

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ



DIPLOMOVÁ PRÁCE

**Eliminace pohybových artefaktů z
fotopletysmografického signálu**

Praha, 2014

Autor: Jan Horký

Autor: Bc. Jan Horký
Vedoucí: Ing. Matouš Pokorný
Biomedical Electronics Group, Katedra teorie obvodů
Fakulta elektrotechnická, ČVUT v Praze
Oponent: Ing. Jan Novotný, Ph.D.
Rok: 2014

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 9.5.2014

podpis

Poděkování

Chtěl bych zde poděkovat vedoucímu diplomové práce Ing. Matouši Pokornému za odborné vedení a pomoc při řešení diplomové práce, dále pak rodině a kamarádům za podporu.

Abstrakt

Diplomová práce se zabývá návrhem a implementací programových knihoven pro EvoPrimer sloužících k eliminaci pohybových artefaktů z fotoplethysmografického signálu. Signál zbavený pohybových artefaktů je následně použit k výpočtu srdeční frekvence při dlouhodobém monitorování srdeční činnosti. Knihovny jsou zaměřené na výpočet vlnkové transformace, použití akcelerometru a na rychlý záznam dat na micro SD kartu.

Klíčová slova

Fotoplethysmografie, PPG, srdeční frekvence, akcelerometr, vlnková transformace, pohybové artefakty, STM32, dlouhodobé monitorování

Abstract

The objective of this diploma thesis is to design and implement programming libraries for EvoPrimer designed to eliminate motion artifacts from photoplethysmographic signal. After the elimination of these artifacts, the signal is used for calculation of heart rate during long-term monitoring of cardiac activity. The libraries are focused on the calculation of wavelet transform, on use of accelerometer and on fast write operations on micro SD card.

Keywords

Photoplethysmography, PPG, heart rate, accelerometer, wavelet transform, motion artifacts, STM32, long-term monitoring

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Jan Horký**

Studijní program: Otevřená informatika (magisterský)
Obor: Počítačové inženýrství

Název tématu: **Eliminace pohybových artefaktů z fotopletyšmografického signálu**

Pokyny pro vypracování:


1. Seznamte se s fotopletyšmografickou metodou (PPG) měření srdečního tepu a problematikou eliminace pohybových artefaktů z měřených dat.
2. Realizujte software jednoduchého přenosného zařízení pro kontinuální snímání PPG signálu a pohybu související části lidského těla pomocí akcelerometru.
3. Navrhněte a na PC implementujte zpracování naměřených signálu zvyšující účinnost potlačení pohybových artefaktů. Při návrhu vyjděte z publikace [2]. Ověřte funkci a parametry implementace.
4. Diskutujte a případně implementujte algoritmy z bodu 3 přímo do software přenosného zařízení.

Seznam odborné literatury:

- [1] J. Allen, Photoplethysmography and its application in clinical physiological measurement, *Physiol. Meas.*, vol. 28, pp. 1-39, 2007.
- [2] L. Paroulek, Dlouhodobý monitoring srdečního tepu s využitím fotopletyšmografického snímače, diplomová práce, Fakulta elektrotechnická, ČVUT v Praze, Praha, 2013.
- [3] Raisonance. (2013, Nov. 25). CircleOS V4.6 Conception document [Online]. Available: http://www.stm32circle.com/resources/download.php?CircleOS_Conception.pdf

Vedoucí: Ing. Matouš Pokorný

Platnost zadání: do konce letního semestru 2014/2015


prof. Ing. Michael Šebek, DrSc.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 27. 1. 2014

Obsah

Seznam obrázků.....	13
Seznam tabulek.....	15
Seznam použitých zkratk.....	16
1 Úvod.....	17
2 Fotopletysmografie a metody pro odstranění pohybových artefaktů z PPG signálu.....	19
2.1 Úvod do fotopletysmografie.....	19
2.2 Pohybové artefakty v PPG signálu.....	21
2.3 Metody odstranění rušení z PPG signálu.....	22
2.4 Vlnková transformace.....	23
2.5 Využití vlnkové transformace k filtraci.....	24
2.6 Filtrace pomocí SWT s adaptivním nastavením prahování koeficientů.....	27
3 EvoPrimer a koncepce CircleOS.....	28
3.1 Úvod.....	28
3.2 Základní informace o vývojovém kitu EvoPrimer.....	28
3.3 Operační systém CircleOS.....	30
3.4 Vývojové prostředí Ride7 a nástroje Rkit-ARM.....	32
4 Popis funkce vyvinutých knihoven.....	33
4.1 Knihovna pro rychlý zápis na micro SD kartu.....	33
4.1.1 Úvod.....	33
4.1.2 Souborový systém FAT.....	33
4.1.3 DosFs.....	34
4.1.4 FatFs.....	34
4.1.5 Volba vhodného souborového systému.....	35
4.1.6 Optimalizace DosFs pro rychlý záznam dat.....	35
4.1.7 Výsledky optimalizace.....	37
4.2 Knihovna pro použití akcelerometru.....	38
4.2.1 Úvod.....	38
4.2.2 Popis implementace.....	38
4.3 Knihovna pro filtraci založenou na SWT.....	41
4.3.1 Úvod.....	41
4.3.2 Popis implementace.....	42
5 Uživatelská aplikace HR Monitor.....	44
5.1 Cíle uživatelské aplikace.....	44
5.2 Koncepce aplikace.....	44

5.2.1 Inicializační funkce.....	44
5.2.2 Aplikační handler.....	44
6 Ověření funkce systému a výsledky měření.....	47
6.1 Popis průběhu měření.....	47
6.2 Konfigurace aplikace.....	47
6.3 Výsledky měření.....	48
6.4 Shrnutí výsledků měření.....	51
6.5 Úprava filtrační metody.....	52
6.5.1 Zkrácení výpočtu SWT.....	52
6.5.2 Zlepšení detekce vyšších frekvencí a potlačení rušení.....	52
6.5.3 Dvoufázový výpočet srdeční frekvence.....	52
6.5.4 Shrnutí provedených úprav.....	53
6.6 Konfigurace upravené aplikace.....	54
6.7 Výsledky měření při použití upravené aplikace.....	55
7 Shrnutí výsledků testování.....	58
8 Závěr.....	59
9 Literatura.....	61
Internetové stránky.....	62
Použitý software.....	63
Příloha A.....	64
Obsah příloženého CD.....	64

