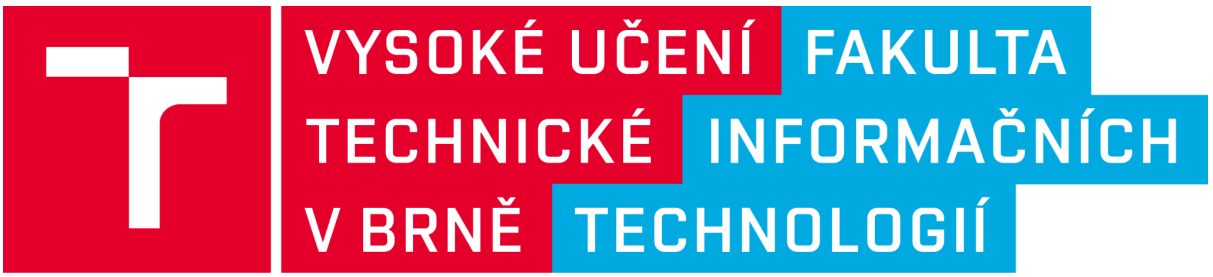


Mikroprocesorové a vestavěné systémy - projekt

Měření srdečního tepu
[digitální senzor]

dokumentace

Kristián Dobeš - xdobes22

14. prosince 2023

Úvod do problému

Cílem projektu byla implementace systému pro měření srdečního tepu pro mikrokontrolér Espressif ESP32, který pomocí snímače srdečního tepu MAX30102 zobrazí uživatelův tep na grafický OLED displej.

Pro implementaci projektu jsem použil vývojové prostředí Arduino kvůli jeho jednoduchosti použití a kvalitě standardních knihoven.

Popis řešení

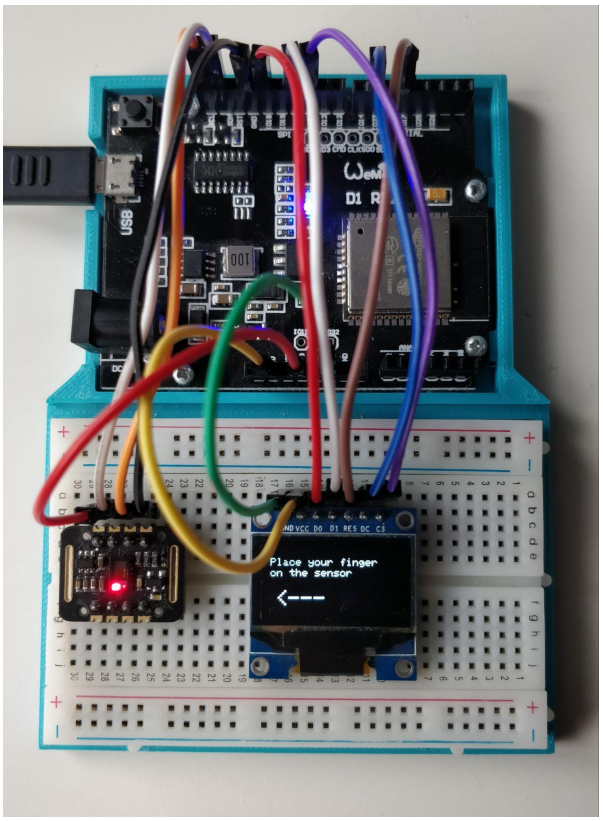
Hardware

- Mikrokontrolér: Espressif ESP32, deska: Wemos D1 R32.
- Senzor Srdečního Tepu: MAX30102.
- Displej: OLED displej s rozlišením 128x64 pixelů, 7 pinů, Driver: SSD1306.

Komunikační Rozhraní:

- I2C: použité pro komunikaci mezi ESP32 a senzorem MAX30102.
- SPI: použité pro připojení OLED displeje k ESP32.

Zapojení:



Fotografie zapojení displeje a senzoru do desky

Implementace

Použité knihovny

Knihovna SparkFun MAX3010x Pulse and Proximity Sensor Library [2]

- MAX30105.h

Pro práci se senzorem MAX30102 slouží knihovna MAX30105.h. Umožňuje snadné načítání dat z čidel a jejich zpracování.

V projektu je použita pro nastavení parametrů senzoru a snadné čtení hodnot srdečního tepu (funkce getIR()).

- heartRate.h

Knihovna slouží pro zpracování hodnot srdečního tepu získaných ze senzoru. V projektu byla použita pro funkci checkForBeat() sloužící k detekci srdečního tepu.

