

IMP - Meranie srdcového tepu digitálnym senzorom

Richard Seipel - xseipe00



Obsah

1	Popis	2
2	Použitý hardvér	2
3	Schéma zapojenia	2
3.1	Rozhrania	3
4	Implementácia programu	3
4.1	Vývojové prostredie	3
4.2	Použité knižnice	3
4.3	Makrá a globálne premenné	3
4.3.1	Displej	3
4.3.2	Snímač tepu	3
4.4	Vykresľovanie na displej	3
4.5	Počiatkové nastavenia	4
4.6	Hlavný cyklus	4
5	Demonštračné video	4

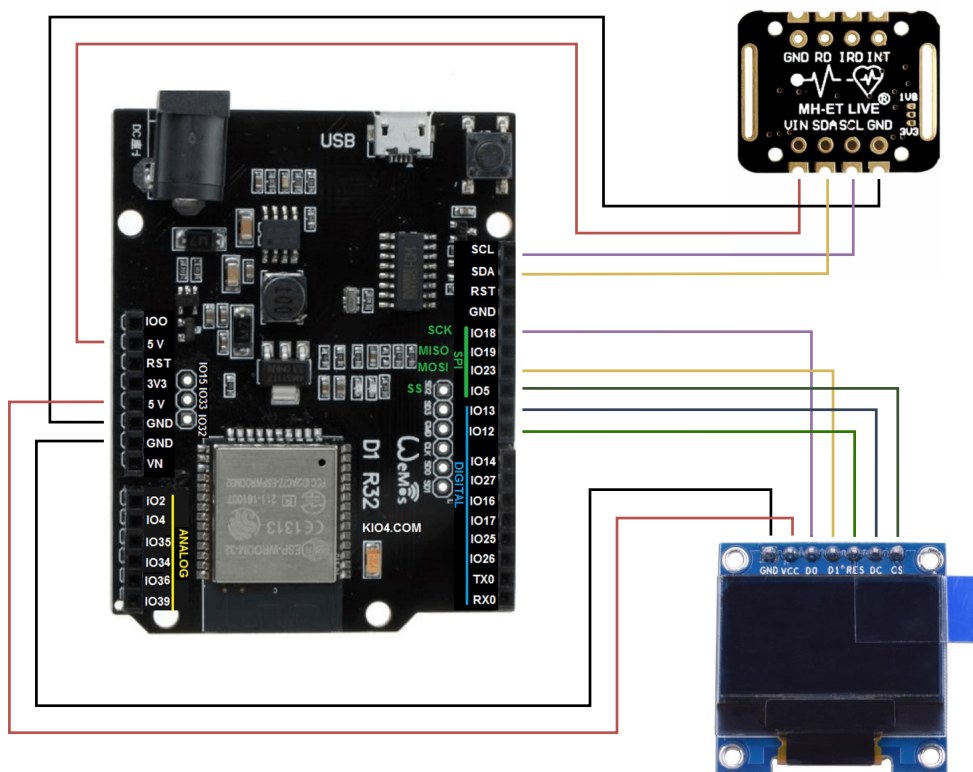
1 Popis

Tento projekt sa zaoberá zostavením merača srdcového tepu s jeho aktuálnym zobrazovaním na displeji. Použitie je po zapnutí systému priložiť prst na senzor a snažiť sa udržať silu tlaku na senzor na konštantnej úrovni. Systém rozpozná priloženie prsta a začne kalibráciu. Po prvej kalibrácii začne zobrazovať namerané hodnoty. Následne ešte chvíľu trvá kým sa hodnoty ustália. Po ich ustálení je možné sledovať aktuálny srdcový tep.

2 Použitý hardvér

Hlavným komponentom systému je doska Wemos D1 R32, založená na platforme ESP32. Na meranie tepu je použitý snímač srdcového tepu s pulzným oxymetrom MAX30102. Pre zobrazenie výstupov slúži dvojfarebný OLED displej SSD1306 vo variante so siedmimi pinmi a rozmermi 128x64 pixelov.

3 Schéma zapojenia



Obr. 1: Zdroje obrázkov: Wemos D1 R32 [3], MAX30102 Sensor [1], SSD1306 Display [2]

3.1 Rozhrania

Displej je zapojený cez rozhranie SPI. Z portu SCK prichádza hodinový signál, z portu MOSI vstup pre displej a z portu IO12 sínal na reset. Na porte IO13 je určené, či prebieha prenos dát alebo príkazov a na porte IO5 (SS) je nastavovaná hodnota slave select, no v tomto prípade ju nie je potrebné meniť, keďže je zapojené len jedno zariadenie v móde slave. Tepový senzor je zapojený cez rozhranie I2C, s hodinovým signálom prenášaným cez SCL a dátami cez SDA.

4 Implementácia programu

4.1 Vývojové prostredie

Na tvorbu programu bolo využité prostredie Visual Studio Code s rozšírením PlatformIO. Bol zvolený projekt pre zariadenie Espressif ESP32 Dev Module a framework Arduino.

