

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická
Katedra elektroniky a informačních technologií

BAKALÁŘSKÁ PRÁCE

Měření tepové frekvence pomocí optického senzoru

Autor práce: **Tomáš Tolar**
Vedoucí práce: **Ing. Ondřej Vavroch**

2023

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Tomáš TOLAR**
Osobní číslo: **E19B0120P**
Studijní program: **B2612 Elektrotechnika a informatika**
Téma práce: **Měření tepové frekvence pomocí optického senzoru**
Zadávací katedra: **Katedra elektroniky a informačních technologií**

Zásady pro vypracování

1. Seznamte se s principem měření tepu srdce pomocí absorpce světla.
2. Vyberte vhodné komponenty, nakreslete schéma zapojení a realizujte prototyp zařízení.
3. Na prototypu proveďte měření a výsledky porovnejte s komerčními přístroji.
4. Zhodnoťte dosažené výsledky a navrhněte případné možnosti vylepšení.


Rozsah bakalářské práce: **30 – 40**
Rozsah grafických prací: **5**
Forma zpracování bakalářské práce: **elektronická**

Seznam doporučené literatury:

1. CHMELAŘ, M: ROZMAN, J.: Lékařská přístrojová technika. Praha:SNTL, s 182, 1982, ISBN 412-33326.

Vedoucí bakalářské práce: **Ing. Ondřej Vavroch**
Research and Innovation Centre for Electrical
Engineering

Datum zadání bakalářské práce: **7. října 2022**
Termín odevzdání bakalářské práce: **26. května 2023**



L.S.

Prof. Ing. Zdeněk Peroutka, Ph.D.
děkan



Doc. Ing. Jiří Hammerbauer, Ph.D.
vedoucí katedry

V Plzni dne 7. října 2022

Abstrakt

Bakalářská práce se zabývá měřením tepové frekvence pomocí optického senzoru. Cílem této práce je seznámení se s principem měření tepu pomocí absorpce světla. Začátek práce je věnován srdeční činnosti a metodám jejího měření. Mezi těmito metodami se blíže věnujeme právě metodě optické. Práce obsahuje také návrh, zhotovení obvodu a programu funkčního prototypu pracujícího na principu snímání absorpce světla. Toto zařízení je podrobena srovnávacímu testu s komerčně dostupnými zařízeními pro měření tepové frekvence. Na závěr jsou navržena možná vylepšení.

Klíčová slova

Tepová frekvence, měření tepové frekvence, kardiotachometr, fotoelektrická pletysmografie, optický snímač, STM32, OPT101

Abstract

The bachelor's thesis deals with the measurement of heart rate using an optical sensor. The aim of this work is to get acquainted with the principle of measuring pulse using light absorption. The beginning of the work is devoted to heart activity and methods of its measurement. Among these methods, we take a closer look at the optical method. The work also includes the design, construction of a circuit and a program of functional prototype working on the principle of light absorption. This device is subjected to a comparative test with commercially available cardiometers. At the end, possible improvements are suggested.

Key Words

Heart rate, heart rate measurement, cardiometer, photoelectric plethysmography, optical sensor, STM32, OPT101

Poděkování

Tímto bych rád poděkoval vedoucímu své bakalářské práce panu Ing. Ondřeji Vavrochovi za rychlou a kvalitní komunikaci a cenné odborné rady při vypracování bakalářské práce.

Obsah

Úvod.....	- 1 -
1 Základy srdeční činnosti.....	- 2 -
1.1 Srdce.....	- 2 -
1.2 Tep.....	- 3 -
1.2.1 Tepová frekvence.....	- 4 -
1.3 Význam tepu	- 4 -
2 Měření tepové frekvence	- 5 -
2.1 Nepřístrojové metody měření tepové frekvence	- 5 -
2.2 Přístrojové metody měření tepové frekvence.....	- 6 -
2.2.1 Proces přístrojového měření tepové.....	- 6 -
2.2.2 Přístrojové měření elektrických projevů srdeční činnosti.....	- 7 -
2.2.3 Přístrojové měření akustických projevů srdeční činnosti	- 8 -
2.2.4 Přístrojové měření mechanických projevů srdeční činnosti	- 8 -
2.3 Optické měření tepové frekvence.....	- 9 -
2.4 Přesnost a chyby měření tepové frekvence pomocí optického senzoru	- 11 -
2.4.1 Vliv pohybu	- 11 -
2.4.2 Vliv špatného prokrvení měřené tkáně	- 12 -
2.4.3 Vliv vnějšího světelného rušení	- 12 -
3 Návrh prototypu měřiče tepové frekvence	- 13 -
3.1 Blokové schéma	- 13 -
3.2 Senzor.....	- 14 -
3.2.1 Světelný zdroj	- 14 -
3.2.2 Fotodetektor	- 15 -
3.2.3 Integrovaný obvod OPT101.....	- 15 -
3.3 Úprava signálu.....	- 16 -
3.3.1 Filtry.....	- 16 -
3.3.2 Zesilovač	- 18 -
3.4 Zpracování signálu	- 18 -
3.5 Zobrazení dat.....	- 20 -

4	Realizace měřiče tepové frekvence	- 21 -
4.1	Schéma a deska plošného spoje finálního obvodu	- 21 -
4.2	Program	- 24 -
5	Testování	- 29 -
6	Možná vylepšení.....	- 32 -
	Zhodnocení a závěr	- 33 -
	Literatura.....	- 34 -
	Přílohy.....	I

Seznam symbolů a zkratek

Značka	Popisek	Jednotka
A_u	Zesílení zesilovače	[-]
C	Kapacita	[F]
f_m	Mezní frekvence	[Hz]
f_{max}	Maximální frekvence	[Hz]
f_{min}	Minimální frekvence	[Hz]
I_f	Proud diody v propustném směru	[A]
p_t	Počet tepů	[-]
R_x	Odpor	[Ω]
t_m	Čas jedné minuty	[s]
U_0	Napájecí napětí	[V]
U_d	Napětí diody v propustném směru	[V]
U_{in}	Vstupní napětí	[V]
U_{out}	Výstupní napětí	[V]
λ	Vlnová délka	[nm]
π	Ludolfovo číslo	[-]

Zkratka	Význam
<i>ADC</i>	Převodník analogového signálu na digitální (Analog to Digital Converter)
<i>BPM</i>	Tepy za minutu (Beats Per Minute)
<i>DP</i>	Dolní propust
<i>EKG</i>	Elektrokardiografie, Elektrokardiograf
<i>GND</i>	Zem (Ground)
<i>HP</i>	Horní propust
<i>IR</i>	Infračervené záření (Infrared)
<i>LED</i>	Světlo-emitující dioda (Light-emitting diode)
<i>OZ</i>	Operační zesilovač

Úvod

Srdce patří mezi nejdůležitější orgány lidského organismu. Tepová frekvence je jedním z nejlepších indikátorů správné srdeční funkce. Právě proto je měření tepové frekvence nezbytnou součástí medicíny, a to nejen v současné době, ale i v minulosti. Zároveň má měření tepové frekvence své opodstatnění i ve sporu a při jiných fyzicky či psychicky náročných aktivitách.

Samotné měření tepové frekvence může probíhat mnoha způsoby a pomocí různých nástrojů. Od jednoduché metody využívající změny tlaku krve, při které přiložíme prst na zápěstí nebo krk a počítáme počet úderů za určitý časový úsek. Přes využití sluchu za pomoci stetoskopu, až po snímání elektrokardiografem, který mimo tepu zaznamenává i jiné veličiny.

Další možnou metodou je měření tepové frekvence pomocí optického senzoru. Tato metoda se hojně používá pro svoji jednoduchost a snadnou integrovatelnost. Optické senzory tepové frekvence jsou dnes mimo jiné součástí mobilních telefonů, chytrých hodinek a sportovních náramků. Výše zmíněná zařízení jsou často využívána sportovci, kteří s jejich pomocí kontrolují a případně vylepšují efektivitu svých sportovních výkonů.

Ve své bakalářské práci se zabývám právě měřením tepové frekvence pomocí optického senzoru. Cílem této práce je blíže se seznámit s principem měření tepové frekvence pomocí absorpce světla. Součástí práce je také návrh a zhotovení prototypu přístroje, který bude schopen díky optickému senzoru měřit tepovou frekvenci. Začátek práce obsahuje teoretický základ anatomie srdce, srdeční činnosti a jejích projevů. Dále jsou představeny základní metody pro měření tepové frekvence. Se zaměřením na metodu měření tepu pomocí absorpce světla. Praktická část práce se skládá z návrhu a následné realizace měřiče tepové frekvence, který využívá právě metodu absorpce světla. V závěru je pak přístroj podroben srovnávacímu testu s komerčně dostupnými přístroji a jsou navržena možná zlepšení.

1 Základy srdeční činnosti

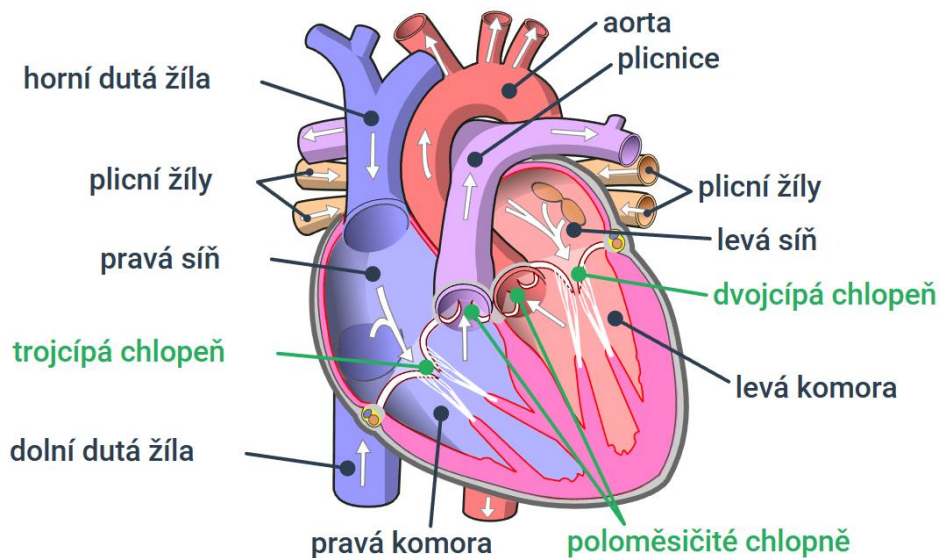
Chceme-li správně pochopit a měřit tepovou frekvenci, potřebujeme znát, co vlastně je tepová frekvence. Kde a proč vzniká a k čemu je nám znalost naší tepové frekvence prospěšná.

1.1 Srdce

Srdce je nepárový, nezávislý, samočinný a rytmický dutý svalovitý orgán, který je uložen v přední části hrudní dutiny. Srdce funguje jako pumpa, jejímž úkolem je kontinuálně přečerpávat krev v krevním oběhu a zásobovat tělo kyslíkem.

Srdce tvoří tři vrstvy. První vrstva se nazývá endokard. Jedná se o hladký epitel, který pokrývá vnitřní prostory srdce. Druhá je myokard. Myokard je srdeční svalovina tvořená buňkami spojenými pomocí spojek, které šíří signály pro stah. Třetí vrstvu tvoří perikard neboli osrdečník. Úkolem osrdečníku je povrchová ochrana. Skládá se ze dvou vazivových vrstev, mezi kterými je štěrbina naplněná kapalinou pro snížení tření.

Srdce je složeno ze dvou síní a dvou komor. Pravá síň, do které vstupuje neokysličená krev z horní a dolní duté žíly. Dále neokysličená krev pokračuje skrz trojcípou chlopeň do pravé komory. Z pravé komory je neokysličená krev vedena přes poloměsíčitou chlopeň a následně plicními tepnami do plic, kde dojde k okysličení krve. Okysličená krev je z plic vedena plicními žilami do levé síně. Z levé síně jde okysličená krev skrz dvoucípou chlopeň do levé komory. Levá komora díky silné vrstvě myokardu přečerpává okysličenou krev skrz poloměsíčitou chlopeň do aorty. Aorta neboli srdečnice je největší tepna lidského těla, která vede okysličenou krev dále do celého těla. Anatomie srdce je vidět na obrázku 1.



Obr. 1 Anatomie lidského srdce [11]

Během činnosti jednotlivých oddílů srdce dochází ke střídání systoly (stahu) a diastoly (ochabnutí). Součinnost síní i komor probíhá cyklicky ve třech fázích. První fáze systola síní a diastola komor. Druhá fáze diastola síní a systola komor. Během této fáze je slyšet v hrudníku zřetelný úder známý jako první srdeční ozva. Třetí fáze diastola síní a diastola komor. Lze slyšet slabší úder nazývaný druhá srdeční ozva.

Řízení srdeční činnosti má na starosti převodní systém srdeční. Základem tohoto systému jsou speciální přeměněná svalová vlákna, která dokážou vytvářet a přenášet signál. Strukturu převodního systému srdečního tvoří: sinoatriální uzel, internodiální síňové spoje, atrioventrikulární uzel, Hisův svazek, Tawarova raménka, Purkyňova vlákna.

1.2 Tep

Tep je jedním ze zevních projevů srdeční činnosti. Jedná se o tlakovou a objemovou vlnu šířící se oběhovou soustavou. Vlna vzniká při srdečním stahu (systole). Tep měřený na těle je konkrétně způsoben systolou levé komory. Při systole levé komory je ze srdce do aorty vypuzeno určité množství krve a vzniká tlaková vlna, která se dále šíří tepnami velkého tělního krevního oběhového systému. Rychlost šíření tlakové vlny je větší než rychlost proudění krve a je závislá na elasticitě tepen. Platí, čím menší elasticita tepen, tím vyšší rychlost šíření tlakové vlny.

1.2.1 Tepová frekvence

Tepová frekvence udává počet průběhů srdečního cyklu za určitý časový úsek. Nejčastěji se používá časový úsek jedné minuty. Lze tedy mluvit o počtu stahů (úderů) za minutu neboli anglicky: beats per minute (BPM).

Člověk má průměrně v klidu tepovou frekvenci přibližně 70 úderů za minutu. Průměrná hodnota se liší pro každého jedince v závislosti na několika faktorech. Mezi určující faktory se řadí věk, pohlaví, fyzická kondice a zdravotní stav. Tyto faktory ovlivňují tepovou frekvenci z dlouhodobého hlediska. Například u atletů se klidová tepová frekvence může pohybovat okolo 50 tepů za minutu. Naopak novorozenci mají průměrnou tepovou frekvenci znatelně vyšší a to přibližně 140 tepů za minutu.

Tepová frekvence však může být ovlivněna i krátkodobě, a to fyzickou nebo psychickou zátěží. U fyzické zátěže platí, čím intenzivnější fyzická zátěž, tím vyšší tepová frekvence. Tento jev je způsoben větší potřebou okysličené krve pro správnou funkci svalů. Jako příklad vlivu psychiky lze uvést vzestup tepové frekvence vlivem strachu.

1.3 Význam tepu

V lékařství může tepová frekvence podhalit základní informace o zdravotním stavu pacienta. Hlavně se pozoruje pravidelnost, frekvence a síla tepu.

V současné době je měření tepu také nedílnou součástí sportu. Znalost výše tepové frekvence při sportovních aktivitách pomáhá sportovcům optimalizovat zátěž a intenzitu tréninku a tím zlepšit jeho kvalitu.

2 Měření tepové frekvence

Existuje mnoho způsobů měření tepové frekvence. Obecně lze říci, že tepovou frekvenci měříme nebo odvozujeme z projevů, které doprovázejí srdeční činnost. Jsou to zejména projevy mechanické, akustické a elektrické.

Základní rozdělení metod měření tepové frekvence je rozdělení na metody přístrojové a nepřístrojové.

2.1 Nepřístrojové metody měření tepové frekvence

Nepřístrojové metody byly hojně užívány v historii. V současné době jsou použity v případech, kdy není k dispozici žádný měřicí přístroj. U nepřístrojové metody je pro detekci tepu použit zejména lidský hmat nebo sluch. V ojedinělých případech lze pulz kontrolovat i zrakem, daný způsob má však velké riziko chyby. Detekované tepy počítáme po určitý časový úsek a poté přepočteme na úderů za minutu.

Měření tepové frekvence pomocí hmatu spočívá v přiložení prstů na místa, kde je tep dobře hmatatelný. Tato místa nazýváme palpačně přístupné tepny. Jedná se hlavně o krční tepnu (krkavici) a radiální tepnu na zápěstí. Při lehkém přiložení prstů na palpační body lze nahmatat tep v podobě mechanického projevu srdeční činnosti, kdy při systole dojde ke zřetelnému zvýšení tlaku krve protékající tepnou. Jednotlivé tepy počítáme například po dobu třiceti sekund. Číslo, do kterého napočítáme, pak jednoduše zdvojnásobíme a dostaneme hodnotu naší tepové frekvence za minutu. Dobu, po kterou počítáme volíme dle situace. Většinou zde platí pravidlo čím kratší časový úsek pro měření, tím nižší přesnost výsledné hodnoty.