Seznam obrázků

1.1	Průběh EKG signálu	1
1.2	Připojení elektrod	1
1.3	EKG hrudní pás	1
2.1	Porovnání EKG a PPG křivek	3
2.2	Transmisní a reflexní senzor	4
2.3	PPG senzor do ucha	4
2.4	PPG křivka se dvěma vrcholy	4
2.5	PPG křivka s jedním vrcholem	4
2.6	Nezarušené PPG	5
2.7	PPG s pohybovým artefaktem	5
2.8	PPG s artefakty způsobenými chůzí	5
2.9	PPG s artefakty způsobenými během	6
2.10	PPG s vysokofrekvenčním šumem	6
2.11	Schéma vlnkové transformace	7
2.12	Vlnka Mexican hat a Morletova vlnka	8
2.13	PPG signál zarušený pohybovými artefakty	8
2.14	Koeficient swa1 PPG signálu	9
2.15	Koeficient swa3	9
2.16	Koeficient swa5	9
2.17	Koeficient swa8	9
2.18	PPG signál po odstranění artefaktů	10
2.19	Princip prahování signálu	10
3.1	Hlavní části vývojového kitu EvoPrimer	13
3.2	Procesorový modul STM32F103VET6	13
3.3	Inicializace handlerů v CircleOS	14

3.4	Organizace paměti typu FLASH	15
3.5	Organizace paměti typu ROM	15
3.6	Organizace operační paměti RAM	15
3.7	Komponenty vývojového prostředí Ride7	16
4.1	Struktura FAT32	18
4.2	Část tabulky FAT32	18
4.3	FAT před vytvořením souboru	20
4.4	FAT po vytvoření souboru	20
4.5	FAT po uzavření souboru	20
4.6	Neúspěšné založení souboru	20
4.7	Data z akcelerometru při chůzi	24
4.8	Frekvenční analýza dat z akcelerometru naměřených při chůzi	24
4.9	Data z akcelerometru při běhu	24
4.10	Frekvenční analýza dat z akcelerometru naměřených při běhu	25
4.11	Vlnky z rodiny Daubechies	25
4.12	Postup při filtraci PPG signálu	27
5.1	Struktura aplikace HR Monitor	30
6.1	Srdeční frekvence v klidu, původní filtrace	32
6.2	Srdeční frekvence při chůzi 1, původní filtrace	33
6.3	Srdeční frekvence při chůzi 2, původní filtrace	33
6.4	Srdeční frekvence při chůzi 3, původní filtrace	34
6.5	Srdeční frekvence při běhu, původní filtrace	35
6.6	Postup při přepracované filtraci	37
6.7	Srdeční frekvence v klidu, nová filtrace	38
6.8	Srdeční frekvence při chůzi 1, nová filtrace	39
6.9	Srdeční frekvence při chůzi 2, nová filtrace	39
6.10	Srdeční frekvence při chůzi 3, nová filtrace	40
6.11	Srdeční frekvence při běhu, nová filtrace	40

Seznam tabulek

4.1	Porovnání frekvence záznamu souborových systémů	21
6.1	Nastavení prahování koeficientů	32
6.2	Nastavení upravené aplikace	38

Seznam použitých zkratek

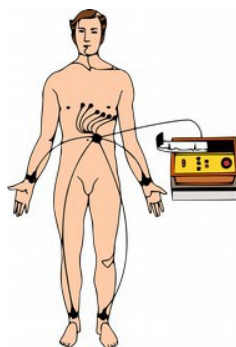
EKG	elektrokardiogram
PPG	fotopletysmografie
LED	dioda emitující světlo
AC	střídavá složka signálu
DC	stejnoseměrná složka signálu
ADC	převodník analogového signálu na digitální
ADU	analog-to-digital unit, jednotka znázorňující výstup z ADC
LMS	metoda nejmenších čtverců
STFT	krátkodobá Fourierova transformace
DWT	diskrétní vlnková transformace
SWT	stacionární vlnková transformace
ISWT	zpětná stacionární vlnková transformace
SPI	sériové periferní rozhraní
RAM	paměť s přímým přístupem
API	aplikačně-programové rozhraní
ROM	paměť pouze pro čtení
EEPROM	elektricky vymazatelná ROM
FAT	file allocation table, rodina souborových systémů, struktura tabulky v tomto systému
EOF	end-of-file, konec souboru
FPU	matematický koprocessor
DSP	digital signal processing, zpracování digitálních signálů
NVIC	nested vector interrupt controller, řadič přerušení
PLL	phase-locked loop, fázový závěs
RMS	root mean square, efektivní hodnota

1 Úvod

Monitorování srdeční aktivity patří mezi jedny z nejdůležitějších diagnostických metod v lékařství, pro záznam srdeční činnosti se v současnosti používá několik technik lišících se jak přesností měření, tak komfortem použití. Nejpoužívanější metodou je elektrokardiogram (EKG), neinvazivní metoda, která zaznamenává časovou změnu elektrického potenciálu způsobenou srdeční aktivitou. Během měření je pacient připojen k EKG pomocí elektrod umístěných obvykle na ruce a na nohu. Toto umístění elektrod se nazývá třísvodové zapojení a používá se pro běžnou diagnostiku. Další elektrody mohou být umístěny na pacientově hrudi, pak se jedná o šestisvodové zapojení. EKG se vyznačuje vysokou přesností měření, kdy je možné ze signálu vyčíst nejen srdeční frekvenci, ale i množství dalších údajů týkajících se činnosti srdce. Typický průběh jedné periody EKG křivky je zobrazen na obrázku 1.1, připojení pacienta na EKG je na obrázku 1.2. Nevýhodou takového EKG je naopak nutnost připojení pacienta pomocí elektrod, což výrazně snižuje komfort použití a znemožňuje jeho použití pro monitorování srdeční činnosti v běžném životě. Monitorováním v běžném životě zde myslíme monitorování mimo lékařská zařízení, například sledování sportovců nebo pacientů s kardiologickými problémy. Z toho důvodu se v běžném životě často používá jiná forma EKG, sestávající z pásu kolem hrudi – vysílače, a přijímače na zápěstí, který obvykle vypadá jako hodinky. EKG je podrobněji probráno například v [1].