4.2 Použité knižnice

Program využíva dostupné knižnice od spoločností Adafruit a SparkFun, pre jednoduchšiu prácu s používanými modulmi. Konkrétne knižnice:

- **Adafruit GFX Library** - hlavná knižnica pre prácu s grafikou a displejmi [5]
- **Adafruit SSD1306** - využívaná pre prácu s používaným displejom [6]
- **Adafruit BusIO** - vyžadovaná ako závislosť knižnice Adafruit GFX, používaná na komunikáciu [4]
- **SparkFun MAX3010x Pulse and Proximity Sensor Library** - využívaná pre prácu so snímačom srdcového tepu MAX30102 [7]

4.3 Makrá a globálne premenné

Prvá časť programu zahŕňa vytvorenie makier a globálnych premenných (Nie je potrebné aby všetky z týchto premenných boli globálne, no v čase písania dokumentácie som už nemal možnosť otestovať prípadné zmeny v kóde).

4.3.1 Displej

V tejto časti sú nastavené rozmery displeja a porty použité na jeho pripojenie. Ďalej je vytvorený objekt knižnice displeja, pomocou ktorého je možné s displejom pracovať. Nakoniec sú deklarované premenné na ukladanie výšky a šírky textu vykresľovaného na displej pre možnosť jeho centrovania.

4.3.2 Snímač tepu

V tejto časti je vytvorený objekt knižnice pre prácu so snímačom srdcového tepu, pole na ukladanie získaných hodnôt a ďalšie pomocné premenné.

4.4 Vykresľovanie na displej

V tejto časti sú vytvorené funkcie pre vykresľovanie na displej. Prvá funkcia zabezpečuje výpis do stredu displeja a ostatné zobrazenie konkrétnej “obrazovky”.

4.5 Počiatočné nastavenia

Nastavenie displeja, veľkosti písma a jeho farby pri rýchlosti 9600 bitov za sekundu. Nastavenie rýchlosti na 115200 bitov za sekundu a konfigurácia senzoru.

4.6 Hlavný cyklus

Na začiatku cyklu je načítaná hodnota odrazeného infračerveného svetla. Následne prebehne kontrola, či bol zaznamenaný úder srdca. O túto činnosť sa stará funkcia `checkForBeat` knižnice senzoru, ktorá kontroluje priebeh signálu a tak vyhodnocuje tvar krivky.

Ak bol úder zaznamenaný, zistí sa rozdiel v čase od posledného úderu a uloží sa do premennej `delta`. Čas posledného úderu sa aktualizuje. Následne je vypočítaná aktuálna hodnota úderov za minútu (BPM) výpočtom:

$$BPM = \frac{60}{delta[s]}$$

Vypočítaná hodnota BPM je uložená do poľa `rates` s veľkosťou `RATE_SIZE`, ktoré je dookola prepisované novými hodnotami. Po zápise hodnoty BPM je vypočítaná priemerná hodnota z celého poľa a tá je uložená ako priemerný tep v danom cykle do premennej `beatAvg`.

$$average = \frac{1}{size} \sum_{i=1}^{size} rates_i$$

Ak prebieha kalibrácia senzoru, prebehne kontrola, či už senzor nie je nakalibrovaný a následne sa rozhodne, ktorá obrazovka bude na displeji zobrazovaná.

Ak je hodnota odrazeného infračerveného svetla menšia ako 50000 program predpokladá, že prst nie je priložený a hodnoty priemerného tepu a kalibrácie sú resetované. Ak je prst priložený a prebieha kalibrácia, je zobrazovaná obrazovka kalibrácie a nakoniec, ak je kalibrácia ukončená, je zobrazovaná aktuálna priemerná hodnota úderov srdca za minútu.

5 Demonštračné video

Video je dostupné na tomto odkaze:

<https://drive.google.com/file/d/1XwMUblZ0-rVU92nXH-ZpRC0zMOWM8xtt/view?usp=sharing>

Literatúra

- [1] *MAX30102 Sensor Image*. Dostupné z: <https://www.phippselectronics.com/product/heart-rate-and-pulse-oximeter-sensor-module-max30102/>.
- [2] *SSD1306 Display Image*. Dostupné z: <https://robu.in/product/0-96-oled-display-module-sp112c-128x64-7-pin-blue/>.
- [3] *Wemos D1 R32 Image*. Dostupné z: <https://www.b0tnroll.com/en/esp/3639-wemos-d1-r32-w-esp32-un0-r3-pinout.html>.
- [4] ADAFRUIT. *Adafruit BusIO*. arduino.cc. Dostupné z: <https://www.arduino.cc/reference/en/libraries/adafruit-busio/>.
- [5] ADAFRUIT. *Adafruit GFX Library*. arduino.cc. Dostupné z: <https://www.arduino.cc/reference/en/libraries/adafruit-gfx-library/>.
- [6] ADAFRUIT. *Adafruit SSD1306*. arduino.cc. Dostupné z: <https://www.arduino.cc/reference/en/libraries/adafruit-ssd1306/>.
- [7] ELECTRONICS, S. *SparkFun MAX3010x Pulse and Proximity Sensor Library*. arduino.cc. Dostupné z: <https://www.arduino.cc/reference/en/libraries/sparkfun-max3010x-pulse-and-proximity-sensor-library/>.