Měření tepové frekvence pomocí sluchu funguje podobně jako pomocí hmatu. Pouze místo přikládání prstů na palpační body, přiložíme ucho na hrud' a posloucháme akustický projev činnosti srdce. Konkrétně zřetelný úder srdce, který se nazývá první srdeční ozva. Opět počítáme počet úderů za daný časový úsek.

2.2 Přístrojové metody měření tepové frekvence

Jednotlivé metody přístrojového měření tepové frekvence jsou si velice podobné. Liší se jen typem projevu srdeční činnosti, který měříme a převádíme.

2.2.1 Proces přístrojového měření tepové

Obecně má proces přístrojového měření tepové několik kroků:

- Snímání tepu (biosignálu)
- zesílení a elektronická úprava signálu
- Zpracování signálu
- Zobrazení nebo zaznamenání naměřené hodnoty

Pro snímání využíváme snímače a měniče v závislosti na typu projevu srdeční činnosti. V případě snímání elektrického projevu srdeční činnosti použijeme elektrody. Pro akustický projev lze použít mikrofon. U mechanického projevu pak záleží na konkrétním typu sledovaného projevu. Například změna tlaku oběhové soustavy, změna objemu tkáně, změna rychlosti proudění krve, změna absorpce světla protékající krve a další.

Zesílení a elektronická úprava je zajištěna zejména pomocí zesilovačů, filtrů a převodníků. V současné době lze tento krok z části vynechat. Snímanou hodnotu přivedeme pomocí převodníku přímo do počítače. Zesílení a filtraci pak provádíme softwarově.

Ve fázi zpracování signálu dojde k přípravě signálu pro zobrazení nebo zaznamenání. Tento krok opět nemusí být vždy implementován. Například, je-li signál po zesílení a elektronické úpravě již ve stavu, ve kterém jsme schopni ho zobrazit.

Zobrazení a zaznamenání naměřené hodnoty se liší v závislosti na konkrétní aplikaci. Nejčastěji se však zobrazuje hodnota na displej, a to buď v podobě čísla nebo grafu. Naměřené hodnoty také mohou být ukládány na paměťové médium.

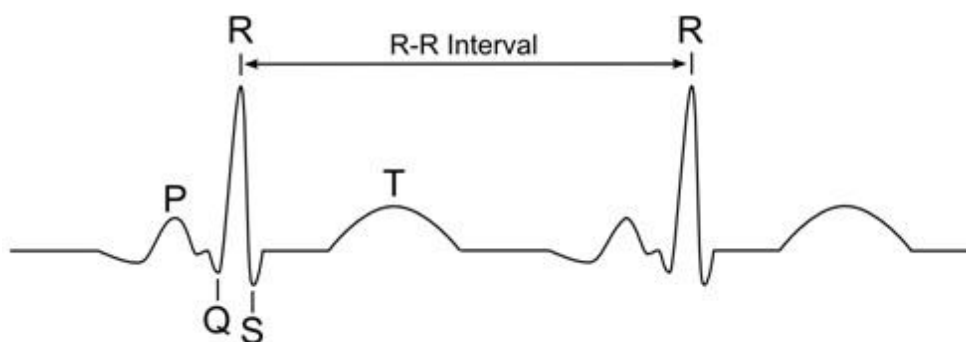
2.2.2 Přístrojové měření elektrických projevů srdeční činnosti

Elektrické projevy srdeční činnosti se v lékařství nejčastěji zaznamenávají za pomoci metody zvané elektrokardiografie. Užívá se zejména pro svoji přesnost a množství dat, které lze z jednoho měření získat. Přístroj používaný při elektrokardiografii nazýváme elektrokardiograf. Jako snímač jsou použity elektrody. Počet elektrod a jejich umístění na těle je standardizované, ne vždy je však pro přesné měření potřeba standard dodržet. Výstupem z elektrokardiografu je elektrokardiogram. Jedná se o časový záznam elektrokardiografických křivek. Z elektrokardiogramu lze spolehlivě vyčíst velké množství dat a identifikovat většinu srdečních vad a onemocnění. Mezi měřená data se řadí i tepová frekvence.

Elektrokardiografická křivka má typický tvar, který se skládá z vln, kmitů a linií. Tyto výchyly odpovídají specifickým fázím elektrického srdečního cyklu.

Křivka na obrázku 2 zobrazuje P vlnu, vznikající při depolarizaci síní. Dále máme PQ interval, při kterém dochází k převodu vzruchu ze síní na komory. Následuje QRS komplex, který je důsledkem depolarizace komor spolu s repolarizací síní. Poté přichází vlna T signalizující počátek repolarizace komor. V některých případech lze za vlnou T detekovat ještě jednu, znatelně menší vlnu U. Vlna U není konstantní a její původ není přesně stanoven.

Pro výpočet srdeční tepové frekvence z elektrokardiogramu stanovíme vzdálenost mezi dvěma periodicky se opakujícími ději. Nejčastěji se bere vzdálenost mezi kmity R (viz obrázek 2), ale není to pravidlo. Daná vzdálenost odpovídá určitému časovému úseku. Pro výpočet tepu pak stačí vydělit minutu tímto časovým úsekem. Tepovou frekvenci je také možno získat rovnou v průběhu vykreslování elektrokardiografické křivky, a to za pomoci detektoru R vlny.



Obr. 2 R-R interval na elektrokardiografické křivce [12]

2.2.3 Přístrojové měření akustických projevů srdeční činnosti

Akustické projevy srdeční činnosti nazýváme ozvy. Ozvy vznikají v důsledku otevírání a zavírání srdečních chlopní a díky změnám charakteru proudění krve. Rozlišujeme čtyři ozvy. První ozva je systolická, druhá ozva diastolická, třetí protodiastolická a čtvrtá presystolická. Mimo ozev lze zaznamenat i jiné zvuky doprovázející srdeční činnost. Akustický signál můžeme zaznamenat pomocí fonokardiografu. Pro snímání zvuku je zde použit jeden nebo více mikrofonů, které jsou umístěny na hrud'

Vlastní tepovou frekvenci potom lze zjistit díky periodicitě signálu podobně jako u elektrokardiografu. Jako záchytný bod se užívá první srdeční ozva. Nevýhodou této metody je náročnost zpracování získaného signálu. Vyskytuje se zde značný šum.

Akustická metoda se často využívá v těhotenství pro kontrolu srdeční činnosti plodu. Je při tom kladen velký důraz na odfiltrování rušení v podobě tepu matky.

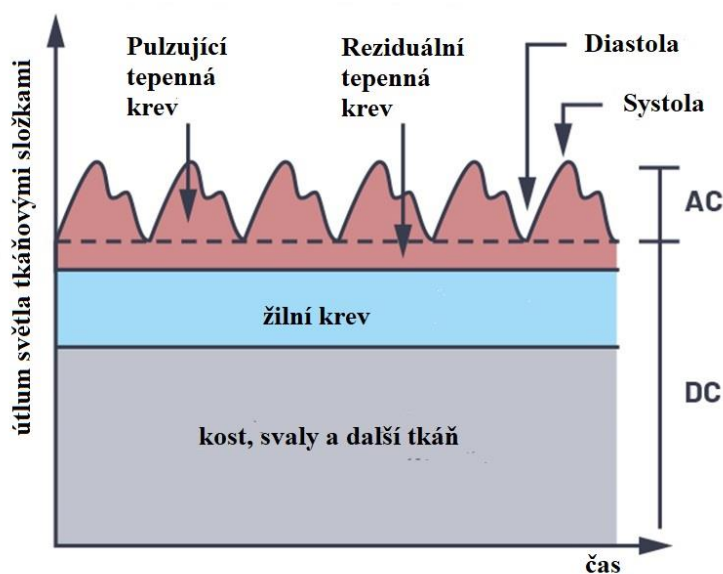
2.2.4 Přístrojové měření mechanických projevů srdeční činnosti

Existuje více metod pro měření mechanických projevů srdeční činnosti. Jednou z těchto metod je pletysmografie.

Pletysmografie je neinvazivní metoda měření periodicky se opakujících objemových změn lidského těla. Jedná se zejména o objemové změny způsobené dýcháním nebo prokrvením pozorované části těla. Měřicí přístroj používaný při pletysmografii lze obecně nazvat pletysmograf. Výstupem z pletysmografu je pletysmografická křivka. Hlavním účelem pletysmografie je kontrola kvality prokrvení tkání, ale umožňuje i měření tepové frekvence. Rozlišujeme více typů pletysmografů, a to v závislosti na způsobu měření objemových změn. Mezi tyto typy řadíme pletysmografy pneumatické, tenzometrické, impedanční, elektrokapacitní a fotoelektrické.

2.3 Optické měření tepové frekvence

Měření tepové frekvence pomocí optického senzoru je založeno na principu fotoelektrické pletysmografie. Fotoelektrická pletysmografie se zabývá změnou světelné propustnosti tkáně, v závislosti na změně krevního objemu při srdeční činnosti. Samotná změna světelné propustnosti tkáně je způsobena změnou tlaku krve při srdeční systole. Tlak krve má vliv na objem kapilár, kterými krev protéká. V důsledku změny objemu kapilár se mění světelná absorpce, odrazivost a světelný rozptyl tkáně. Grafické znázornění absorpce světla tkání je na obrázku 3.

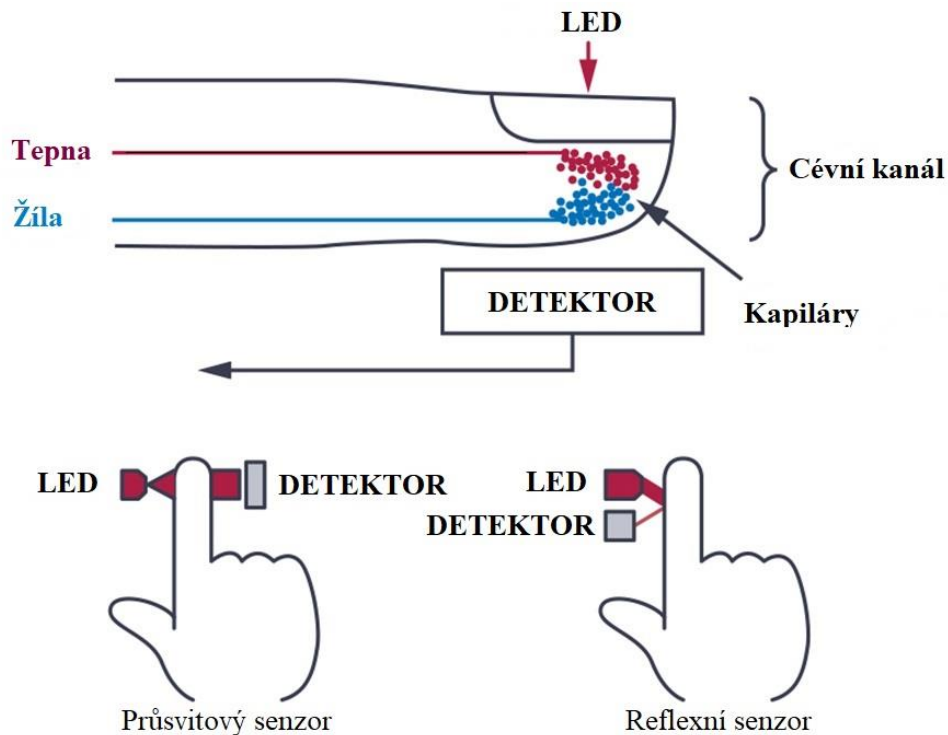


Obr. 3 Faktory ovlivňující velikost světelné absorpce [7]

Fotoelektrický pletysmograf je přístroj, pomocí kterého můžeme měřit nejen tepovou frekvenci, ale i okysličení krve. Fotoelektrický pletysmograf se skládá z několika základních bloků. Tyto bloky jsou: světelný zdroj, světelný přijímač (fotodetektor), obvod pro úpravu a zpracování signálu a zobrazovací obvod.

- Světelný zdroj může být IR dioda nebo LED dioda. Pro měření samotné tepové frekvence stačí LED dioda. IR dioda se používá v aplikacích, kde se mimo tepu měří i okysličení krve.
- Jako fotodetektor lze použít fotodiodu či fototranzistor. U fotodetektoru je kladen důraz na jeho rychlost a citlivost
- Obvod pro úpravu a zpracování signálu obsahuje zesilovače, filtry a převodníky. V některých případech může v závislosti na složitosti zařízení obsahovat i mikropočítač.
- Zobrazovací obvod tvoří obrazovka.

Fotoelektrické pletysmografy lze rozdělit dle použitého senzoru na reflexní a průsvitové. Princip senzoru fotoelektrického pletysmografu je vyobrazen na obrázku 4.



Obr. 4 Princip fotoelektrické prstové pletysmografie [7]

- U reflexního senzoru je světelný zdroj a detektor umístěn v jedné rovině. Detektor zachytává světelný signál odražený od tkáně. Reflexní senzor je použit například u chytrých hodinek, sportovních náramků nebo hrudních pásů.
- Průsvitový senzor má světelný zdroj umístěn naproti detektoru. Detektor zachytává signál procházející měřenou tkání. Senzory lze umístit pouze na určitá místa (prsty a ušní lalůčky).

Fotoelektrické pletysmografy jsou nedílnou součástí nejen většiny nemocničních zařízení, ale můžeme je najít i v domácnostech v podobě pulzních oxymetrů nebo výše zmíněných chytrých zařízení. Jsou oblíbeny pro svoji nenáročnou obsluhu a velikost. Nevýhodou těchto zařízení je citlivost na rušení a tím způsobená nepřesnost. Z tohoto důvodu jsou zařízení vhodná spíše pro orientační a srovnávací měření.

2.4 Přesnost a chyby měření tepové frekvence pomocí optického senzoru

Měření tepové frekvence optickým senzorem není dokonalé. Může zde docházet k nepřesnostem způsobeným vícero faktory. Mezi které patří například pohyb, vnější světelné rušení a špatné prokrvení měřeného místa. Vliv na měření může mít ale i pigmentace kůže nebo v případě měření na prstu lak na nehty. Většinu z těchto nežádoucích vlivů můžeme částečně redukovat použitím různých filtrů a dodržováním správnosti postupu měření. Ovšem málo kdy jde rušení eliminovat úplně. Z tohoto důvodu jsou optické měřiče tepové frekvence používány spíše v aplikacích, kde nejsou vysoké nároky na přesnost.

V lékařství se optické senzory pro měření tepové frekvence užívají například na jednotce intenzivní péče. Zde jsou dlouhodobě připojeny k pacientovi. Jejich úkolem je spustit varovný signál v případě, že se tepová frekvence dostane mimo stanovený rozsah.

2.4.1 Vliv pohybu

Chyby způsobené pohybem mohou vznikat při pohybu senzoru nebo měřené osoby. Jedná se zde nejen o pohyb měřeného místa, ale například i o pohyb způsobený dýcháním, otočením hlavy nebo poklepávání nohou z důvodu nervozity. Důsledkem je zkreslení křivky a negativní ovlivnění až znehodnocení měřených dat.

Velikost a četnost chyb způsobených pohybem lze redukovat pečlivým upevněním senzoru a měřeného místa. Dále můžeme instruovat měřenou osobu, aby vykonávala co možná nejméně nežádoucích pohybů. Dalším způsobem řešení chybovosti jsou různé filtry. Filtry mohou například průměrovat naměřené hodnoty, a tím snížit vliv jednorázových výkyvů. Nebo odfiltrovat hodnoty, které logicky neodpovídají tepu.

V případě měření tepové frekvence v průběhu fyzické aktivity, je většinou problém s pohybem řešen pomocí akcelerometru, který je součástí měřícího zařízení. Data z akcelerometru jsou porovnávána s daty z měřiče pulzu. Výkyvy křivky způsobené pohybem jsou poté odfiltrovány.

2.4.2 Vliv špatného prokrvení měřené tkáně

Špatné prokrvení měřené tkáně může být způsobeno přílišným utažením senzoru, ale i zdravotním stavem měřené osoby.

Přílišné utažení senzoru zapříčiní špatné prokrvení. Tím následně dochází k postupnému zmenšování křivky indikující srdeční činnost. Dojde-li k úplnému odkrvení, není možné tep na daném místě zachytit vůbec. V tomto případě lze prokrvení zajistit povolením senzoru a zabezpečením správné cirkulace krve.