Obrázek 1.1 – Typický průběh EKG signálu



Obrázek 1.2 – Připojení elektrod na pacienta



Obrázek 1.3 – EKG z hrudního pásu a hodinek na zápěstí

Další metodou používanou pro detekci srdečního tepu je fotopletysmografie (PPG) a právě touto metodou se budeme dále zabývat. PPG je optická metoda, při které je prosvěcována tkáň pacienta. Při prosvětlování tkáně se mění intenzita světla na základě aktuálního objemu krve v tkáni. Vzhledem k tomu, že objem krve v tkáni se mění v závislosti na srdeční činnosti, je možné na základě této změny detekovat srdeční tep. Na rozdíl od EKG stačí k snímání jen jeden lehký a malý optický senzor, nejsou potřeba žádné elektrody. Tato metoda dovoluje vysoký komfort

1 Úvod

použití v běžném životě, její nevýhodou je ale náchylnost k rušení a následnému chybnému výpočtu srdeční frekvence. Rušení je způsobeno jak pohyby pacienta, tak i například změnou osvětlení. Aby bylo možné správně detekovat srdeční frekvenci, je nutné tyto artefakty odstranit. PPG senzor se obvykle umísťuje buď na prst ruky, na ušní lalůček nebo přímo do ucha.

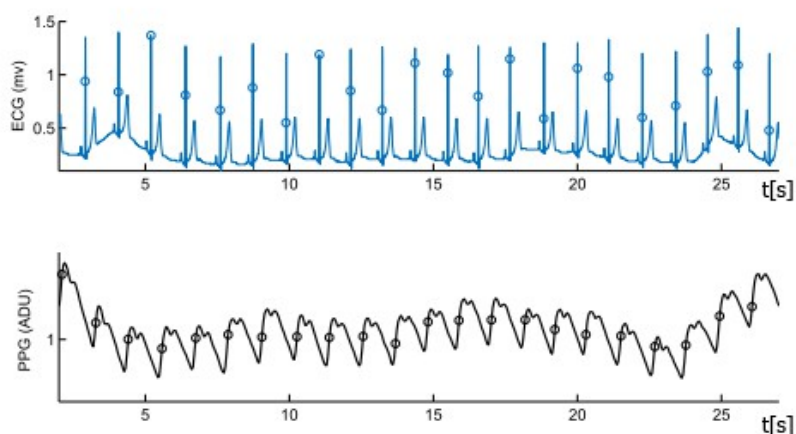
Cílem diplomové práce je navrhnout a implementovat programové knihovny, které budou sloužit k eliminaci pohybových artefaktů PPG signálu získaného z PPG senzoru umístěného v uchu při dlouhodobému monitorování srdeční činnosti pacienta. Umístění senzoru do ucha bylo zvoleno z toho důvodu, že na rozdíl od senzoru umístěného na prstu umožňuje vysoký komfort nošení pro pacienta a tato část je vždy dobře prokrvená. K tomuto účelu je použita filtrace založená na vlnkové transformaci, pomocí které potlačíme nežádoucí frekvence. Ke zjištění, které frekvence je třeba potlačit, je zaznamenávána frekvence otřesů PPG senzoru, vypočtená z dat naměřených akcelerometrem připojeným k PPG senzoru. Pro validaci správné funkce filtrace je zároveň zaznamenáván signál z EKG. Zároveň je nutné naměřená a vypočtená data ukládat na micro SD kartu s dostatečnou vzorkovací frekvencí. K tomuto účelu byla vyvinuta knihovna pro rychlý záznam na micro SD kartu. Pro vyzkoušení knihoven byla vyvinuta aplikace pro vývojový kit EvoPrimer, používající operační systém CircleOS.

Kapitola 2 se zabývá fotopletysmografií a metodami pro odstranění artefaktů z PPG signálu, v kapitole 3 je popsán vývojový kit EvoPrimer a koncepce operačního systému CircleOS. Kapitola 4 obsahuje popis knihoven pro filtraci, záznam dat z akcelerometru a rychlý záznam na micro SD kartu, kapitola 5 popisuje funkce vyvinuté aplikace. V kapitole 6 jsou pak prezentovány výsledky měření, shrnutí výsledků použití celého systému se nachází v kapitole 7.

2 Fotopletysmografie a metody pro odstranění pohybových artefaktů z PPG signálu

2.1 Úvod do fotopletysmografie

Fotopletysmografie je neinvazivní optická metoda měření změny objemu krve v dané části tkáně, použitelná pro monitorování srdeční činnosti, například srdečního tepu. Měření se provádí pomocí PPG senzoru umístěného obvykle na prstu, na ušním lalůčku nebo přímo v uchu, který sestává ze dvou hlavních součástí – LED diody a fotodiody či fototranzistoru. LED dioda je zde zdrojem světelného záření, fotodioda nebo fototranzistor jeho detektorem. S měnícím se objemem krve v prosvěcované části tkáně se lineárně mění intenzita takto detekovaného světelného záření. Dopadající světelné záření je detekováno, zesíleno, odfiltrováno a zaznamenáno jako napěťový PPG signál. PPG křivka se skládá ze dvou složek - z pulzující AC složky a pomalu se měnící DC složky s nízkou frekvencí. AC složka reprezentuje změnu objemu krve, zatímco DC složka je takzvaná absorbance – množství světla, které bylo pohlceno tkáněmi, kostmi, kůží a žilní krví. AC složka je namodulována na pomalu se měnící DC složku a její amplituda je obvykle 1 – 2% DC složky. Tvar PPG křivky závisí také na věku pacienta a na umístění senzoru.

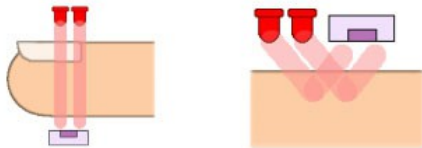


Obrázek 2.1 – Porovnání ECG a PPG křivek, obrázek převzat z [2].