Knihovny Adafruit_GFX.h a Adafruit_SSD1306.h

Tyto dvě knihovny poskytují funkce a metody pro ovládání OLED displejů, konkrétně modelu SSD1306. Umožňují snadné zobrazení hodnoty tepu nebo ikony srdce na displej.

Adafruit_GFX.h je základní grafická knihovna, která poskytuje obecné grafické funkce. V projektu například použití funkce drawBitmap pro vykreslení ikony srdce nebo funkce setTextSize pro nastavení velikosti textu na displeji.

Adafruit_SSD1306.h je specifická pro ovládání displeje SSD1306.

Program

Inicializace komponent: Na začátku programu je inicializována sériová komunikace, OLED displej a senzor srdečního tepu s vhodnými parametry.

V hlavní smyčce je prvně zkontrolována hodnota infračerveného světla detekovaného senzorem (funkce particleSensor.getIR()), aby se zjistilo, zda je na senzoru prst. Pokud je hodnota IR > 7000, program prst detekuje a pokračuje výpočtem BPM.

Funkce checkForBeat(irValue) vrací hodnotu true, pokud je detekován srdeční tep. Po detekci tepu je vypočítána tepová frekvence a průměrná hodnota BPM ze série měření.

Výpočet BPM: Když je detekován tep, zaznamená se aktuální čas (funkce millis()), tento čas se porovná s časem posledního detekovaného tepu, aby se určila doba mezi dvěma srdečními tepy - delta (nebo také Heart rate variability - HRV).

BPM (tep za minutu) = 60 / (delta / 1000.0)

Každá nově vypočtená hodnota BPM se ukládá do pole rates[], které je cyklicky aktualizováno, takže obsahuje pouze několik posledních měření BPM.
Průměrná hodnota BPM se vypočítává jako součet hodnot v poli rates děleno jejich počtem. Tím se získá průměrná hodnota z nedávných měření, což pomáhá vyhladit krátkodobé fluktuace a poskytuje stabilnější a spolehlivější odhad srdečního tepu.

Výpis na displej: Průměrná hodnota BPM je zobrazena na displeji včetně času mezi tepy srdce. Program dynamicky mění grafiku na displeji, zobrazuje buď malé nebo velké srdce, v závislosti na tom, zda byl právě detekován srdeční tep. Indikátor detekce srdečního tepu je také znázorněn bliknutím červené LED na desce včetně času mezi jednotlivými tepy srdce (HRV).

Grafika na displeji byla vygenerována programem LCD Assistant[3], převedením obrázků srdečního tepu do datového pole. Použité rozměry jsou 24x21 px pro malé srdce[3] a 32x32 pixelů pro velké srdce[3].

Pokud hodnota IR klesne pod hranici 50000, program nedetekuje prst na senzoru a vypíše na displej zprávu uživateli včetně ukazatele na senzor.

Zhodnocení

Tento projekt úspěšně demonstruje využití ESP32 a senzoru MAX30102 pro měření srdečního tepu. Integrace s OLED displejem efektivně poskytuje vizuální zpětnou vazbu. Díky Arduino IDE a dostupným knihovnám bylo možné rychle vyvinout a testovat funkční implementaci.

Jedním z omezení projektu je potřeba udržovat stabilní polohu a tlak prstu na senzoru pro přesné měření. Převážně z toho důvodu jsem nebyl schopný implementovat měření okysličení krve, jelikož i přes několik pokusů o změně polohy/tlaku prstu na senzoru a době měření nebyla většina dat získaných měřením validní (viz demonstrace řešení).

Demonstrace řešení: https://youtu.be/KCjrCQ5FxI4

Zdroje

1. Logo na úvodní stránce
    a. Fakulta informačních technologií Vysokého učení technického v Brně
        i. https://www.vut.cz/jvs
2. Příklady z knihovny SparkFun MAX3010x Pulse and Proximity Sensor Library
    a. https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library/tree/master
    b. kostra programu, základní měření tepu, převzato a upraveno pro tento projekt
        i. https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library/blob/master/examples/Example5_HeartRate/Example5_HeartRate.ino
    c. kód pro měření okysličení krve, upraven kvůli spustitelnosti pro tento projekt, použito v demonstračním videu
        i. https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library/blob/master/examples/Example8_SPO2/Example8_SPO2.ino
3. BMP ikony
    a. malé srdce
        i. https://www.onlinewebfonts.com/icon/492378
    b. velké srdce
        i. https://www.klipartz.com/en/sticker-png-mkwng
    c. program LCD Assistant
        i. https://en.radzio.dxp.pl/bitmap_converter/