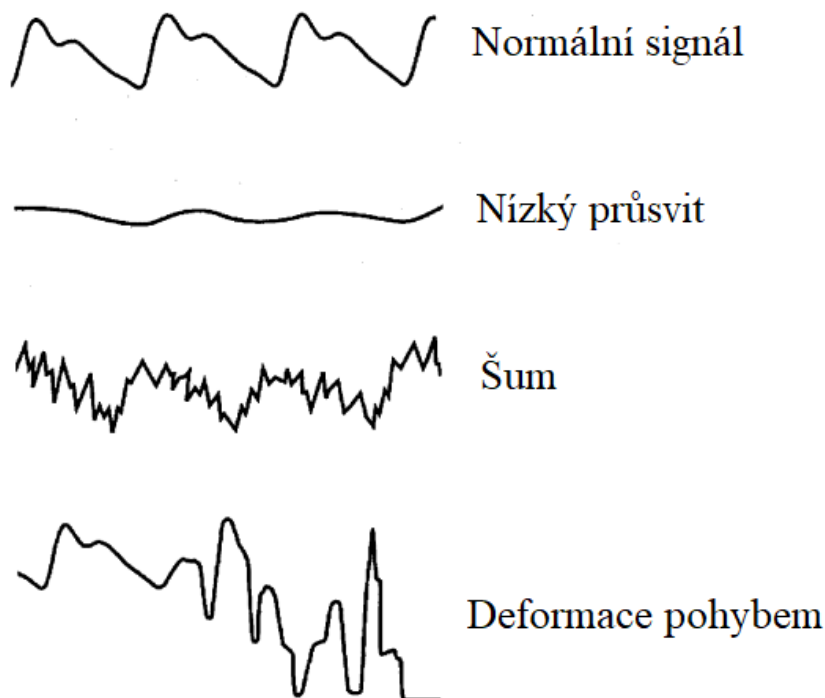
Je-li špatné prokrvení způsobeno zdravotním stavem měřené osoby, například nízkým krevním tlakem nebo podchlazením, je měřený signál příliš slabý pro správné fungování měřicího přístroje.

Problém však může vzniknout nejen při nedostatečném prokrvení daného místa, ale i v případě překrvení. V tomto případě může docházet ke vzniku falešných pulzů. A tím opět ke zkreslení měřených dat.

2.4.3 Vliv vnějšího světelného rušení

Vnější světelné rušení může mít významný vliv na naměřená data. I tento problém však lze alespoň částečně vyřešit. Řešením je dostatečná světelná izolace senzoru od okolí.

Obrázek 5 znázorňuje signál deformovaný různými vlivy.



Obr. 5 Vlivy chyb na průběh pletysmografické křivky [13]

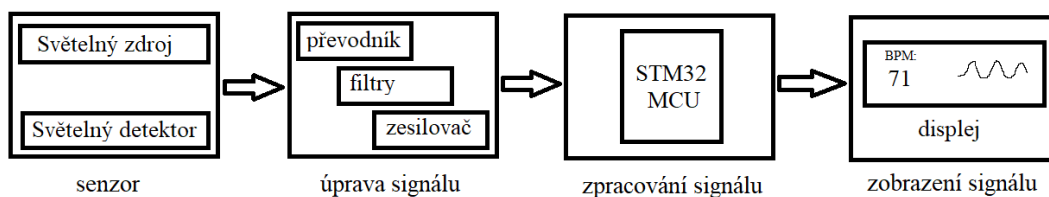
3 Návrh prototypu měřiče tepové frekvence

Následující kapitola obsahuje informace o návrhu prototypu zařízení, pomocí kterého lze měřit tepovou frekvenci. Navrhovaný prototyp užívá pro měření tepové frekvence metodu fotoelektrické pletysmografie, která byla popsána v kapitole 2.3.

3.1 Blokové schéma

Navrhovaný prototyp je tvořen čtyřmi funkčními bloky. Blokové schéma je vyobrazeno na obrázku 6. Jednotlivé bloky jsou blíže popsány v následujícím textu.

Prvním blokem je snímač skládající se ze světelného zdroje a fotodetektoru. Následuje blok úpravy signálu. Tento blok obsahuje převodník, filtry a zesilovač. Třetí blok má na starosti zpracování signálu za pomoci mikrokontroleru. Posledním blokem je blok zobrazovací. Jedná se o displej, na kterém se zobrazuje hodnota měřené tepové frekvence.



Obr. 6 Blokové schéma přístroje na měření tepové frekvence

3.2 Senzor

Jako senzor je zvolen průsvitový typ fotoelektrického senzoru. U tohoto typu senzoru je světelný zdroj umístěn proti fotodetektoru. Průsvitový fotoelektrický senzor je ideální pro měření tehu z prstu a vyznačuje se větší přesností než senzor reflexní.

3.2.1 Světelný zdroj

Jako světelný zdroj lze použít jeden ze tří typů diod: infračervená LED dioda, červená LED dioda a zelená LED dioda. Tyto diody mají odlišnou vlnovou délku. Infračervené LED diody mají vlnovou délku nad 760nm, červené LED diody mají vlnovou délku v rozmezí 610nm až 760nm a zelené LED diody pracují s vlnovou délkou mezi 500nm a 570nm. Vlnová délka má vliv na hloubku a kvalitu průniku světelného záření skrz tkáň. Platí zde čím větší vlnová délka, tím lepší průnik tkání. Současně ovšem se stoupající vlnovou délkou stoupá i citlivost na pohyb.

Navrhovaný prototyp využívá červenou LED diodu společně s infračervenou LED diodou. Toto řešení umožní v budoucnu prototyp obohatit o funkci měření saturace krve.

Samotný obvod je napájen pěti volty a skládá se ze dvou paralelních větví. První větev obsahuje sériově zapojený rezistor R1 a trimer R11, umožňující nastavení proudu diodou. Dále je v první větvi zapojena červená LED dioda 530FR4C v propustném směru a unipolární tranzistor BSS138. Druhá větev se skládá z rezistoru R2, trimru R12, infračervené LED diody 510E850C a druhého unipolárního tranzistoru BSS138. Unipolární tranzistory umožňují přepínat mezi jednotlivými větvemi, respektive LED diodami.

Hodnoty R1, R2, R11, R12 jsou závislé na používaných diodách. Červená LED dioda 530FR4C má napětí v propustném směru $U_d = 1,95V$ a proud v propustném směru $I_f = 20mA$. Pro výpočet odporu je použita rovnice (1).

$$R = \frac{U_0 - U_d}{I_f}, \quad (1)$$

Kde R je výsledný odpor [Ω], U_0 je napájecí napětí [V], U_d je napětí diody v propustném směru [V], I_f je proud diodou v propustném směru [A].

Výsledný odpor pro červenou LED diodu je roven 152Ω . Odpor se skládá z rezistoru R1 s odporem 66Ω a trimru R11 s nastaveným odporem 86Ω .

U infračervené LED diody 510E850C s $U_d = 1,5V$ a $I_f = 50mA$. Má výsledný odpor hodnotu 70Ω . Tato hodnota je složena z $R2 = 33\Omega$ a $R12 = 37\Omega$.

Schéma obvodu světelného zdroje včetně hodnot součástí zobrazuje obrázek 12 v kapitole 4.1.

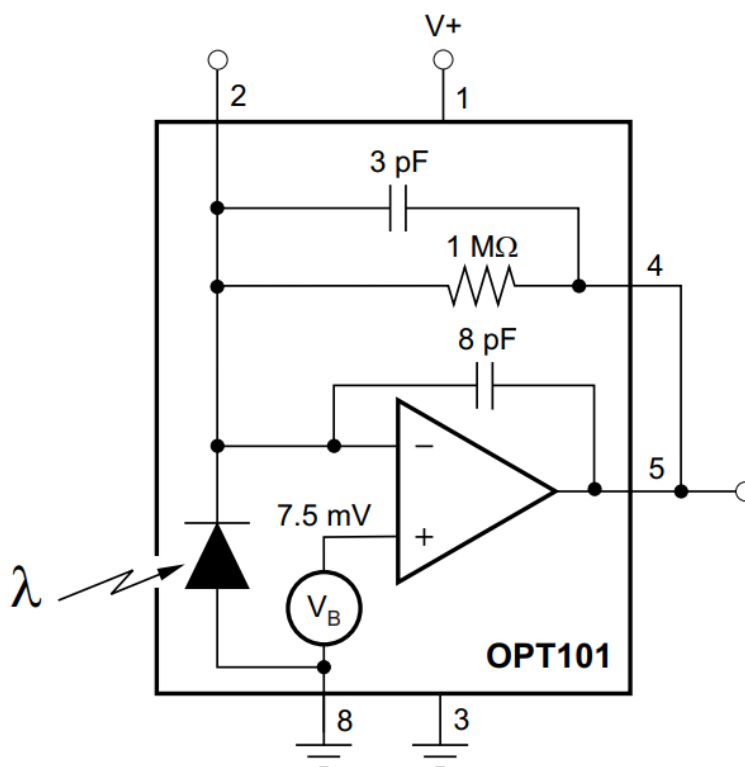
3.2.2 Fotodetektor

U fotodetektoru je na výběr mezi fotodiodou, fototranzistorem a fotorezistorem. Fotorezistor lze z výběru vyřadit z důvodu jeho vysoké tepelné závislosti. Tepelná závislost by se mohla negativně projevit v případě dlouhodobého přiložení prstu na senzor. Fotodioda a fototranzistor jsou velice podobné polovodičové součástky. Obě součástky je možné použít jako světelný detektor pro detekci tepové frekvence. Relevantními rozdíly pro danou aplikaci jsou citlivost a rychlost odezvy. Fototranzistor je obecně citlivější než fotodioda, naopak rychlost odezvy je vyšší u fotodiody.

3.2.3 Integrovaný obvod OPT101

Pro prototyp je jako fotodetektor zvolen integrovaný obvod OPT101. Hlavním důvodem použití OPT101 je úspora místa na desce plošného spoje a z toho vyplývající zmenšení velikosti celého prototypu.

Součástí OPT101 je fotodioda společně s převodníkem proud/napětí oboje integrované v jednom pouzdře. Převodník proud/napětí převede detekovaný signál do formy vhodné pro další zpracování. Vnitřní schéma OPT101 zobrazuje obrázek 7.



Obr. 7 Schéma vnitřního zapojení integrovaného obvodu OPT101 [10]

Integrovaný obvod OPT101 pracuje s napájením v rozsahu 2,7V – 36V. Prototyp je napájen 5V. Zpětnovazební rezistor 1MΩ obsažený v obvodu je příliš velký a výstupní signál proto neodpovídá předpokladům z důvodu saturace OZ. Problém lze vyřešit připojením rezistoru s odporem 1MΩ mezi piny 2 a 4 respektive 5. Výsledkem připojení rezistoru je snížení zpětnovazebního odporu na 0,5MΩ. Snížením odporu klesne zesílení a na výstupu je již zřetelný požadovaný tvar signálu.

3.3 Úprava signálu

Díky použití integrovaného obvodu OPT101 není potřeba signál z fotodetektoru převádět. Je však třeba signál filtrovat a zesílit pro zjednodušení následujícího zpracování.

3.3.1 Filtry

Tepová frekvence zdravého člověka v klidu i při zátěži se pohybuje v rozmezí 40-220 tepů za minutu. Přibližně v tomto rozmezí měří i prototyp. Spodní hranice je nastavena na 36 tepů, to odpovídá dle rovnice (2) frekvenci 0,6 Hz. Horní hranice je 245 tepů, což dle rovnice (2) odpovídá frekvenci 4,08 Hz. Frekvence, které se nenachází ve stanoveném rozsahu jsou frekvence parazitní a je možné je odfiltrovat pomocí filtrů.

$$f_{min/max} = \frac{1}{t_m \cdot p_t}, \quad (2)$$

Kde $f_{min/max}$ je minimální, respektive maximální frekvence pro daný počet tepů [Hz], t_m je časový úsek [s] (konkrétně 60s), p_t je počet tepů [-] (36 pro minimum a 245 pro maximum).

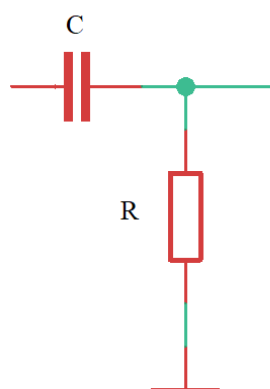
Prototyp obsahuje dva filtry. První filtr je pasivní filtr prvního řádu typu horní propust, který má na starosti odfiltrovat stejnosměrnou složku a frekvence nižší, než je stanovená spodní hranice. Jedná se o CR filtr, jehož schéma ukazuje obrázek 8. Pro výpočet hodnot prvků v obvodu slouží rovnice (3). Nejdříve je volen jeden z prvků a poté dopočítána hodnota druhého tak, aby mezní frekvence odpovídala dolní hranici rozsahu (konkrétně 0,6Hz). Pro prototyp je zvolen rezistor 120kΩ, čemuž po dosazení do upravené rovnice (4) odpovídá kondenzátor 2,2μF.

$$f_m = \frac{1}{2 \cdot \pi \cdot R \cdot C}, \quad (3)$$

Z této rovnice lze odvodit vztah pro výpočet hodnoty kondenzátoru.

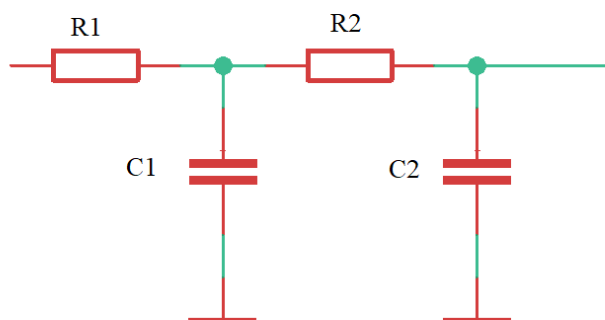
$$C = \frac{1}{2 \cdot \pi \cdot R \cdot f_m}, \quad (4)$$

Kde C je hodnota kondenzátoru (kapacita) [F], R je hodnota rezistoru (odporu) [Ω] a f_m je mezní frekvence [Hz].



Obr. 8 Schéma pasivního filtru horní propust

Druhý filtr je pasivní filtr druhého řádu typu dolní propust. Jde vlastně o dva pasivní filtry typu dolní propust zapojené do série. Oba filtry filtrují šum o frekvenci vyšší, než je stanovená horní hranice. Schéma filtrů je vidět na obrázku 9. Stanovení hodnot jednotlivých prvků je obdobné výše zmíněnému filtru HP. Zvoleny jsou rezistory $R_1 = 10\text{k}\Omega$ a $R_2 = 3,9\text{k}\Omega$. Mezní frekvence pro první filtr je 7,23 Hz. Tento filtr slouží k hrubému odfiltrování. Druhý filtr má mezní frekvenci 4,08 Hz. Pro výpočet hodnot kondenzátorů je použita rovnice (4). Po dosazení do rovnice (4) vyjde hodnota pro $C_1 = 2,2\mu\text{F}$ a pro $C_2 = 10\mu\text{F}$.



Obr. 9 Schéma sériově zapojených pasivních filtrů typu dolní propust

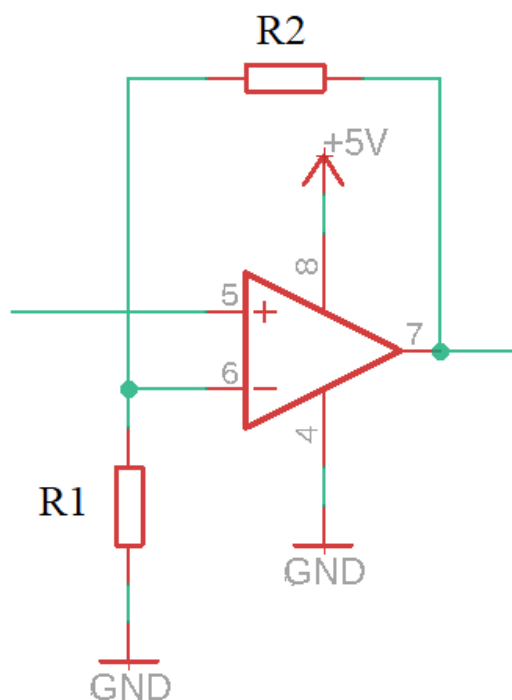
3.3.2 Zesilovač

Zesílení signálu je zabezpečeno neinvertujícím zapojením operačního zesilovače. Schéma zapojení je na obrázku 10.

Jedná se o operační zesilovač, na jehož neinvertující vstup je přiveden měřený signál. Invertující vstup je přes rezistor $R1 = 10\text{k}\Omega$ uzemněn. Zpětnou vazbu mezi výstupem OZ a invertujícím vstupem tvoří rezistor $R2$. Pomocí rezistorů je nastaveno zesílení neinvertujícího operačního zesilovače dle rovnice (5). Snahou je dostatečné zesílení signálu pro využití celého rozsahu ADC.

$$A_u = \frac{U_{out}}{U_{in}} = \left(1 + \frac{R2}{R1}\right), \quad (5)$$

Kde A_u je zesílení [-], U_{out} je výstupní napětí [V], U_{in} je vstupní napětí [V], $R1$ je zemnicí odpor [Ω] a $R2$ je zpětnovazební odpor [Ω].