K měření PPG signálu se používají dva základní typy senzorů – transmisní senzor a reflexní senzor. Transmisní senzor má LED umístěnou na jedné straně tkáně a detektor záření na druhé straně, zatímco reflexní senzor má obě části umístěny na stejné straně tkáně a využívá odrazu záření

2 Fotopletysmografie a metody pro odstranění pohybových artefaktů z PPG signálu

od tkáně, viz obrázek 2.2. U transmisních senzorů prochází světlo přes větší objem tkáně a měření je tak přesnější, reflexní senzory naopak umožňují vyšší komfort pro pacienta. Senzor musí být připevněn ke tkáni tak, aby bylo minimalizováno rušení PPG signálu, které vzniká hlavně pohybem pacienta a působením okolního světla. PPG signál je velmi náchylný na zarušení, které může být způsobeno pohybem pacienta, změnou osvětlení, pohybem samotného PPG senzoru, ale i třeba dýcháním, mluvením či polykáním. Signál se vyznačuje také tím, že je oproti EKG zpožděný.



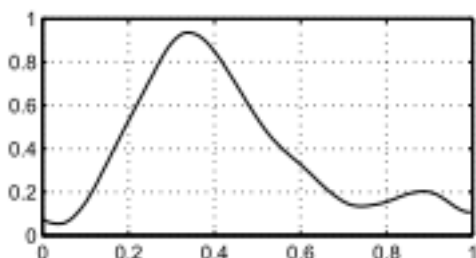
Obr 2.2 – Transmisní senzor (vlevo) a reflexní senzor (vpravo), převzato z [3]



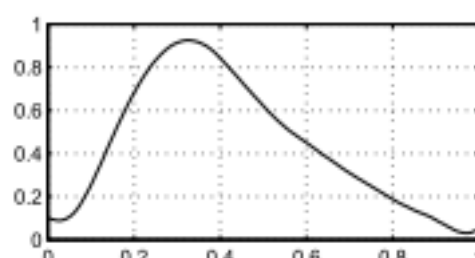
Obr 2.3 – PPG senzor určený k umístění v uchu pacienta, převzato z [4]

Jak již bylo zmíněno, PPG signál získaný ze senzoru zachycuje změnu objemu krve v dané části tkáně. Při každém srdečním stahu dochází k šíření okysličené krve a tím způsobené tlakové vlně, na PPG křivce tak vzniká vrchol značící srdeční stah. Po odrazu krve pak vzniká na křivce menší druhý vrchol, viz obrázek 2.4. Zpoždění oproti EKG je způsobeno tím, že PPG senzor je umístěn dále od srdce a určitou dobu tedy trvá, než k němu krev doputuje. Toto zpoždění bývá nižší u pacientů s vysokým krevním tlakem nebo arteriosklerózou. Zpoždění se rovněž snižuje s věkem pacienta.

Na základě tvaru PPG křivky je možné nejen detekovat tepovou frekvenci pacienta, ale také měřit okysličení krve, krevní tlak nebo monitorovat dýchání pacienta.



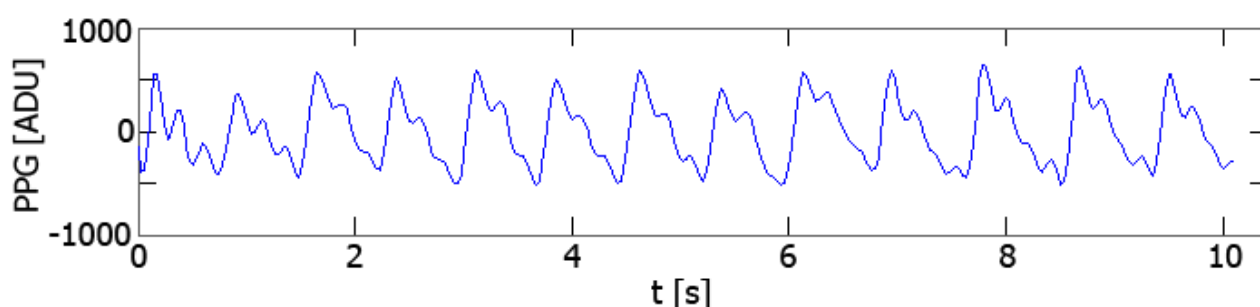
Obr 2.4 – PPG křivka, u které je patrný jak vrchol značící tlakovou vlnu, tak i vrchol značící odraženou vlnu



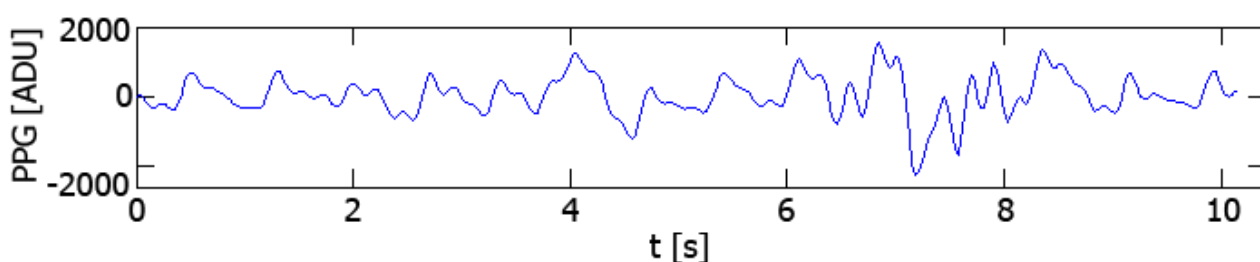
Obr 2.5 – PPG křivka, u které splynuly oba vrcholy v jeden.

2.2 Pohybové artefakty v PPG signálu

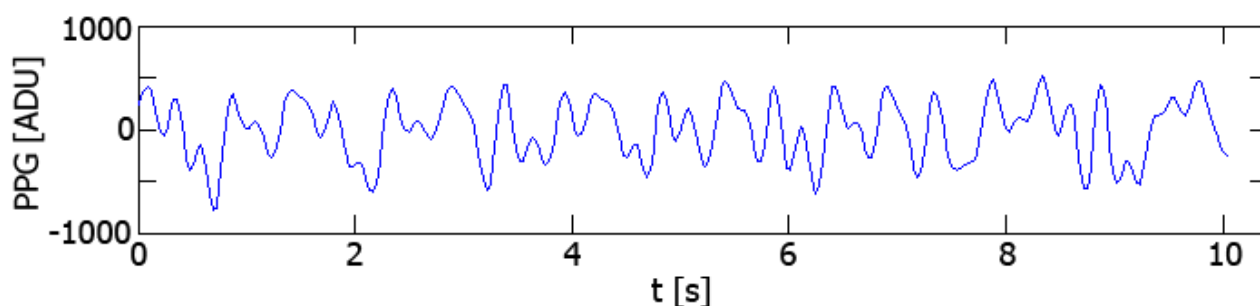
Jak bylo uvedeno, PPG signál může být snadno zarušen jak pohyby pacienta, tak i dýcháním či změnou osvětlení. Úroveň zarušení je dále ovlivněna například umístěním PPG senzoru a také jeho typem – transmisní senzory snímají PPG signál spolehlivěji než reflexní, protože u nich světlo prochází větším objemem tkáně. Nejvýraznější artefakty rušící PPG signál jsou nicméně způsobené pohybem pacienta, proto se budeme nadále zabývat pouze jimi. Na následujících obrázcích jsou zobrazeny různé varianty zarušení signálu pohybovými artefakty. Aby bylo možné detekovat srdeční tep, je nutné tyto artefakty odstranit. Způsoby odstranění artefaktů z PPG signálu se zabývá další podkapitola.



Obr 2.6 – Téměř nezarušený PPG signál ze senzoru umístěného v uchu

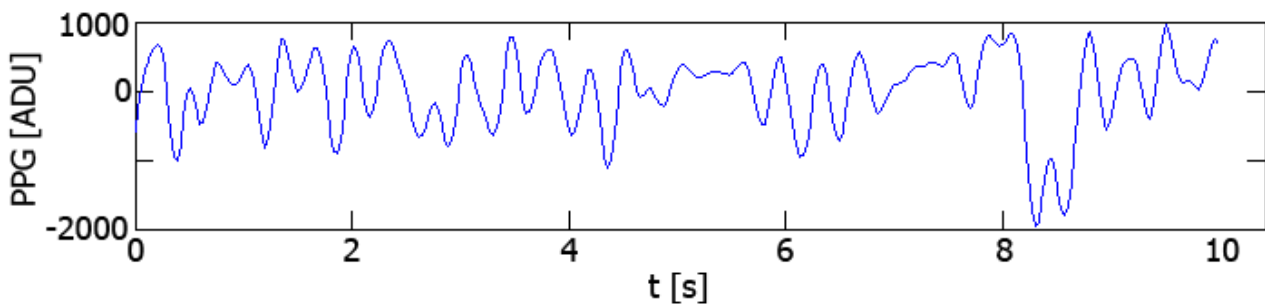


Obr 2.7 – Signál ze stejného senzoru s pohybovým artefaktem



Obr 2.8 – Signál s četnými pohybovými artefakty, které jsou způsobené chůzí pacienta.

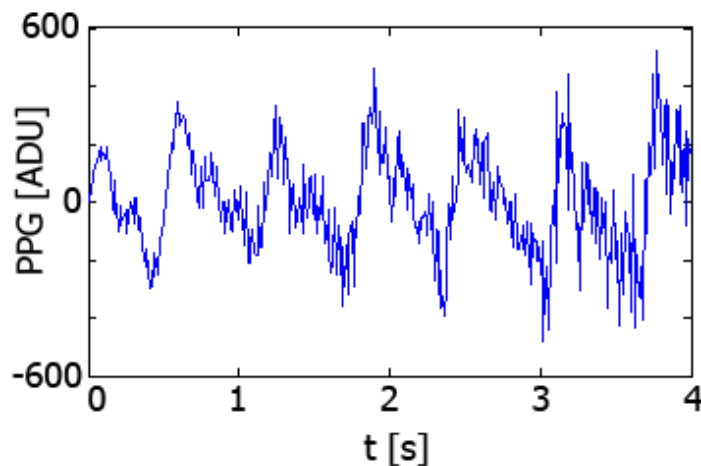
2 Fotopletysmografie a metody pro odstranění pohybových artefaktů z PPG signálu



Obr 2.9 – Signál s pohybovými artefakty způsobenými rychlou chůzí. Na první pohled zde není možné rozlišit užitečný signál od artefaktů.

2.3 Metody odstranění rušení z PPG signálu

Jak již bylo zmíněno, signál z PPG senzoru bývá zarušen, a to hned několika typy rušení. Kromě pohybových artefaktů a působení okolního světla vzniká rušení také na vodičích senzoru. Jedná se o vysokofrekvenční šum a je možné ho odstranit filtrem typu dolní propust'. Vzhledem k tomu, že frekvence vysokofrekvenčního šumu je řádově jiná než užitečné frekvence PPG signálu, nedojde tímto postupem ke znehodnocení signálu. Než se přistoupí k samotnému odstraňování pohybových artefaktů, je rovněž nutné odfiltrovat pomalu se měnící DC složku signálu. Toho dosáhneme pomocí filtru typu horní propust'. Opět nehrozí degradace signálu, protože frekvence DC složky a užitečného signálu jsou řádově rozdílné. Tímto postupem získáme PPG signál zobrazený například na obrázcích 2.6 až 2.9 a můžeme přistoupit k samotné eliminaci pohybových artefaktů a detekci srdečního tepu.