Obr. 10 Schéma neinvertujícího operačního zesilovače

3.4 Zpracování signálu

Zpracování dat je zabezpečeno mikrokontrolerem. Mikrokontroler převádí vstupní analogový signál do číslicové podoby, provádí výpočet tepové frekvence, připravuje a následně expeduje data pro zobrazení. Zároveň mikrokontroler řídí chování světelného zdroje.

U prototypu je pro zpracování signálu použita vývojová deska Nucleo-32, která obsahuje mikrokontroler STM32G031K8. Tato deska je zvolena z důvodu splnění požadavků aplikace. Mezi požadavky se řadí: Dostupnost, rozměry, analogově digitální převodník (ADC), výstupy pro řízení světelného zdroje, vstupy a výstupy pro komunikaci s displejem, napájení přes USB a následný rozvod 5V nebo 3,3V pro napájení ostatních bloků, paměť FLASH (64KB) a paměť SRAM (8KB), vnitřní hodiny s frekvencí alespoň 16MHz, nástroj pro programování a debugging ST-LINK/V2-1.

Na obrázku 11 je vidět rozložení pinů Nucleo desky.

NUCLEO-G031K8					
PB6	1	D1	VIN	1	VIN
PB7	2	D0	GND	2	GND
NRST	3	NRST	NRST	3	NRST
GND	4	GND	+5V	4	+5V
PA15	5	D2	A7	5	PA7
PB1	6	D3	A6	6	PA6
PA10	7	D4	A5	7	PA11
PA9	8	D5	A4	8	PA12
PB0	9	D6	A3	9	PA5
PB2	10	D7	A2	10	PA4
PB8	11	D8	A1	11	PA1
PA8	12	D9	A0	12	PA0
PB9	13	D10	AREF	13	AREF
PB5	14	D11	+3V3	14	+3V3
PB4	15	D12	D13	15	PB3
	CN3		CN4		

Obr. 11 Pinout desky Nucleo STM32-G031K8 [9]

Používané piny a jejich účel:

- GND – zemnicí piny
- +5V – napájení světelného zdroje, OPT101 a operačního zesilovače
- +3V3 – napájení displeje
- D3, D6 – výstupní piny sloužící k ovládní LED a IR diody (D3-LED, D6-IR)
- D4 – Pin datového kanálu (SDA) sériové sběrnice I²C pro displej
- D5 – Pin hodinového signálu (SCL) sériové sběrnice I²C pro displej
- A4 – vstupní pin s ADC sloužící jako vstup pro naměřený signál.

3.5 Zobrazení dat

Zobrazení dat lze dosáhnout pomocí displeje. Zde je velké množství možností. Při výběru vhodného displeje záleží na více faktorech, jejichž důležitost se mění v závislosti na aplikaci a zobrazovaných datech.

U měřičů tepové frekvence je pro dosažení požadované funkčnosti nutné zobrazit číselnou hodnotu tepové frekvence. Toho lze docílit použitím jednoduchých segmentových displejů nebo alfanumerických LCD displejů. Mimo číselné hodnoty lze u měření tepové frekvence zobrazit například i průběh srdeční činnosti v podobě grafu. V tomto případě je potřeba grafický displej.

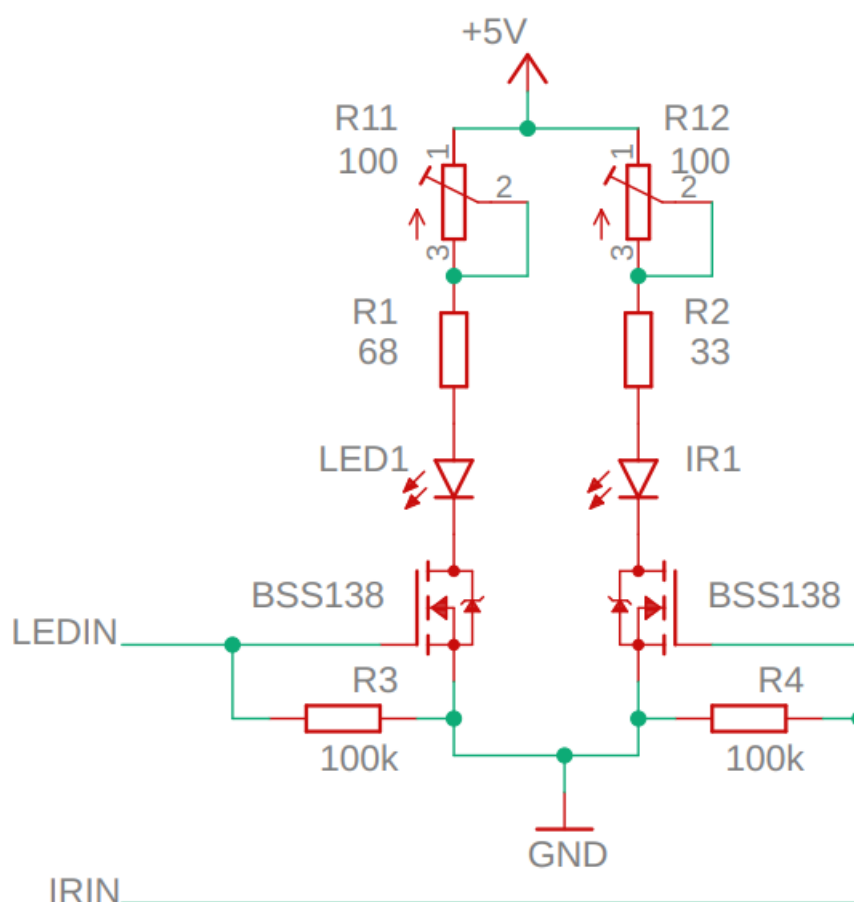
U prototypu je pro zobrazení použit grafický OLED displej s rozlišením 128 x 64 pixelů. Displej umožňuje zobrazení číselné hodnoty tepu i vykreslení grafického průběhu srdeční činnosti. Displej komunikuje s mikrokontrolerem přes rozhraní I²C pomocí řídicího obvodu SSD1306 a je napájen 3,3V přímo z desky Nucleo-32.

4 Realizace měřiče tepové frekvence

V první části této kapitoly je popsána deska plošného spoje a schéma celého obvodu prototypu. Druhá část kapitoly se věnuje programovému řešení této práce.

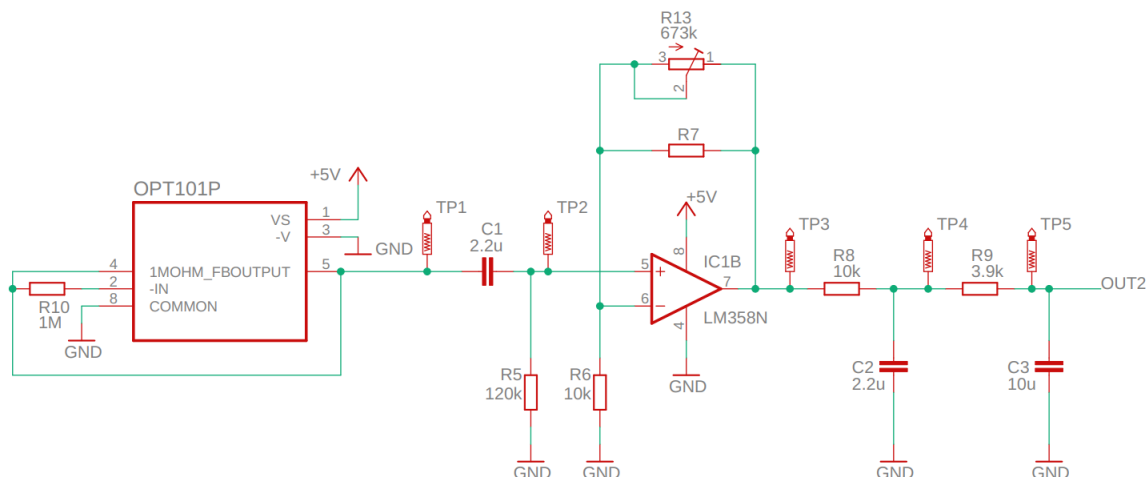
4.1 Schéma a deska plošného spoje finálního obvodu

Finální schéma prototypu měřiče tepové frekvence je pro lepší čitelnost rozděleno na tři části. První část je světelný zdroj. Schéma světelného zdroje včetně hodnot je na obrázku 12. Na trimeru R11 je nastaven odpor 86 Ω a na trimeru R12 je odpor 42 Ω .



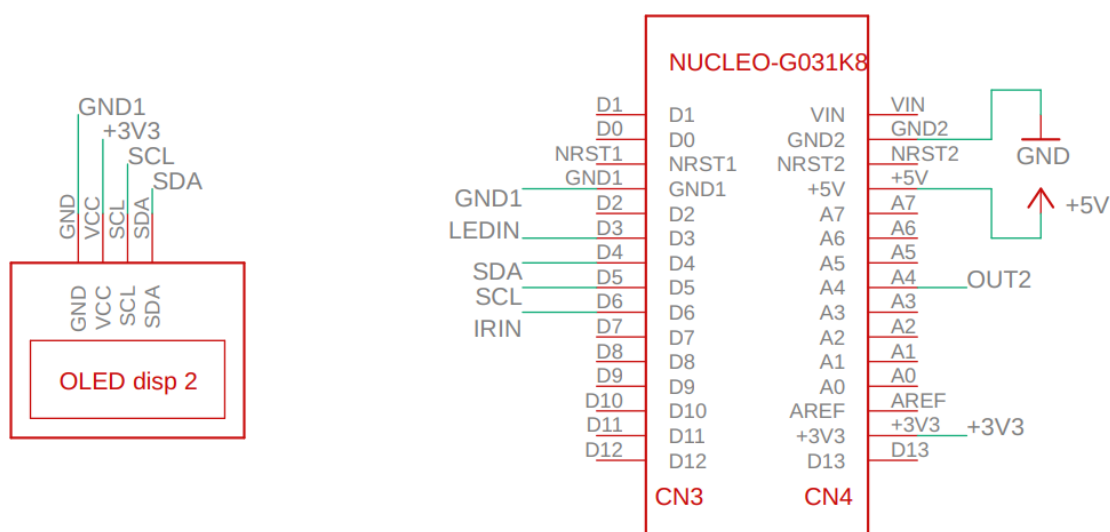
Obr. 12 Schéma světelného zdroje včetně hodnot

Druhá část obsahuje schéma obvodu detektoru společně s obvodem pro úpravu signálu. Schéma lze vidět na obrázku 13. Obvod pro úpravu signálu se skládá z filtru horní propust, neinvertujícího operačního zesilovače se zesilením 68,3, který má ve zpětné vazbě trimer R13 nastavený na 673 k Ω . Odpor R7 není osazen. Obvod uzavírají dva v sérii zapojené filtry typu dolní propust. Výstup z filtrů je připojen na ADC vstup mikrokontroleru. Ve schématu jsou i testovací body, které slouží k případnému odladění filtrů a zesílení.



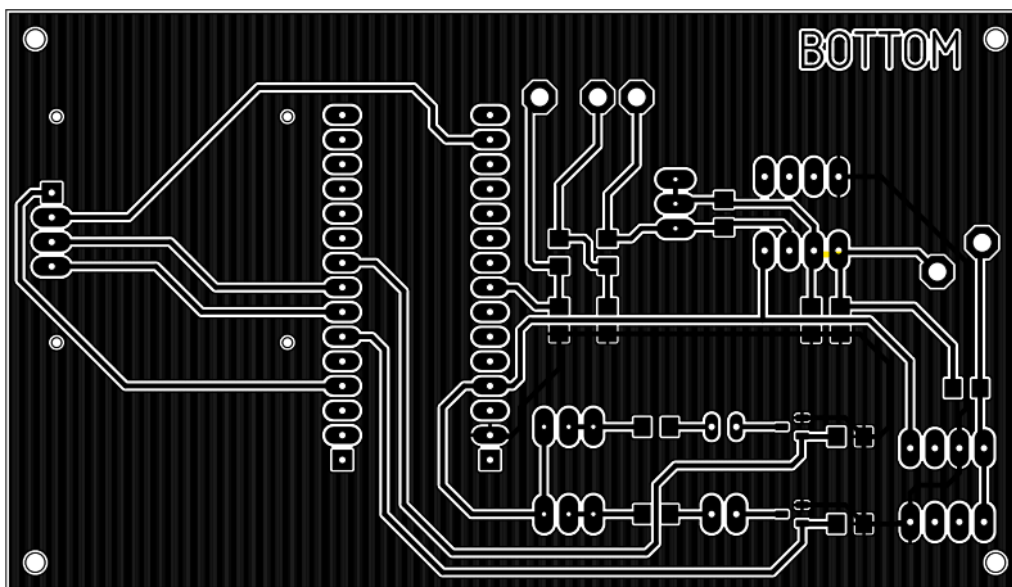
Obr. 13 Schéma detektoru a obvodu pro úpravu signálu

Třetí část zobrazuje obrázek 14. Sestává se z mikrokontroleru a displeje. Z důvodu lepší přehlednosti je propojení jednotlivých vodičů naznačeno pomocí popisek.



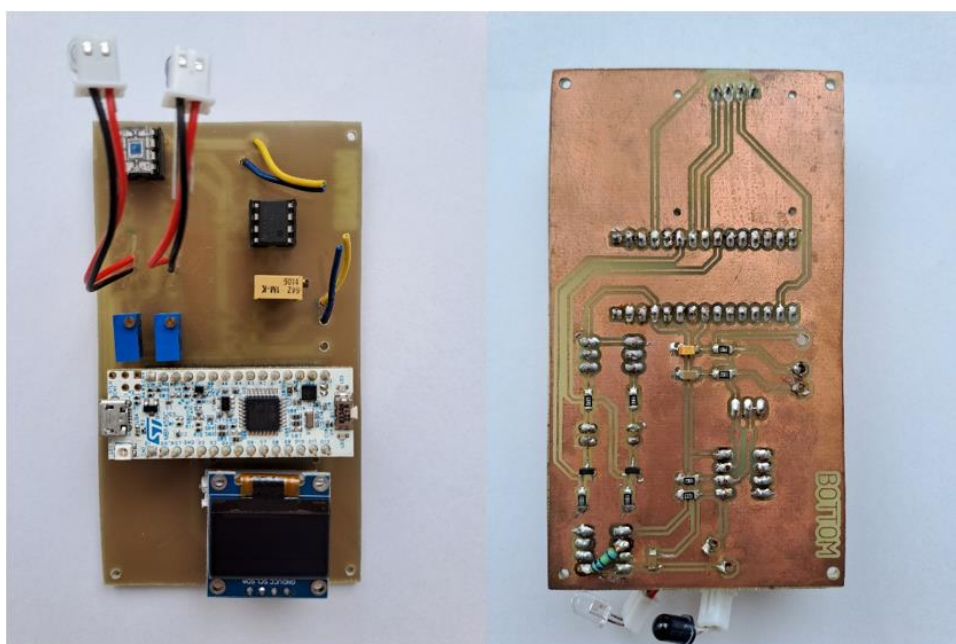
Obr. 14 Schéma propojení mikrokontroleru s displejem

Celý obvod je umístěn na jednu desku plošného spoje o rozměrech 105x60 mm. Připojení mikrokontroleru, displeje, operačního zesilovače a OPT101 je zajištěno pomocí patic a dutinových lišt. Patice společně s dutinovými lištami a THT trimery jsou připájeny z horní strany desky plošného spoje. Na spodní straně desky plošného spoje jsou dle obrázku 15 vytvořeny, pomocí osvitové jednotky, vodivé cesty s rozlitou zemí.



Obr. 15 Šablona desky plošného spoje

Spodní strana je osazena SMD součástkami. Obrázek 16 vyobrazuje obě strany již osazené desky plošného spoje.



Obr. 16 Obě strany osazené desky plošného spoje

4.2 Program

Program je vytvořen ve vývojovém prostředí STM32CubeIDE v programovacím jazyce C. Prototyp neobsahuje žádné rozhraní pro vstup od uživatele. To znamená, že po připojení ke zdroji se prototyp zapne a spustí se program. Program běží, dokud není zařízení odpojeno od zdroje. Jednotlivé části programu jsou prováděny buď v určitých časových intervalech nebo v závislosti na dokončení specifické části kódu. Díky tomu je možné udržet dostatečnou výpočetní rychlost.

Po spuštění programu dojde k inicializaci potřebných periférií, konfiguraci systémových hodin a nastavení počátečních hodnoty deklarovaných proměnných. Následuje zapnutí diody a spuštění časovačů TIM14 a TIM16. Úkolem časovače TIM14 je pravidelně každou 1ms volat přerušení pro spuštění konverze ADC.