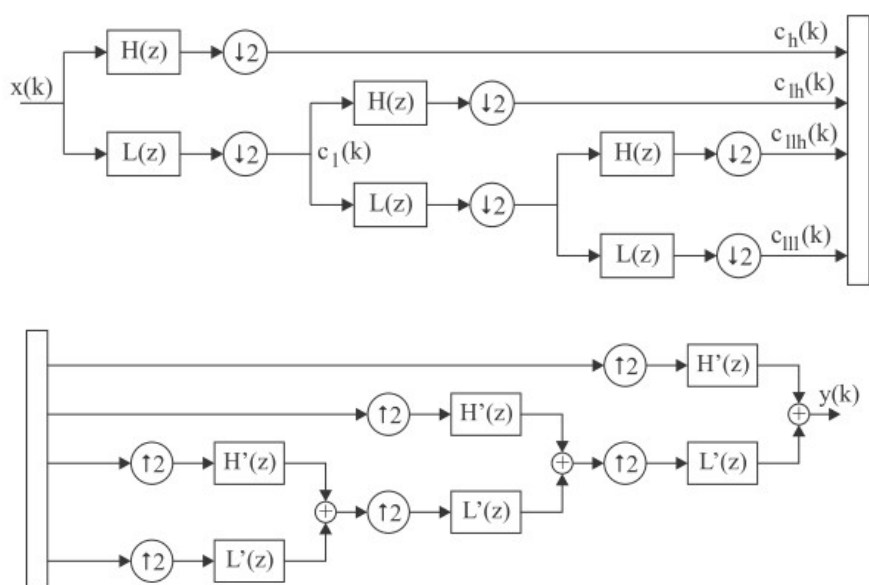


Obr 2.10 – PPG signál s vysokofrekvenčním šumem

Mezi metody sloužící k eliminaci pohybových artefaktů patří například klouzavý průměr, periodický klouzavý průměr, adaptivní filtrace LMS a filtrace založená na vlnkové transformaci. Na základě výsledků diplomové práce Lukáše Paroulka [4] byla pro odstranění artefaktů vybrána filtrace založená na vlnkové transformaci, ovšem s adaptivním prahováním koeficientů. Dále se budeme zabývat pouze touto filtrací. Ostatní filtrační metody jsou popsány v [4].

2.4 Vlnková transformace

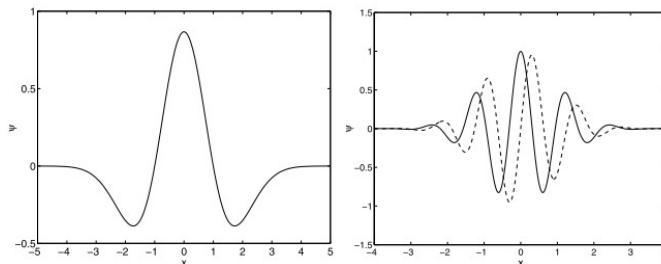
Vlnková transformace (Wavelet transform) je transformace umožňující získat popis signálu v časově-frekvenční oblasti. Výstupem vlnkové transformace je rozložení původního signálu na lineární kombinaci jednodušších signálů. Fourierova transformace převádí signál z časové oblasti do frekvenční oblasti, poskytuje nám tak informaci o tom, jaké frekvence signál obsahuje. Neříká nám ale nic o tom, kde se tyto frekvence nachází. Tento problém Fourierovy transformace je částečně odstraněn Krátkodobou Fourierovo transformací (STFT – Short Time Fourier Transform) [8], která provádí transformaci pouze nad krátkým úsekem signálu (takzvané okno). Nevýhodou STFT je konstantní šířka okna. To způsobuje, že časově široké okno má velkou rozlišitelnost ve frekvenci a malou v čase, zatímco časově úzké okno má naopak velkou rozlišitelnost v čase a malou ve frekvenci. Tyto nedostatky odstraňuje právě vlnková transformace, u které je možné vhodnou změnou šířky a tvaru okna dosáhnout optimální rozlišení v čase a frekvenci. Okno zde nazýváme „vlnka“ - odtud také pochází název transformace. Vlnku (funkci, kterou používáme k rozkladu signálu), která není posunutá ani nemá změněné měřítko nazýváme mateřskou vlnkou. Díky svým vlastnostem se vlnková transformace lépe hodí pro analýzu nestacionárních signálů, tedy takových, kde se frekvenční složení mění v čase. Pro diskrétní signály pak používáme Diskrétní vlnkovou transformaci (DWT) – zde musí být počet vzorků původního signálu roven mocnině dvou. DWT má mnoho aplikací, používá se mimo jiné například v algoritmech pro kompresi dat.



Obr 2.11 – Dopředná (nahore) a zpětná (dole) diskrétní vlnková transformace, je zde vidět použití filtrů typu horní propust' (H) a dolní propust' (L). Při každé úrovni rozkladu signálu je provedeno podvzorkování signálu faktorem 2. Celkový počet vzorků výstupních signálů $c(k)$ je tak shodný s počtem vzorků vstupního signálu $x(k)$ – samozřejmě za předpokladu, že počet vzorků vstupního signálu je roven mocnině dvou. Dekompozicí vznikají takzvané detailní koeficienty – výstupy filtrů H , a aproximační koeficienty – výstupy filtrů L . Obrázek převzat z [5]

2 Fotopletysmografie a metody pro odstranění pohybových artefaktů z PPG signálu

Existuje několik typů mateřských vlnek, například Haar, Daubechies, Biorthogonal, Gaussian, Morlet. Typ vlnky vybíráme na základě tvaru signálu s tím, že vlnka by se měla signálu co nejvíce podobat. Zvláštním typem vlnkové transformace je stacionární vlnková transformace (SWT) [9]. Ta se vyznačuje tím, že zatímco u DWT dochází při každém stupni dekompozice k podvzorkování faktorem 2, u SWT k podvzorkování nedochází. Toho je dosaženo vkládáním nul do filtrů. Z toho vyplývá, že je SWT už z principu redundantní a při dekompozici o N stupních je zde redundance stupně N . Například signál o 128 vzorcích bude při dekompozici stupně 3 rozložen na 4 signály o délce 128 – 3 detailní koeficienty a 1 aproximační koeficient. Výstup DWT by měl opět délku 128 vzorků. SWT se používá například při odstraňování šumu ze signálu. Vlnkovou transformací se blíže zabývají publikace [5], [6], [7], [20] a [21].

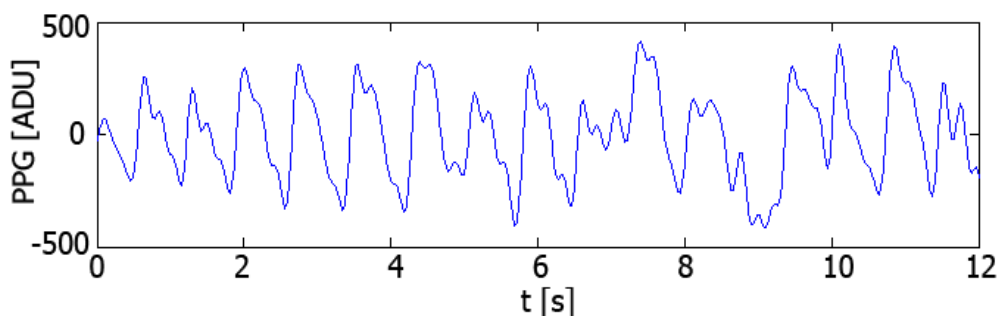


Obr 2.12 – vlevo vlnka Mexican hat, vpravo Morletova vlnka

2.5 Využití vlnkové transformace k filtraci

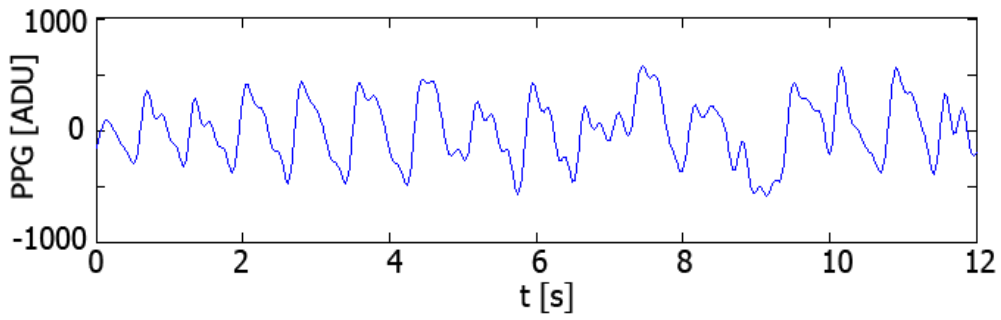
Při filtraci založené na vlnkové transformaci se postupuje tak, že je signál nejdříve rozložen na jednotlivé koeficienty (obvykle N detailních a 1 aproximační), na detailních koeficientech je následně provedena filtrace a pak následuje zpětná vlnková transformace. Úpravy detailních koeficientů mohou zahrnovat například prahování nebo použití filtru pásmová propust'. Právě pro filtraci založenou na vlnkové transformaci je nutné použít SWT, což znamená vyšší paměťovou i časovou složitost výpočtu.

Pro odstranění pohybových artefaktů v PPG signálu je filtrace založená na SWT vhodnější než potlačení nežádoucích frekvencí pomocí FFT a STFT. Problémem při použití těchto transformací je, že artefakty v PPG signálu jsou reprezentovány frekvencemi velmi blízkými frekvencím užitečného signálu, filtrace za použití FFT tak sice odstraní pohybové artefakty, s nimi ale i užitečný signál. Oproti tomu při rozkladu PPG signálu pomocí SWT je zde velká šance, že bude užitečný signál reprezentován jinou sadou koeficientů než pohybové artefakty a bude tak možné artefakty odstranit. Na následujících obrázcích je ukázán rozklad signálu pomocí SWT. Detailní koeficienty budeme označovat swd , aproximační koeficienty swa . Použitím vlnkové transformace k filtraci PPG signálu se zabývají články [4] a [10].

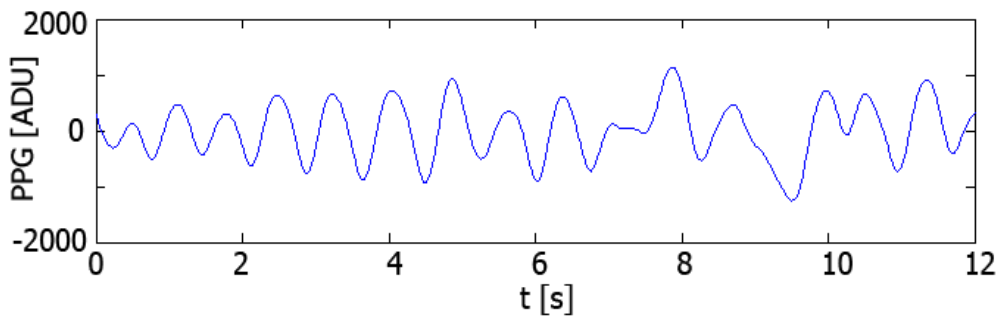


Obr 2.13 – signál zarušený pohybovými artefakty

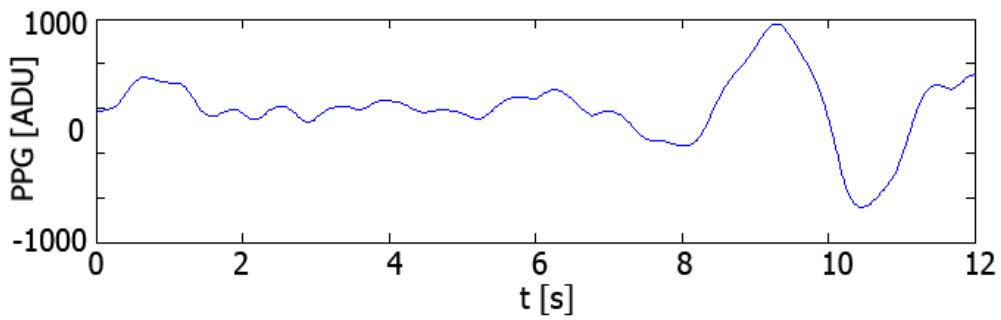
2.5 Využití vlnkové transformace k filtraci



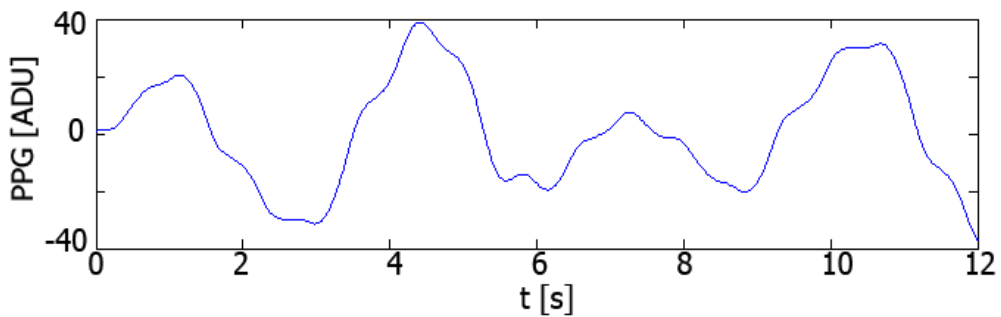
Obr 2.14 – aproximační koeficient $swa1$, téměř shodný se vstupním signálem