Čtení hodnot ADC

Součástí přerušení ADC je filtrace pomocí aritmetického průměru šesti po sobě jdoucích hodnot z ADC. Tato filtrace slouží k částečnému odfiltrování výkyvů hodnot způsobených jemným pohybem a šumem, který se nepodařilo odstranit pomocí obvodových filtrů.

```
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
{
    if (a<6) { // zápis do prvního pole (pole A)
        d_A_ready = 0; // znemožnění čtení hodnoty pole A
        if (a==0){ //když probíhá zápis do pole A
            ADCValue=(d_B[0]+d_B[1]+d_B[2]+d_B[3]+d_B[4]+d_B[5])/6; //průměr B
            d_B_ready = 1; //hodnota pole B připravena ke čtení
        }
        d_A[a]=HAL_ADC_GetValue(&hadc1); //načítání hodnoty z ADC do pole A
        a++;
    }else { // zápis do druhého pole (pole B) je-li pole A naplněné
        d_B_ready = 0; //znemožnění čtení hodnoty pole B
        if (a==6){ //když probíhá zápis do pole B
            ADCValue=(d_A[0]+d_A[1]+d_A[2]+d_A[3]+d_A[4]+d_A[5])/6; //průměr A
            d_A_ready = 1; //hodnota pole A připravena ke čtení
        }
        d_B[a-6]=HAL_ADC_GetValue(&hadc1); //načítání hodnoty z ADC do pole B
        a++;
        if (a>12-1){ //reset po naplnění pole B
            a=0;
        }
    }
}
```

Popis funkce filtru: filtr je tvořen dvěma poli (d_A a d_B). Pole d_A se postupně plní šesti hodnotami z ADC. Po naplnění pole d_A dojde k přepnutí na pole d_B, které se začne plnit hodnotami z ADC. V průběhu plnění pole d_B je vypočten aritmetický průměr hodnot pole d_A. Tento průměr je uložen do proměnné ADCValue. Proměnná d_A_ready indikující připravenost ADCValue pro další zpracování je nastavena na 1. Po naplnění pole d_B se

přepne zpět na pole d_A. Proměnná d_A_ready se nastaví na 0. Pole d_A se začne opět plnit hodnotami z ADC a provede se výpočet aritmetického průměru hodnot pole d_B. Průměr je uložen do ADCValue a proměnná d_B_ready je nastavena na 1. Výše popsaný cyklus se opakuje po celou dobu funkce programu. Dvě pole jsou použita proto, aby nedocházelo k přepsání hodnot v poli před jejich zpracováním.

Výpočet průměrné hodnoty

Pro algoritmus detekce tepu, který je popsán v následující části textu je použita průměrná hodnota. Tato průměrná hodnota se počítá od přiložení prstu až po jeho odebrání. Přiložení, respektive odebrání prstu je detekováno pomocí funkce if. Je-li prst přiložen, hodnoty ADC se pohybují v rozsahu 0-1500. Přiložení (odebrání) prstu je doprovázeno napětíovou špičkou, která vyžene hodnotu ADC nad úroveň 2000. Při správném používání jde o jedinou situaci, kdy k tomuto jevu dochází. Proto jde o nejspolehlivější způsob detekce přiložení (odebrání) prstu.

Algoritmus pro výpočet průměrné hodnoty vypadá takto:

```
if (ticks - test_time > 2000) { //začne 2s po přiložení prstu
    if (ADCValue > 1000) { //zamezení zkreslení vlivem pohybu
        up = 400;
    } else {
        up = ADCValue;
    }
    e++; //jmenovatel
    value = value + up; //výpočet čitatele
    ADC_avg = value / e; //výpočet průměrné hodnoty
    if (value > 4200000000) { //ochrana před přetečením
        e = 0;
        value = 0;
    }
}
```

Dvě sekundy po přiložení prstu se začne vypočítávat aritmetický průměr hodnot ADC. Součástí algoritmu je i ošetření zákmitů, které jsou způsobeny pohybem. Místo hodnoty ADC při zákmitu je připočtena hodnota, která tolik neovlivní celkový průměr.

Výpočet tepové frekvence

Celý blok výpočtu tepové frekvence včetně výpočtu průměrné hodnoty je umístěn v přerušení. Toto přerušení je voláno periodicky časovačem TIM16 nastaveným na 12ms a má nastavenou nižší prioritu, než mají ostatní přerušení. Nedochozí tak k přerušení čtení z ADC, a zároveň je možné přerušit funkci pro zápis na displej. Výpočet průměrné hodnoty a tepové frekvence je podmíněn připraveností dat z ADC (proměnná `d_A_ready` nebo `d_B_ready` musí být rovna 1). Součástí tohoto bloku kódu je mimo vlastní výpočet tepu i výše zmíněná detekce prstu. Kód pro výpočet tepové frekvence vypadá takto:

```

if (ticks - test_time > 5000) { //výpočet začne 5s po přiložení prstu

    peak = (helpp[2] + helpp[3]) / 2;
    /*algoritmus pro výpočet tepu*/
    if (peak < ADC_avg) { //lokální průměr < celkový: propad
        vrchol = 0;
    }
    if (peak > ADC_avg && vrchol == 0) { //lok. prům. > celkový: vrchol
        vrchol = 1;
        g++;
        if (g == 1) { //první vrchol
            ticks1 = HAL_GetTick();
        }
        if (g == 5) { //pátý vrchol
            ticks2 = HAL_GetTick();
            period = ticks2 - ticks1; // čas mezi 5 vrcholy
            pulsew = period / 4; // výpočet šířky jednoho pulzu
            tep = 60000 / pulsew; // výpočet tepové frekvence
            ticks1 = 0;
            ticks2 = 0;
            g = 0;
        }
    }
}
}

```

Funkce algoritmu: Po přiložení prstu se počká 5 sekund. Nejdříve se vypočítá peak. Peak je lokální průměr dvou po sobě jdoucích hodnot ADC. Hodnota proměnné peak je po vypočtení porovnána s celkovou průměrnou hodnotou `ADC_avg`. Je-li hodnota peak vyšší než průměrná hodnota `ADC_avg`, jedná se o vrchol. Při prvním vrcholu se uloží čas, kdy daný vrchol nastal, do proměnné `ticks1`. Další vrchol je nalezen stejným způsobem. Pro detekci dalšího vrcholu je potřeba, aby hodnota ADC, respektive peak, klesla pod průměrnou hodnotu. Po zaznamenání pěti vrcholů je uložen aktuální čas do proměnné `ticks2`. Odečtením `ticks1` od `ticks2` je získána doba mezi pěti vrcholy. Vydělením této hodnoty čtyřmi je stanovena doba jedné periody tepu. Vydělením jedné minuty dobou jedné periody je vypočtena tepová frekvence.

Výpis na displej

Komunikace Nuclea-32 s OLED displejem je zajištěna pomocí sběrnice I²C s použitím knihovny stm32-ssd1306. Autorem této knihovny je uživatel afiskon [14], knihovna je dostupná pro veřejné použití na stránkách GitHub.

Z této knihovny používám funkce:

```
ssd1306_Init(); //inicializace displeje  
ssd1306_SetCursor(souřadnice_x, souřadnice_y); // nastavení pozice textu  
ssd1306_WriteString(mujText, Font_7x10, White); //výpis textu  
ssd1306_UpdateScreen(); //zápis na obrazovku
```

Tyto funkce jsou použity pro výpis textu a hodnoty tepové frekvence na displej. Pro vykreslení grafu a malého srdce je použita funkce pro kreslení přímky z jednoho bodu do druhého:

```
ssd1306_Line(počátek x, počátek y, konec x, konec y, White);
```

K výpisu hodnoty a vykreslení grafu slouží kód na následující straně. Tento kód je proveden každých 120ms a skládá se ze tří částí. V první části se pomocí měřítka přepočítají hodnoty ADC tak, aby se vešly na displej. V druhé části je pomocí linek nakreslen graf průběhu srdeční činnosti. Třetí a poslední část zabezpečuje výpis číselné hodnoty tepové frekvence. Mezi druhou a třetí částí se nachází podmínka pro volání funkce, která vykreslí malé srdce v pravém horním rohu displeje.

Vykreslení grafu a výpis hodnoty na displej:

```

if (ticks-graph_time>120){//funkce je provedena každých 120ms
    i++;

    if(ADCValue>=1092){ //horní omezení grafu
        GRPValue = 20;
    }
    else if(ADCValue<26){ //dolní omezení grafu
        GRPValue = 62;
    } else {
        GRPValue = 62-(ADCValue/26); //přepoččet hodnot z ADC, do grafu
    }

    if (i>10){
        y_position[i] = GRPValue; //uložení převedené hodnoty do pole

        ssd1306_FillRectangle(60, 0, 121, 63, Black); //"vymazání
//vykreslení grafu
        ssd1306_Line(78, y_position[i], 81, y_position[i - 1],White);
        ssd1306_Line(81, y_position[i - 1], 84, y_position[i - 2],White);
        ssd1306_Line(84, y_position[i - 2], 87, y_position[i - 3],White);
        ssd1306_Line(87, y_position[i - 3], 90, y_position[i - 4],White);
        ssd1306_Line(90, y_position[i - 4], 93, y_position[i - 5],White);
        ssd1306_Line(93, y_position[i - 5], 96, y_position[i - 6],White);
        ssd1306_Line(96, y_position[i - 6], 99, y_position[i - 7],White);
        ssd1306_Line(99, y_position[i - 7], 102, y_position[i - 8],White);
        ssd1306_Line(102, y_position[i - 8], 108, y_position[i - 9],White);
        ssd1306_Line(108, y_position[i -9], 111, y_position[i -10], White);

        if (i>990){ //restart pole
            i=10;
        }

        if (GRPValue<62-(ADC_avg/26)){ //bliknutí srdce pro vrchol
            small_heart();
        }
    }
    sprintf(bpm, "%lu", tep); //převedení tepu na string

    ssd1306_FillRectangle(5, 25, 60, 55, Black); //vyčištění displeje
    ssd1306_SetCursor(5, 25); //nastavení polohy textu
    if (tep<36||tep>240){ //ověření validnosti zobrazované hodnoty
        ssd1306_WriteString(dead, Font_16x26, White); //není-li validní -
    }else {
        ssd1306_WriteString(bpm, Font_16x26, White); // je-li validní
    }
    ssd1306_UpdateScreen(); //zápis na displej
    graph_time=ticks;
}

```

5 Testování

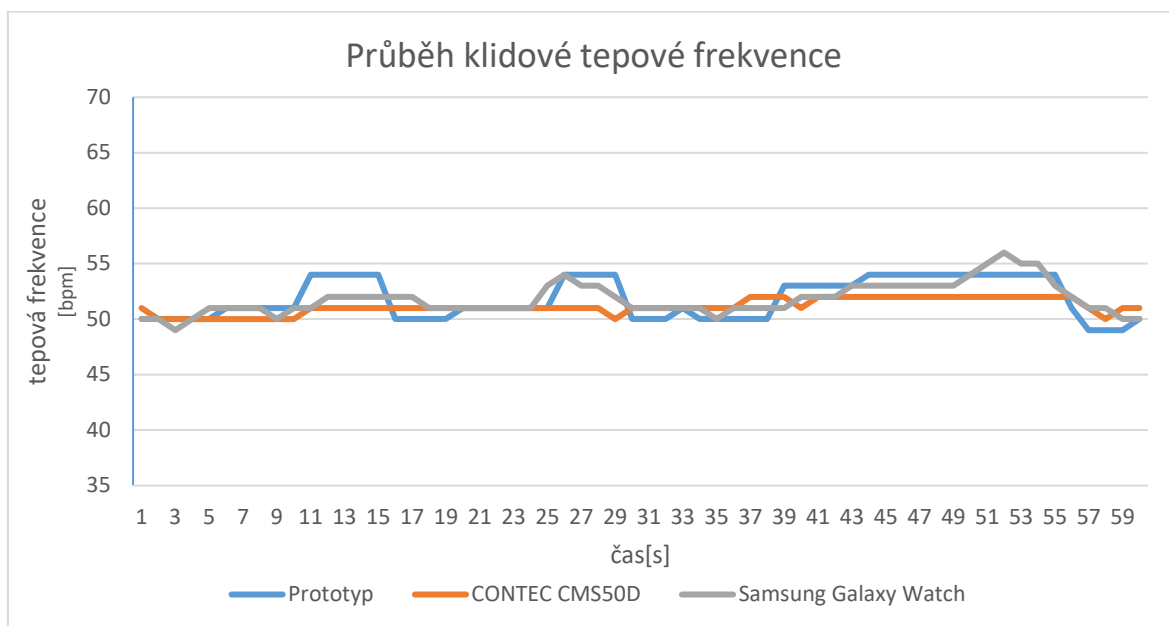
Testování je provedeno srovnáním prototypu s pulzním oxymetrem CMS50D značky CONTEC a chytrými hodinkami Samsung Galaxy Watch. Pulzní oxymetr dle specifikací pracuje s přesností ± 2 bpm pro rozsah 30bpm – 99bpm. Na rozsahu 100bpm až 250bpm pracuje s přesností 2%

První srovnávací test je proveden v rychlosti zobrazení první validní hodnoty po přiložení prstu, respektive po spuštění měření. Měření je čas mezi spuštěním (přiložením) a prvním zobrazením hodnoty tepové frekvence, která odpovídá aktuální tepové frekvenci s tolerancí 5%.

V testu si vedl nejlépe pulzní oxymetr s průměrnou dobou do první validní hodnoty 8,7 sekund. Prototyp zobrazil první validní hodnotu v průměru po 13,1 sekundách. U hodinek v tomto testu byly použity dvě aplikace. Jedna měří jednorázově aktuální hodnotu tepové frekvence. Ta ukázala validní hodnotu tepu v průměru po 9,5 sekundách od spuštění měření. Druhá aplikace měří tepovou frekvenci kontinuálně. V případě této aplikace došlo k zobrazení první validní hodnoty v průměru po 27,3 sekundách.

Důvodů pro nižší rychlost zobrazení první validní hodnoty u prototypu je více. Prvním důvodem je čekání 5s po přiložení prstu, než se spustí algoritmus pro výpočet tepu. Toto čekání je zakomponováno do programu z důvodu ustálení hodnoty ADC po přiložení prstu a stabilizování průměrné hodnoty ADC_avg. Dalším důvodem je samotný výpočet tepu, pro který je potřeba dosáhnout pěti vrcholů (tepů). Zároveň ne vždy je první zobrazená hodnota v toleranci 5% vůči reálné hodnotě. V tomto případě se čeká znovu po dobu pěti vrcholů.

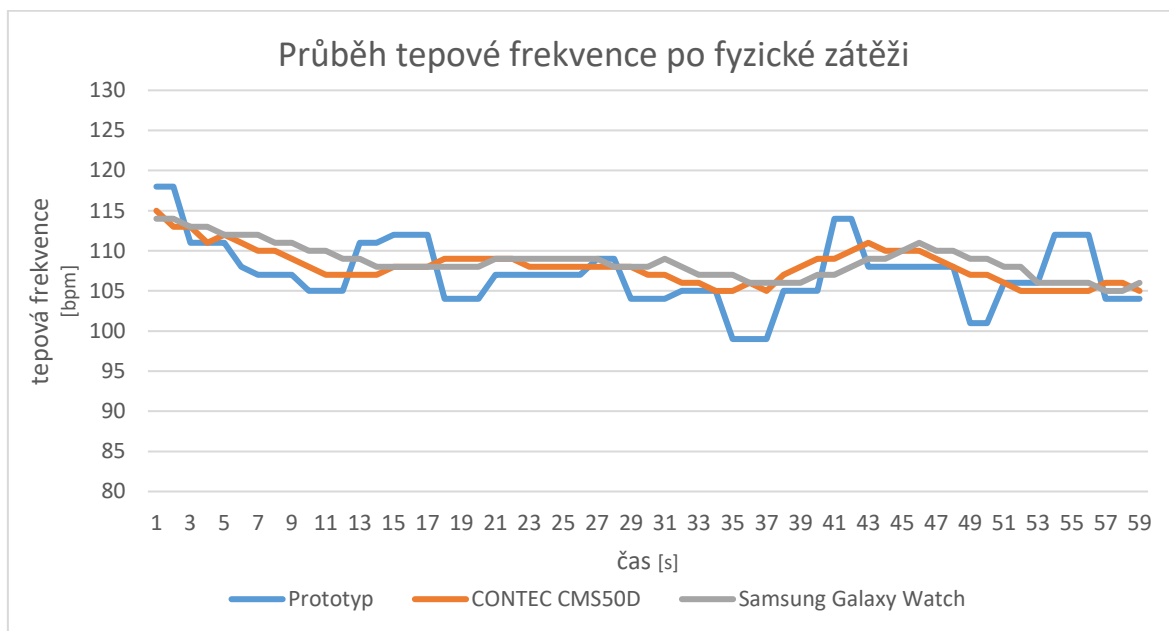
Druhý srovnávací test, testoval přesnost a stabilitu měření tepové frekvence. V grafu 1 je zobrazena hodnota klidové tepové frekvence po dobu jedné minuty.



Graf 1 Průběh klidové tepové frekvence po dobu jedné minuty

V grafu je vidět, že pro klidovou tepovou frekvenci je hodnota zobrazovaná prototypem podobná hodnotám na ostatních zařízeních. Případné výkyvy jsou způsobeny pohybovým faktorem.

Třetí srovnávací test je podobný druhému, ale tepová frekvence je měřena těsně po fyzické zátěži. Průběh tepové frekvence po fyzické zátěži je vidět v grafu 2. Měření opět probíhalo po dobu jedné minuty.



Graf 2 Průběh tepové frekvence po zátěži

Z grafu je patrné, že prototyp je náchylnější na výkyvy než zařízení, se kterými je srovnáván. Výkyvy jsou způsobeny pohybem prstu na senzoru. Se stoupající tepovou frekvencí roste i vliv zákmitů způsobených nedostatečně pevným uchycením prstu.

Výpočtem průměru hodnot tepové frekvence měřené během druhého a třetího testu získáme obsah tabulky 1.

Tabulka 1 Průměrné hodnoty tepové frekvence

	Průměrná hodnota [BPM]	
	Druhý test	Třetí test
Prototyp	51,82	107,17
CONTEC CMS50D	51,13	108
Samsung Galaxy Watch	51,78	108,6

Porovnáním dat v tabulce je zjištěno, že celková přesnost prototypu se pohybuje v toleranci do 2%.

6 Možná vylepšení

V průběhu vývoje prototypu se objevilo několik možných vylepšení, která nebyla implementována. Potencionální vylepšení lze rozdělit do dvou kategorií mechanická a programová.

Mechanická vylepšení

Nejefektivnější mechanické vylepšení je rozdělení jedné desky plošného spoje na dvě. Jedna deska by obsahovala světelný zdroj a displej. Druhá, obvod pro detekci, úpravu a zpracování signálu. Obě desky by byly pomocí pouzdra posazeny proti sobě. Takové uspořádání by umožnilo použití SMD diody namísto THT. Došlo by tak ke zmenšení celého prototypu. Zároveň i k zjednodušení světelného odstínění a upevnění měřicího přístroje na prst.

Programová vylepšení

Prvním vylepšením programu by mohlo být obohacení o algoritmus pro výpočet okysličení krve jako je tomu u pulzních oxymetrů. Obvod je elektricky i mechanicky již pro toto vylepšení připraven díky použití infračervené i červené LED diody.

Dalším možným vylepšením je využití lepších softwarových filtrů pro filtraci hodnot ADC. Používaný filtr aritmetickým průměrem je velice přímočarý, a ne vždy dostačující. Mezi možné filtrace lze zařadit například použití váženého průměru. U něhož by hodnoty, které neodpovídají předpokladu, byly započítávány s nižší vahou. Dojde tak ke zvýšení přesnosti měření. Současně lze zařadit i filtr zobrazované tepové frekvence. Tento filtr by nezobrazil tepovou frekvenci, která se vymyká stanovenému trendu.

Dále lze vylepšit rychlost zobrazení první validní hodnoty. Zvýšení této rychlosti se dá docílit úpravou poměru rychlosti a míry filtrace. Případně je možné také optimalizovat dobu čekání před počátkem výpočtu.

Zhodnocení a závěr

Cílem této bakalářské práce je seznámení se s principem měření tepové frekvence pomocí absorpce světla. Teoretický základ se podařilo vybudovat v první části této práce, kde je popsána anatomie a činnost lidského srdce. Dále došlo k seznámení se s některými způsoby měření tepové frekvence. Zvláštní pozornost je věnována fotoelektrické pletysmografii, která pro měření tepové frekvence užívá optický senzor a princip absorpce světla.

Ve druhé části práce je navrhnout přístroj pro měření tepu. Přístroj pro svoji činnost používá průsvitovou fotoelektrickou pletysmografii. Jako světelný zdroj je u přístroje použita červená LED dioda. Práci detektoru zastává obvod OPT101, jehož součástí je fotodioda. Detekovaný signál je upraven pomocí filtrů a zesilovače. Upravený analogový signál je převeden na digitální a zpracován mikrokontrolerem. Vypočtená hodnota tepové frekvence je zobrazena na OLED displeji společně s grafem srdeční činnosti.

Navržený prototyp byl úspěšně sestaven a podroben srovnávacímu testu s komerčně dostupnými kardiotachometry. Během testování byl odhalen problém s měřením tepové frekvence nad 90 tepů za minutu. Důvodem byla rychlost zpracování dat, která byla snížena časovou náročností programového bloku pro výpis na displej. Vypočtená hodnota tepové frekvence potom neodpovídala realitě. Problém byl vyřešen přesunutím algoritmu pro výpočet tepové frekvence do přerušení, které je voláno časovačem. Díky tomu je výpočet prováděn s dostatečnou frekvencí nezávisle na stavu výpisu na displej. Po opravě bylo testování zopakováno.

Výsledky srovnávacího testu ukázaly malé nedostatky v oblastech přesnosti a rychlosti v průběhu měření u prototypu ve srovnání s komerčními zařízeními. Nepřesnosti jsou způsobeny zejména pohybovým faktorem. Vliv tohoto faktoru stoupá s rostoucí tepovou frekvencí. Nedostatek je však možné vyřešit implementováním některého z navrhovaných vylepšení nebo jejich kombinací. Použitím mechanického vylepšení, rozdělením desky plošného spoje na dvě, lze zlepšit polohu a fixaci prstu na senzor. Zároveň lze doplnit program o softwarovou filtraci pohybu a tím omezit vliv pohybu prstu na výslednou hodnotu tepové frekvence. Celková přesnost zařízení je však srovnatelná s komerčně dostupnými zařízeními. Po splnění všech bodů zadání lze říci, že vyrobený prototyp je schopen pomoci při orientačním měření tepové frekvence v domácích podmínkách. Pro tento způsob využití je přesnost i rychlost prototypu dostačující.

Literatura

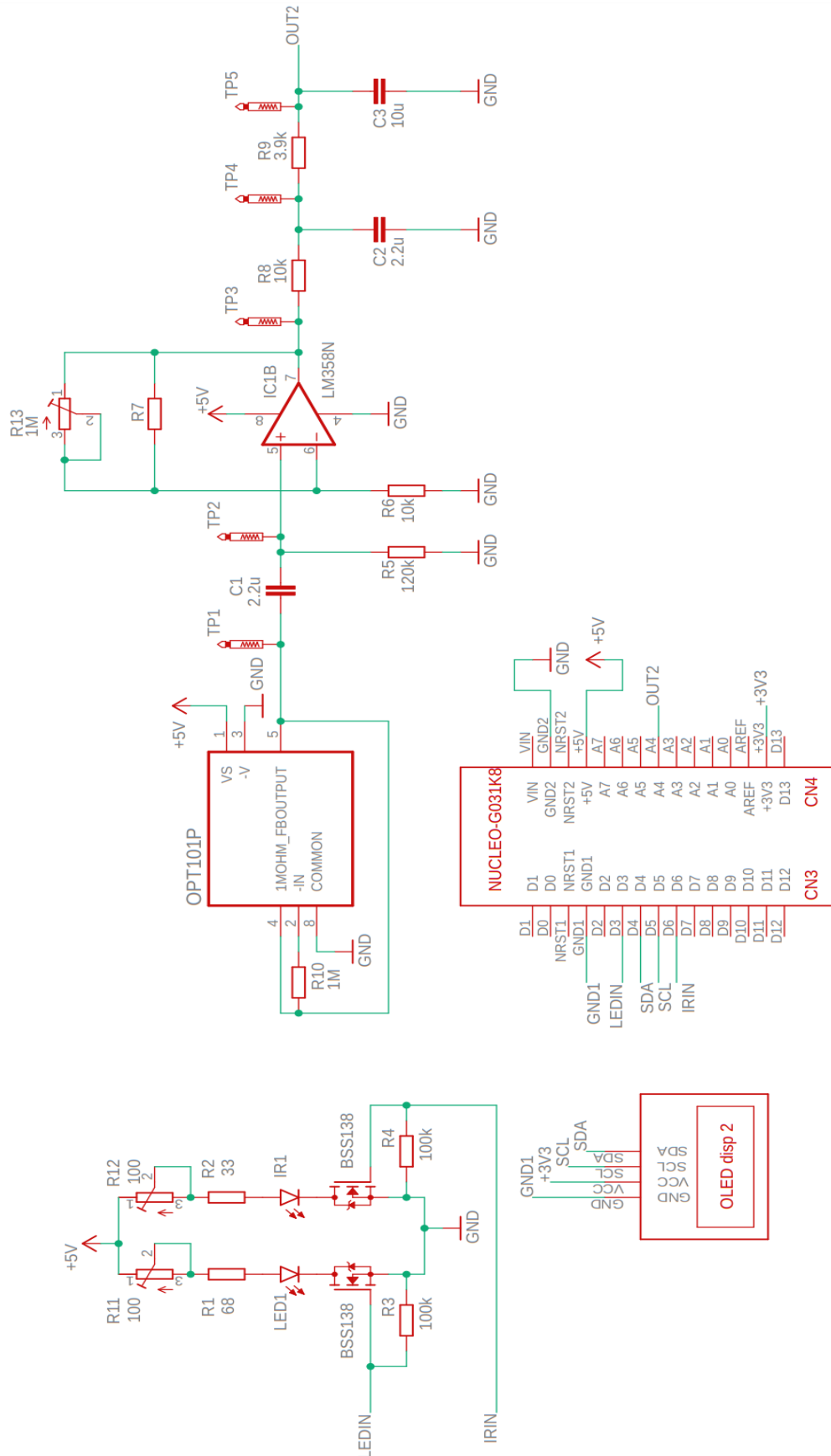
- [1] TROJAN, Stanislav. *Lékařská fyziologie*. Vyd. 4., přeprac. a dopl. Praha: Grada, 2003. ISBN 80-247-0512-5.
- [2] HRAZDIRA, Ivo, Vojtěch MORNSTEIN a Jiřina ŠKORPÍKOVÁ. *Základy biofyziky a zdravotnické techniky*. Brno: Neptun, c2006. ISBN 80-86850-01-3.
- [3] HRAZDIRA, Ivo a Vojtěch MORNSTEIN. *Lékařská biofyzika a přístrojová technika*. Brno: Neptun, 2001. ISBN 80-902896-1-4.
- [4] JIRKŮ, Kateřina. JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH. *Kardiovaskulární soustava I: Apparatus angiologia* [online]. České Budějovice: Katedra medicínské biologie, 2022 [cit. 2023-05-11]. Dostupné z: https://prf.jcu.cz/images/PRF/fakulta/katedry/kme/studijni-materialy/zaklady-funkcni-anatomie-cloveka/duben2022/7-FA_kardiovaskularni_soustava_I_2022.pdf
- [5] Příspěvatelé WikiSkript, *Elektrokardiografie* [online], , c2023, Datum poslední revize 21. 02. 2023, 15:26 UTC, [citováno 11. 05. 2023]. Dostupné z: <https://www.wikiskripta.eu/index.php?title=Elektrokardiografie&oldid=460723>
- [6] Wikipedia contributors. Pulse. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2023, 22.1.2023 [cit. 2023-05-11]. Dostupné z: <https://en.wikipedia.org/w/index.php?title=Pulse&oldid=1135081929>
- [7] FINNERTY, Robert. How to Design a Better Pulse Oximeter. In: *ANALOG DEVICES: AHEAD OF WHAT'S POSSIBLE* [online]. [cit. 2023-05-11]. Dostupné z: <https://www.analog.com/en/technical-articles/how-to-design-a-better-pulse-oximeter.html>
- [8] DEDHIA, Parin. Solar Powered Pulse Oximeter and Heart Rate Meter. In: *Cornell University* [online]. Ithaca: Cornell University, 2012 [cit. 2023-05-11]. Dostupné z: https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/f2012/prd47/PulseOximeter/Pulse_ox.html
- [9] STMicroelectronics. UM2591: User manual. In: *ST: life.augmented* [online]. 25.6.2019 [cit. 2023-05-11]. Dostupné z: https://www.st.com/resource/en/user_manual/dm00622380-stm32g0-nucleo-32-board-mb1455-stmicroelectronics.pdf

- [10] TEXAS INSTRUMENTS. OPT101 Monolithic Photodiode and Single-Supply Transimpedance Amplifier: datasheet. In: *Texas Instruments* [online]. Dallas: Texas Instruments Incorporated, 1994, 6.2015 [cit. 2023-05-11]. Dostupné z: https://www.ti.com/lit/ds/symlink/opt101.pdf?ts=1683809097934&ref_url=https%253A%252F%252Fwww.google.com%252F
- [11] Srdce, cévy, krevní oběh: obrázek srdce popis. In: *Umíme fakta* [online]. [cit. 2023-05-11]. Dostupné z: https://www.umimeto.org/asset/system/uf/img/zadani-biologie/vysvetleni/srdce_popis.svg
- [12] TÝM REHABILITACE.INFO. Základní čtení a hodnocení EKG křivky: obrázek tep. In: *Rehabilitace.info: Magazín o zdraví* [online]. 12.10.2012 [cit. 2023-05-11]. Dostupné z: <https://www.rehabilitace.info/wp-content/uploads/2012/10/tep.jpg>
- [13] JUBRAN, Amal. Pulse oximetry. In: *BioMed Central* [online]. 13.5.1999 [cit. 2023-05-11]. Dostupné z: <https://ccforum.biomedcentral.com/counter/pdf/10.1186/cc341.pdf>
- [14] AFISKON. Stm32-ssd1306: STM32 library for OLEDs based on SSD1306. In: *GitHub* [online]. 2018 [cit. 2023-05-11]. Dostupné z: <https://github.com/afiskon/stm32-ssd1306>
- [15] JIŘÍK, Ondřej. *Měření tepové frekvence* [online]. Brno, 2012 [cit. 2023-05-11]. Bakalářská práce (Bc.). VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ. Vedoucí práce Doc. Ing. Milan Chmelař, CSc.
- [16] BAĐUROVÁ, Olga. *Srovnání a vyhodnocení měření oxymetrie pro různé typy snímačů a jejich umístění* [online]. Ostrava, 2013 [cit. 2023-05-11]. Diplomová práce. VŠB – TECHNICKÁ UNIVERZITA OSTRAVA. Vedoucí práce Ing. Blanka Filipová, Ph.D.

Přílohy

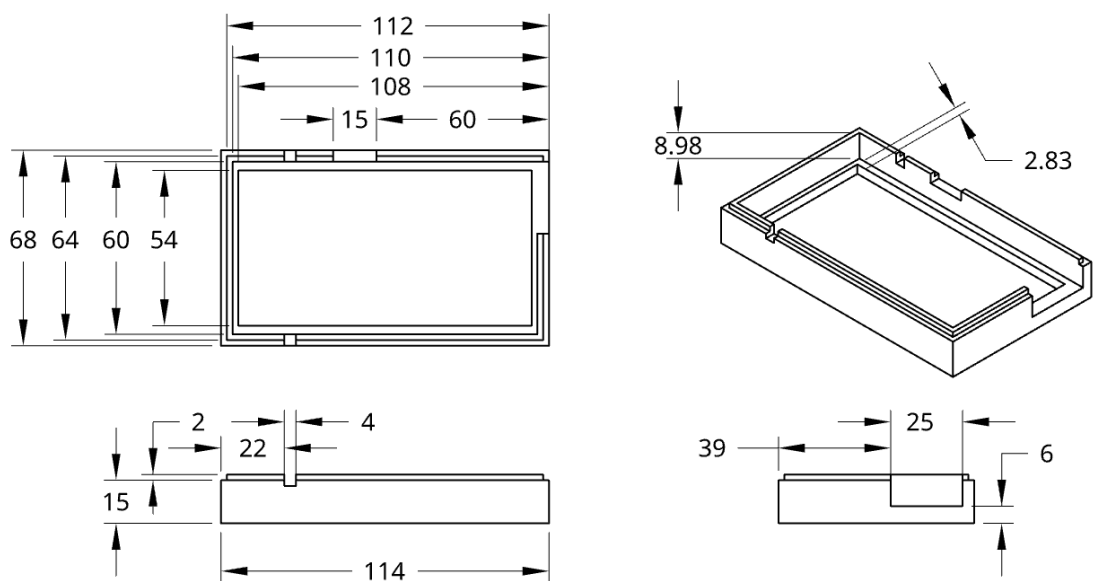
Obsah příloh	Číslo strany
Schéma	II
Plány krabičky	III
3D model krabičky	V
Program	VI
Testování	XV

Schéma:

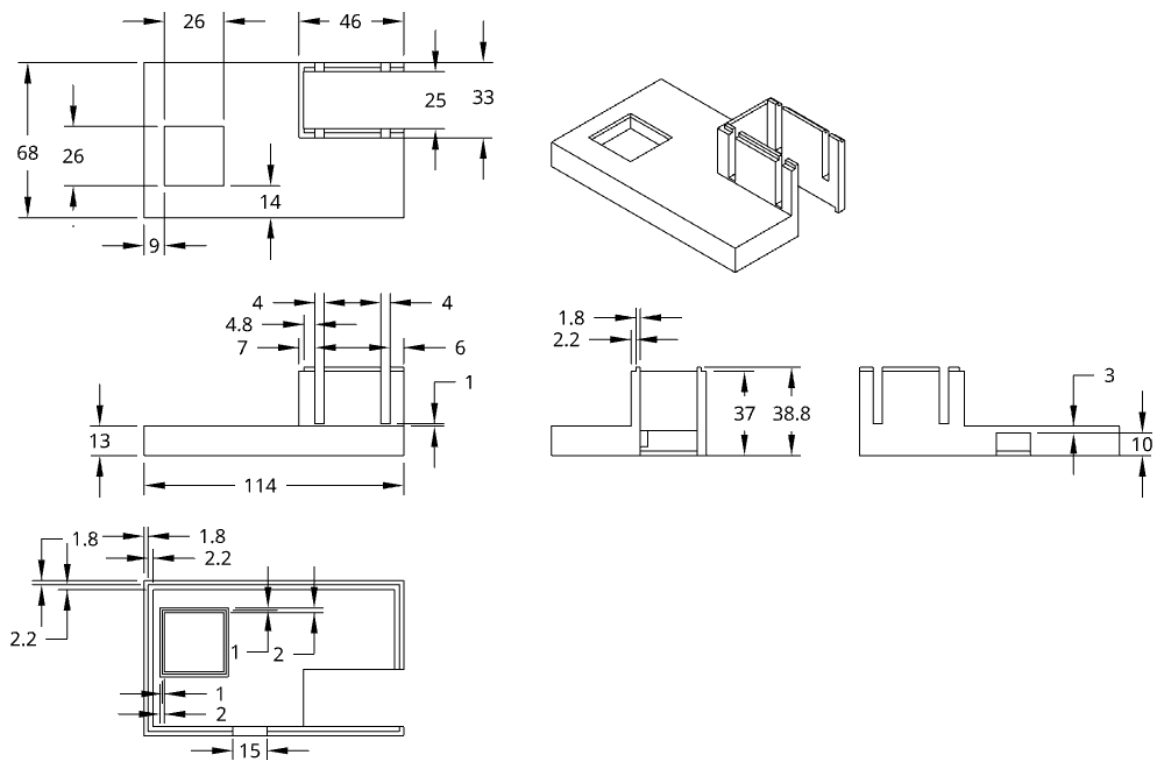


Obrázek 1 Schéma prototypu

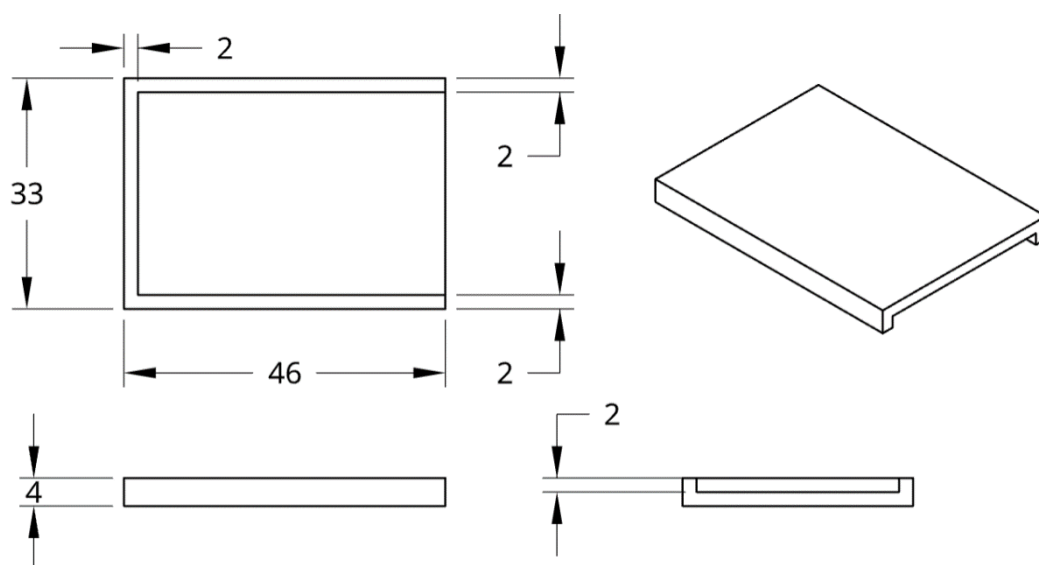
Plány krabičky:



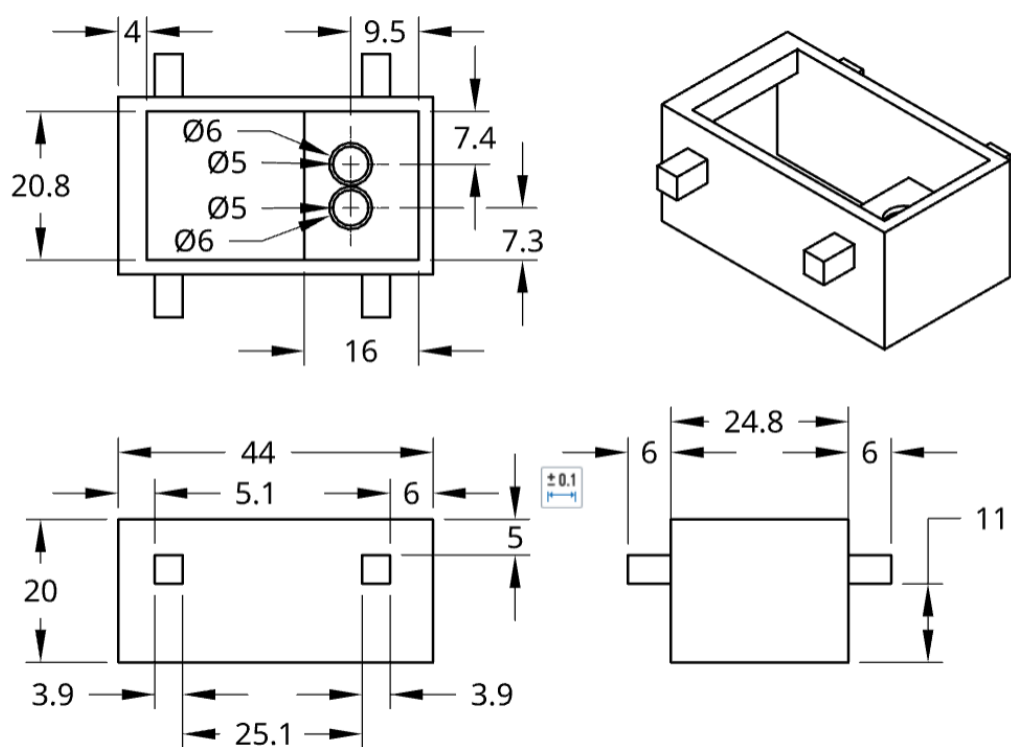
Obrázek 2 Dno krabičky



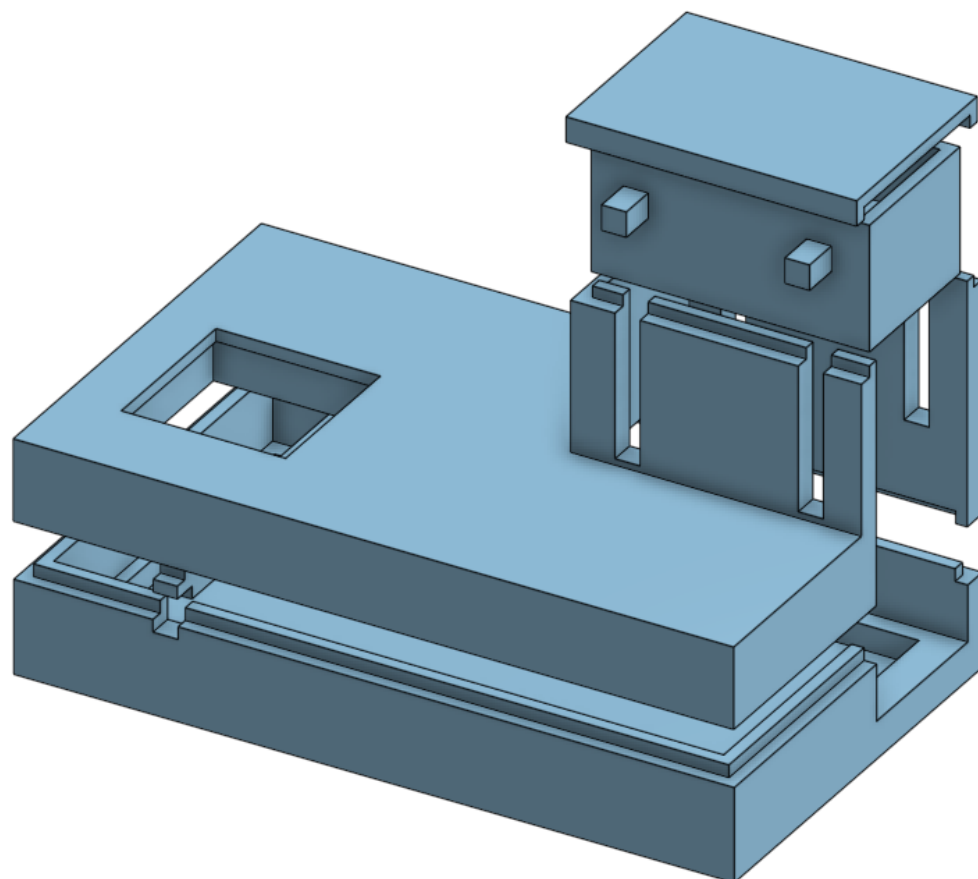
Obrázek 3 Svrchní část krabičky



Obrázek 4 Víčko krabičky



Obrázek 5 Držák na diody

3D model krabičky:

Obrázek 6 3D model krabičky

Program:

```

/* USER CODE BEGIN Header */
/**
 * *****
 * @file      : main.c
 * @brief     : Main program body
 * *****
 * @attention
 *
 * Copyright (c) 2023 STMicroelectronics.
 * All rights reserved.
 *
 * This software is licensed under terms that can be found in the LICENSE file
 * in the root directory of this software component.
 * If no LICENSE file comes with this software, it is provided AS-IS.
 *
 * *****
 */
/* USER CODE END Header */
/* Includes -----*/
#include "main.h"

/* Private includes -----*/
/* USER CODE BEGIN Includes */
#include "ssd1306.h"
#include <string.h>
#include <stdio.h>
/* USER CODE END Includes */

/* Private typedef -----*/
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define -----*/
/* USER CODE BEGIN PD */
/* USER CODE END PD */

/* Private macro -----*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables -----*/
ADC_HandleTypeDef hadc1;

I2C_HandleTypeDef hi2c1;

TIM_HandleTypeDef htim14;
TIM_HandleTypeDef htim16;

UART_HandleTypeDef huart2;

/* USER CODE BEGIN PV */
/* USER CODE END PV */

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_USART2_UART_Init(void);
static void MX_ADC1_Init(void);
static void MX_I2C1_Init(void);
static void MX_TIM14_Init(void);
static void MX_TIM16_Init(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code -----*/
/* USER CODE BEGIN 0 */
uint16_t ADCValue = 0; //hodnota z ADC

```

```

uint16_t d_A[] = { 0, 0, 0, 0, 0, 0, 0, 0 };
uint16_t d_B[] = { 0, 0, 0, 0, 0, 0, 0, 0 };
uint16_t d_A_ready = 0;
uint16_t d_B_ready = 0;
uint16_t a = 0; // počítání ADC
uint16_t GRPValue = 0; // přepočtená hodnota z ADC
volatile uint32_t period = 0; // čas mezi jednotlivými vrcholy tepu
uint32_t ticks1 = 0; // čas
uint32_t ticks2 = 0; // čas
uint32_t helpp[] = { 0, 0, 0, 0 };
uint32_t tep = 0; // hodnota tepu
uint32_t tpp[] = { 70, 70, 70, 70 };
uint16_t vrchol = 0; //počet průběhů whilu
uint16_t ADC_avg = 0; //pomocná proměnná
uint16_t peak = 0; //pomocná proměnná
uint32_t value = 0;
uint32_t e = 0;
uint16_t g = 0; // počítání vrcholů
uint16_t r = 0; //počet průběhů whilu
uint16_t up = 0; //počet průběhů whilu
uint32_t pulseW = 0; // šířka jednoho pulzu
uint32_t ticks = 0;
uint32_t test_time = 0;
/* USER CODE END 0 */

/**
 * @brief The application entry point.
 * @retval int
 */
int main(void) {
    /* USER CODE BEGIN 1 */
    char myText[] = "BPM:"; //text na disp.
    char dead[] = " - "; //text na disp.
    uint16_t k = 96; //počáteční pozice pro obrázek srdce x
    uint16_t l = 1; //počáteční pozice pro obrázek srdce y
    char bpm[10]; // pro výpis hodnot z Tepu
    uint16_t y_position[1000]; // pole přepočtených hodnot pro vykreslení grafu
    uint16_t i = 10; //počet průběhů whilu
    uint16_t x; //pomocná proměnná

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_USART2_UART_Init();
    MX_ADC1_Init();
    MX_I2C1_Init();
    MX_TIM14_Init();
    MX_TIM16_Init();
    /* USER CODE BEGIN 2 */

    /* výpis textu na disp */
    ssd1306_Init(); //inicializace displeje
    ssd1306_SetCursor(15, 5); // nastavení pozice textu
    ssd1306_WriteString(myText, Font_7x10, White); //výpis textu

```

```

ssd1306_UpdateScreen(); //zápis na obrazovku

/*naplnění prvních hodnot v poli pro jistotu*/
for (x = 0; x < 12; x++) {
    y_position[x] = 60;
}
/*vykreslení srdce*/
void small_heart() {
    ssd1306_Line(k + 1, l + 3, k + 1, l + 6, White);
    ssd1306_Line(k + 2, l + 2, k + 2, l + 7, White);
    ssd1306_Line(k + 3, l + 1, k + 3, l + 8, White);
    ssd1306_Line(k + 4, l + 2, k + 4, l + 9, White);
    ssd1306_Line(k + 5, l + 3, k + 5, l + 10, White);
    ssd1306_Line(k + 6, l + 4, k + 6, l + 11, White);
    ssd1306_Line(k + 7, l + 5, k + 7, l + 12, White);
    ssd1306_Line(k + 8, l + 4, k + 8, l + 11, White);
    ssd1306_Line(k + 9, l + 3, k + 9, l + 10, White);
    ssd1306_Line(k + 10, l + 2, k + 10, l + 9, White);
    ssd1306_Line(k + 11, l + 1, k + 11, l + 8, White);
    ssd1306_Line(k + 12, l + 2, k + 12, l + 7, White);
    ssd1306_Line(k + 13, l + 3, k + 13, l + 6, White);
}

uint32_t graph_time = HAL_GetTick();

ticks = HAL_GetTick();
test_time = HAL_GetTick();

//start timer
HAL_TIM_Base_Start_IT(&htim14);
HAL_TIM_Base_Start_IT(&htim16);
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1) {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */

    ticks = HAL_GetTick();
    //rozsvícení diod
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, GPIO_PIN_SET);

    /*vykreslení grafu průběhu teploty */
    if (ticks - graph_time > 120) {
        i++;
        if (ADCValue >= 1092) {
            GRPValue = 20;
        } else if (ADCValue < 26) {
            GRPValue = 62;
        } else {
            GRPValue = 62 - (ADCValue / 26); //přepoččet hodnot z ADC,
        }
        if (i > 10) {
            y_position[i] = GRPValue; //uložení převedené hodnoty do pole

            ssd1306_FillRectangle(60, 0, 121, 63, Black);
            //vykreslení grafu
            ssd1306_Line(78, y_position[i], 81, y_position[i - 1],White);
            ssd1306_Line(81, y_position[i - 1], 84, y_position[i - 2],White);
            ssd1306_Line(84, y_position[i - 2], 87, y_position[i - 3],White);
            ssd1306_Line(87, y_position[i - 3], 90, y_position[i - 4],White);
            ssd1306_Line(90, y_position[i - 4], 93, y_position[i - 5],White);
            ssd1306_Line(93, y_position[i - 5], 96, y_position[i - 6],White);
            ssd1306_Line(96, y_position[i - 6], 99, y_position[i - 7],White);
            ssd1306_Line(99, y_position[i - 7], 102, y_position[i - 8],White);
            ssd1306_Line(102, y_position[i - 8], 108, y_position[i - 9],White);
            ssd1306_Line(108, y_position[i - 9], 111, y_position[i - 10], White);
            if (i > 990) {
                i = 10;
            } //restart pole
        }
    }
}

```

```

        if (GRPValue < 62 - (ADC_avg / 26)) { // bliknutí srdce
            small_heart();
        }
    }

    sprintf(bpm, "%lu", tep); //převedení tepu na string

    ssd1306_FillRectangle(5, 25, 60, 55, Black); //vyčištění displeje
    ssd1306_SetCursor(5, 25); //nastavení polohy textu
    if ( tep < 36 || tep > 240) { //ověření validnosti zobrazované hodnoty
        ssd1306_WriteString(dead, Font_16x26, White); //není-li validní -
    } else {
        ssd1306_WriteString(bpm, Font_16x26, White); // validní - hodnota
    }
    ssd1306_UpdateScreen(); //zápis na displej

    graph_time = ticks;
}
}

/* USER CODE END 3 */
}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void) {
    RCC_OscInitTypeDef RCC_OscInitStruct = { 0 };
    RCC_ClkInitTypeDef RCC_ClkInitStruct = { 0 };

    /** Configure the main internal regulator output voltage
     */
    HAL_PWREx_ControlVoltageScaling(PWR_REGULATOR_VOLTAGE_SCALE1);

    /** Initializes the RCC Oscillators according to the specified parameters
     * in the RCC_OscInitTypeDef structure.
     */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
    RCC_OscInitStruct.HSISState = RCC_HSI_ON;
    RCC_OscInitStruct.HSIDiv = RCC_HSI_DIV1;
    RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
    RCC_OscInitStruct.PLL.PLLM = RCC_PLLM_DIV1;
    RCC_OscInitStruct.PLL.PLLN = 8;
    RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
    RCC_OscInitStruct.PLL.PLLQ = RCC_PLLQ_DIV2;
    RCC_OscInitStruct.PLL.PLLR = RCC_PLLR_DIV2;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK) {
        Error_Handler();
    }

    /** Initializes the CPU, AHB and APB buses clocks
     */
    RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK | RCC_CLOCKTYPE_SYSCLK
        | RCC_CLOCKTYPE_PCLK1;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV4;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;

    if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_0) != HAL_OK) {
        Error_Handler();
    }
}

/**
 * @brief ADC1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_ADC1_Init(void) {

```

```

    /* USER CODE BEGIN ADC1_Init 0 */

    /* USER CODE END ADC1_Init 0 */

    ADC_ChannelConfTypeDef sConfig = { 0 };

    /* USER CODE BEGIN ADC1_Init 1 */

    /* USER CODE END ADC1_Init 1 */

    /** Configure the global features of the ADC (Clock, Resolution, Data Alignment and number
    of conversion)
    */
    hadc1.Instance = ADC1;
    hadc1.Init.ClockPrescaler = ADC_CLOCK_ASYNC_DIV1;
    hadc1.Init.Resolution = ADC_RESOLUTION_12B;
    hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
    hadc1.Init.ScanConvMode = ADC_SCAN_SEQ_FIXED;
    hadc1.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
    hadc1.Init.LowPowerAutoWait = DISABLE;
    hadc1.Init.LowPowerAutoPowerOff = DISABLE;
    hadc1.Init.ContinuousConvMode = DISABLE;
    hadc1.Init.NbrOfConversion = 1;
    hadc1.Init.DiscontinuousConvMode = DISABLE;
    hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
    hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
    hadc1.Init.DMAContinuousRequests = DISABLE;
    hadc1.Init.Overrun = ADC_OVR_DATA_PRESERVED;
    hadc1.Init.SamplingTimeCommon1 = ADC_SAMPLETIME_12CYCLES_5;
    hadc1.Init.OversamplingMode = DISABLE;
    hadc1.Init.TriggerFrequencyMode = ADC_TRIGGER_FREQ_HIGH;
    if (HAL_ADC_Init(&hadc1) != HAL_OK) {
        Error_Handler();
    }

    /** Configure Regular Channel
    */
    sConfig.Channel = ADC_CHANNEL_16;
    sConfig.Rank = ADC_RANK_CHANNEL_NUMBER;
    if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK) {
        Error_Handler();
    }
    /* USER CODE BEGIN ADC1_Init 2 */

    /* USER CODE END ADC1_Init 2 */

}

/**
 * @brief I2C1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_I2C1_Init(void) {

    /* USER CODE BEGIN I2C1_Init 0 */

    /* USER CODE END I2C1_Init 0 */

    /* USER CODE BEGIN I2C1_Init 1 */

    /* USER CODE END I2C1_Init 1 */
    hi2c1.Instance = I2C1;
    hi2c1.Init.Timing = 0x00303D5B;
    hi2c1.Init.OwnAddress1 = 0;
    hi2c1.Init.AddressingMode = I2C_ADDRESSINGMODE_7BIT;
    hi2c1.Init.DualAddressMode = I2C_DUALADDRESS_DISABLE;
    hi2c1.Init.OwnAddress2 = 0;
    hi2c1.Init.OwnAddress2Masks = I2C_OA2_NOMASK;
    hi2c1.Init.GeneralCallMode = I2C_GENERALCALL_DISABLE;
    hi2c1.Init.NoStretchMode = I2C_NOSTRETCH_DISABLE;
    if (HAL_I2C_Init(&hi2c1) != HAL_OK) {
        Error_Handler();
    }
}

```

```

    /** Configure Analogue filter
    */
    if (HAL_I2CEx_ConfigAnalogFilter(&hi2c1, I2C_ANALOGFILTER_ENABLE)
        != HAL_OK) {
        Error_Handler();
    }

    /** Configure Digital filter
    */
    if (HAL_I2CEx_ConfigDigitalFilter(&hi2c1, 0) != HAL_OK) {
        Error_Handler();
    }
    /* USER CODE BEGIN I2C1_Init 2 */

    /* USER CODE END I2C1_Init 2 */

}

/**
 * @brief TIM14 Initialization Function
 * @param None
 * @retval None
 */
static void MX_TIM14_Init(void) {

    /* USER CODE BEGIN TIM14_Init 0 */

    /* USER CODE END TIM14_Init 0 */

    /* USER CODE BEGIN TIM14_Init 1 */

    /* USER CODE END TIM14_Init 1 */
    htim14.Instance = TIM14;
    htim14.Init.Prescaler = 16 - 1;
    htim14.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim14.Init.Period = 1000 - 1;
    htim14.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim14.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_ENABLE;
    if (HAL_TIM_Base_Init(&htim14) != HAL_OK) {
        Error_Handler();
    }
    /* USER CODE BEGIN TIM14_Init 2 */

    /* USER CODE END TIM14_Init 2 */

}

/**
 * @brief TIM16 Initialization Function
 * @param None
 * @retval None
 */
static void MX_TIM16_Init(void) {

    /* USER CODE BEGIN TIM16_Init 0 */

    /* USER CODE END TIM16_Init 0 */

    /* USER CODE BEGIN TIM16_Init 1 */

    /* USER CODE END TIM16_Init 1 */
    htim16.Instance = TIM16;
    htim16.Init.Prescaler = 16 - 1;
    htim16.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim16.Init.Period = 12000 - 1;
    htim16.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim16.Init.RepetitionCounter = 0;
    htim16.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_ENABLE;
    if (HAL_TIM_Base_Init(&htim16) != HAL_OK) {
        Error_Handler();
    }
    /* USER CODE BEGIN TIM16_Init 2 */

```

```

        /* USER CODE END TIM16_Init 2 */
    }

    /**
     * @brief USART2 Initialization Function
     * @param None
     * @retval None
     */
    static void MX_USART2_UART_Init(void) {

        /* USER CODE BEGIN USART2_Init 0 */

        /* USER CODE END USART2_Init 0 */

        /* USER CODE BEGIN USART2_Init 1 */

        /* USER CODE END USART2_Init 1 */
        huart2.Instance = USART2;
        huart2.Init.BaudRate = 115200;
        huart2.Init.WordLength = UART_WORDLENGTH_8B;
        huart2.Init.StopBits = UART_STOPBITS_1;
        huart2.Init.Parity = UART_PARITY_NONE;
        huart2.Init.Mode = UART_MODE_TX_RX;
        huart2.Init.HwFlowCtl = UART_HWCONTROL_NONE;
        huart2.Init.OverSampling = UART_OVERSAMPLING_16;
        huart2.Init.OneBitSampling = UART_ONE_BIT_SAMPLE_DISABLE;
        huart2.Init.ClockPrescaler = UART_PRESCALER_DIV1;
        huart2.AdvancedInit.AdvFeatureInit = UART_ADVFEATURE_NO_INIT;
        if (HAL_HalfDuplex_Init(&huart2) != HAL_OK) {
            Error_Handler();
        }
        /* USER CODE BEGIN USART2_Init 2 */

        /* USER CODE END USART2_Init 2 */

    }

    /**
     * @brief GPIO Initialization Function
     * @param None
     * @retval None
     */
    static void MX_GPIO_Init(void) {
        GPIO_InitTypeDef GPIO_InitStruct = { 0 };

        /* GPIO Ports Clock Enable */
        __HAL_RCC_GPIOC_CLK_ENABLE();
        __HAL_RCC_GPIOF_CLK_ENABLE();
        __HAL_RCC_GPIOA_CLK_ENABLE();
        __HAL_RCC_GPIOB_CLK_ENABLE();

        /*Configure GPIO pin Output Level */
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0 | GPIO_PIN_1, GPIO_PIN_RESET);

        /*Configure GPIO pin Output Level */
        HAL_GPIO_WritePin(LD3_GPIO_Port, LD3_Pin, GPIO_PIN_RESET);

        /*Configure GPIO pin : T_NRST_Pin */
        GPIO_InitStruct.Pin = T_NRST_Pin;
        GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
        GPIO_InitStruct.Pull = GPIO_NOPULL;
        HAL_GPIO_Init(T_NRST_GPIO_Port, &GPIO_InitStruct);

        /*Configure GPIO pins : PB0 PB1 */
        GPIO_InitStruct.Pin = GPIO_PIN_0 | GPIO_PIN_1;
        GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
        GPIO_InitStruct.Pull = GPIO_NOPULL;
        GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
        HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

        /*Configure GPIO pin : LD3_Pin */
        GPIO_InitStruct.Pin = LD3_Pin;
        GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;

```



```

GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(LD3_GPIO_Port, &GPIO_InitStruct);
}

/* USER CODE BEGIN 4 */

void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc) {
    if (a < 6) { // zápis do prvního pole (pole A)
        d_A_ready = 0; //zajištění aby se z pole A nedalo číst v průběhu zápisu do pole A
        if (a == 0) { //když probíhá zápis do pole A
            ADCValue = (d_B[0]+d_B[1]+d_B[2]+d_B[3]+d_B[4]+d_B[5])/ 6; //průměr B
            d_B_ready = 1; //hodnota pole B připravena ke čtení
        }
        d_A[a] = HAL_ADC_GetValue(&hadc1); //načítání hodnoty z ADC do pole A
        a++;
    } else { // zápis do druhého pole (pole B) je-li pole A naplněné
        d_B_ready = 0; //zajištění aby se z pole B nedalo číst v průběhu zápisu do pole B
        if (a == 6) { //když probíhá zápis do pole B
            ADCValue = (d_A[0]+d_A[1]+d_A[2]+d_A[3]+d_A[4]+d_A[5])/6; //průměr A
            d_A_ready = 1; //hodnota pole A připravena ke čtení
        }
        d_B[a - 6] = HAL_ADC_GetValue(&hadc1); //načítání hodnoty z ADC do pole B
        a++;
        if (a > 12 - 1) { //reset po naplnění pole B
            a = 0;
        }
    }
}

}

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) {
    if (htim == &htim14) { //pro jistotu kontrola zda je zaznamenán správný timer
        HAL_ADC_Start_IT(&hadc1); //volání ADC_IT
    }
    if (htim == &htim16) { //pro jistotu kontrola zda je zaznamenán správný timer

        /* vypočet pulzu v6.2*/
        if (d_A_ready == 1 || d_B_ready == 1) {

            /*výpočet průměrné hodnoty ADC*/
            if (ticks - test_time > 2000) { //začne 2s po přiložení prstu
                if (ADCValue > 1000) { //zamezení zkreslení vlivem pohybu
                    up = 400;
                } else {
                    up = ADCValue;
                }
                e++; //jmenovatel
                value = value + up; //výpočet čitatele
                ADC_avg = value / e; //výpočet průměrné hodnoty
                if (value > 4200000000) { //ochrana před přetečením
                    e = 0;
                    value = 0;
                }
            }

            /* plnění pomocného pole předchozími hodnotami */
            helpp[0] = helpp[1];
            helpp[1] = helpp[2];
            helpp[2] = helpp[3];
            helpp[3] = ADCValue;

            if (helpp[3] > 2000) { //restart hodnot při přiložení odebrání prstu
                e = 0; //jmenovatel při výpočtu průměru
                value = 0; //čítatel při výpočtu průměru
                test_time = ticks; //čas přiložení(odebrání) prstu
            }

            if (ticks - test_time > 5000) { //výpočet začne 5s po přiložení prstu
                //peak = (helpp[0] + helpp[1] + helpp[2] + helpp[3]) / 4;
                peak = (helpp[2] + helpp[3]) / 2;
            }
        }
    }
}

```

```

/*algoritmus pro výpočet tepu*/
if (peak < ADC_avg) { //lokální průměr menší než celkový - propad
    vrchol = 0;
}

if (peak > ADC_avg && vrchol == 0) {
    vrchol = 1;
    g++;
    if (g == 1) { //první vrchol
        ticks1 = HAL_GetTick();
    }

    if (g == 5) { //pátý vrchol
        ticks2 = HAL_GetTick();
        period = ticks2 - ticks1; // času mezi 5 vrcholy
        pulsew = period / 4; // šířky jednoho pulzu
        //tpp[r] = 60000 / pulsew; // výpočet tepu
        //tep = (tpp[0]+tpp[1]+tpp[2])/3; //průměrování tepu
        tep = 60000 / pulsew;
        //r++;
        //if (r == 3) {
        //    r = 0;
        //}
        ticks1 = 0;
        ticks2 = 0;
        g = 0;
    }
}
}
}
}
}

}

/* USER CODE END 4 */

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void) {
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error return state */
    __disable_irq();
    while (1) {
    }
    /* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source line number
 * where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line number,
    ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
    /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

```

Nejsem autorem použitých knihoven pro práci s OLED displejem. Autorem knihoven je uživatel afiskon. Knihovny jsou k dispozici na stránkách GitHub.

Odkaz: <https://github.com/afiskon/stm32-ssd1306>

Testování:

Test 1:

Tabulka 2 Naměřené hodnoty pro první test

Prototyp Čas[s]	CONTEC CMS50D Čas[s]	Samsung Galaxy Watch Čas[s]	
		Aktuální	Kontinuální
14	8	10	30
12	7	9	27
11	10	11	27
14	11	10	28
13	9	9	26
14	9	8	27
13	7	9	27
14	9	11	28
15	8	11	27
11	9	7	26
Průměrný čas [s]			
13,1	8,7	9,5	27,3

Test 2 a 3:

Tabulka 3 Naměřené hodnoty pro druhý a třetí test

klidové BPM		
50	51	50
50	50	50
50	50	49
50	50	50
50	50	51
51	50	51
51	50	51
51	50	51
51	50	50
51	50	51
54	51	51
54	51	52
54	51	52
54	51	52
54	51	52
50	51	52
50	51	52
50	51	51
50	51	51
51	51	51
51	51	51
51	51	51
51	51	51
51	51	51
51	51	53
54	51	54
54	51	53
54	51	53
54	50	52
50	51	51
50	51	51
50	51	51
51	51	51
50	51	51
50	51	50
50	51	51
50	52	51
50	52	51
53	52	51
53	51	52

BPM po pohybu		
118	115	114
118	113	114
111	113	113
111	111	113
111	112	112
108	111	112
107	110	112
107	110	111
107	109	111
105	108	110
105	107	110
105	107	109
111	107	109
111	107	108
112	108	108
112	108	108
112	108	108
104	109	108
104	109	108
104	109	108
107	109	109
107	109	109
107	108	109
107	108	109
107	108	109
107	108	109
107	108	109
107	108	109
109	108	109
109	108	108
104	108	108
104	107	108
104	107	109
105	106	108
105	106	107
105	105	107
99	105	107
99	106	106
99	105	106
105	107	106
105	108	106
105	109	107

53	52	52		114	109	107
53	52	52		114	110	108
53	52	53		108	111	109
54	52	53		108	110	109
54	52	53		108	110	110
54	52	53		108	110	111
54	52	53		108	109	110
54	52	53		108	108	110
54	52	53		101	107	109
54	52	54		101	107	109
54	52	55		106	106	108
54	52	56		106	105	108
54	52	55		106	105	106
54	52	55		112	105	106
54	52	53		112	105	106
51	52	52		112	105	106
49	51	51		104	106	105
49	50	51		104	106	105
49	51	50		104	105	106
50	51	50		104	105	106
3109	3068	3107	suma	6430	6480	6516
51,81667	51,13333	51,78333	průměr	107,1667	108	108,6