Obr 2.15 - $swa3$

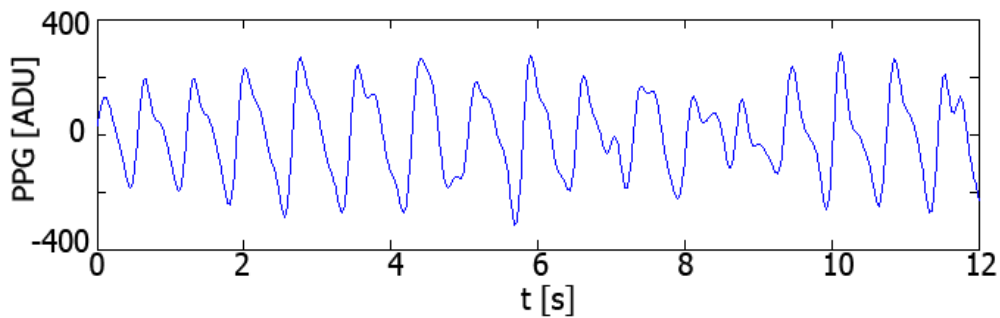


Obr 2.16 - $swa5$

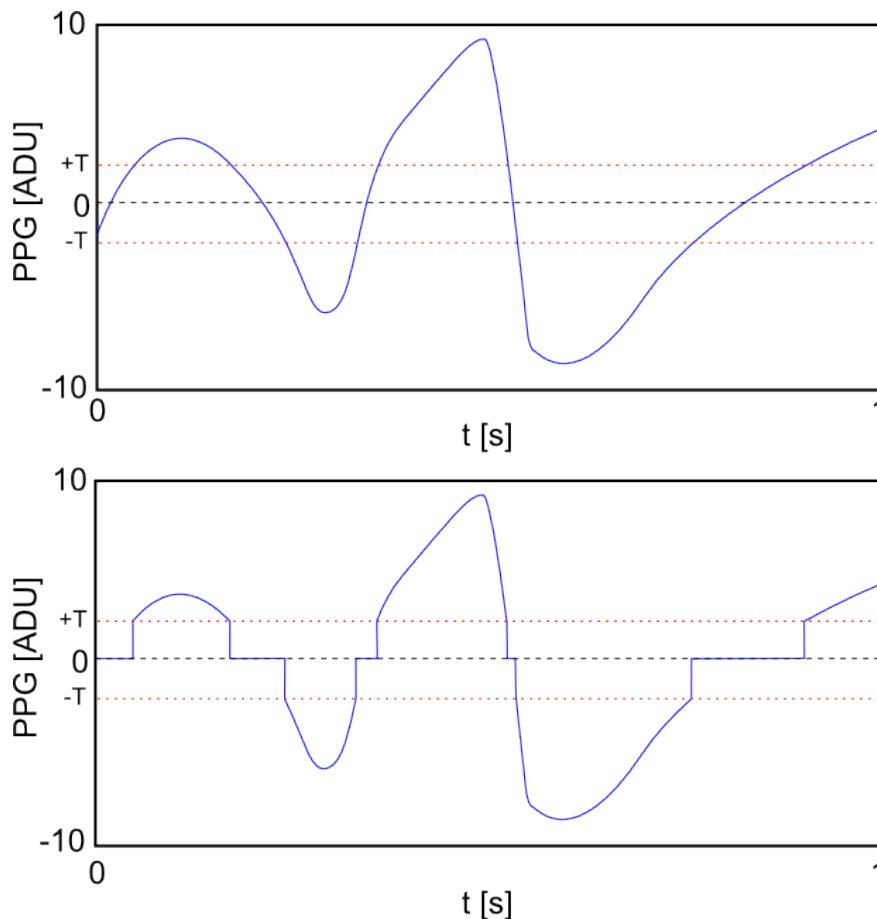


Obr 2.17 - $swa8$

2 Fotopletysmografie a metody pro odstranění pohybových artefaktů z PPG signálu



Obr 2.18 – signál po odstranění koeficientů 1,2 a 5 – 8 prahováním. Koeficienty 5 – 8 byly odstraněny, protože obsahovaly pohybové artefakty. Koeficienty 1 a 2 byly odstraněny také, protože obsahovaly neúčinný signál o vyšších frekvencích, než je frekvence srdečního tepu. Výsledkem je signál zbavený pohybových artefaktů, ve kterém je možné detekovat srdeční tep.



Obr 2.19 – Ukázka prahování signálu. Na horním obrázku je průběh signálu před provedením prahování, na dolním po prahování. Prahování zde probíhalo následujícím způsobem: Pokud byla v daném bodě absolutní hodnota signálu nižší nebo rovna hodnotě prahu T , byl zde hodnota nastavena na 0. Pokud byla hodnota vyšší než prah T , nebyla nijak změněna.

2.6 Filtrace pomocí SWT s adaptivním nastavením prahování koeficientů

V předešlé podkapitole byla popsána filtrace založená na SWT a prahování koeficientů s pohybovými artefakty. Ačkoli je takováto filtrace účinná, je nutné vždy nastavit úroveň prahování koeficientů tak, aby byly odstraněny artefakty a nedošlo ke ztrátě užitečného signálu. Nastavení úrovní prahování pro jednotlivé koeficienty se liší z toho důvodu, že jak srdeční frekvence tak i pohybové artefakty jsou reprezentovány pokaždé jinou sadou koeficientů.

Řešením tohoto problému je nastavení úrovní prahování v závislosti na nějakém dalším signálu. V našem případě je možné použít například signál z akcelerometru, ze kterého zjistíme nejvýznamnější frekvenci rušení a následně nastavíme úroveň prahování koeficientů tak, aby byla tato frekvence odstraněna. Filtrací a zpracováním dat z akcelerometru se budeme podrobně zabývat v kapitolách 4 a 5.

3 EvoPrimer a koncepce CircleOS

3.1 Úvod

Pro ověření filtračních algoritmů založených na použití SWT bylo nutné vyvinout jednoduché zařízení, které by snímalo požadované signály, ukládalo je a zároveň počítalo srdeční frekvenci na základě naměřených dat pomocí filtrace založené na SWT. Jako platforma byl zvolen vývojový kit EvoPrimer s operačním systémem CircleOS. Pro vývoj aplikace, která zajišťuje čtení a ukládání signálů a výpočet srdeční frekvence bylo zvoleno vývojové prostředí Ride7 spolu s nástroji Rkit-ARM určenými pro mikrokontroléry založené na procesorech ARM.

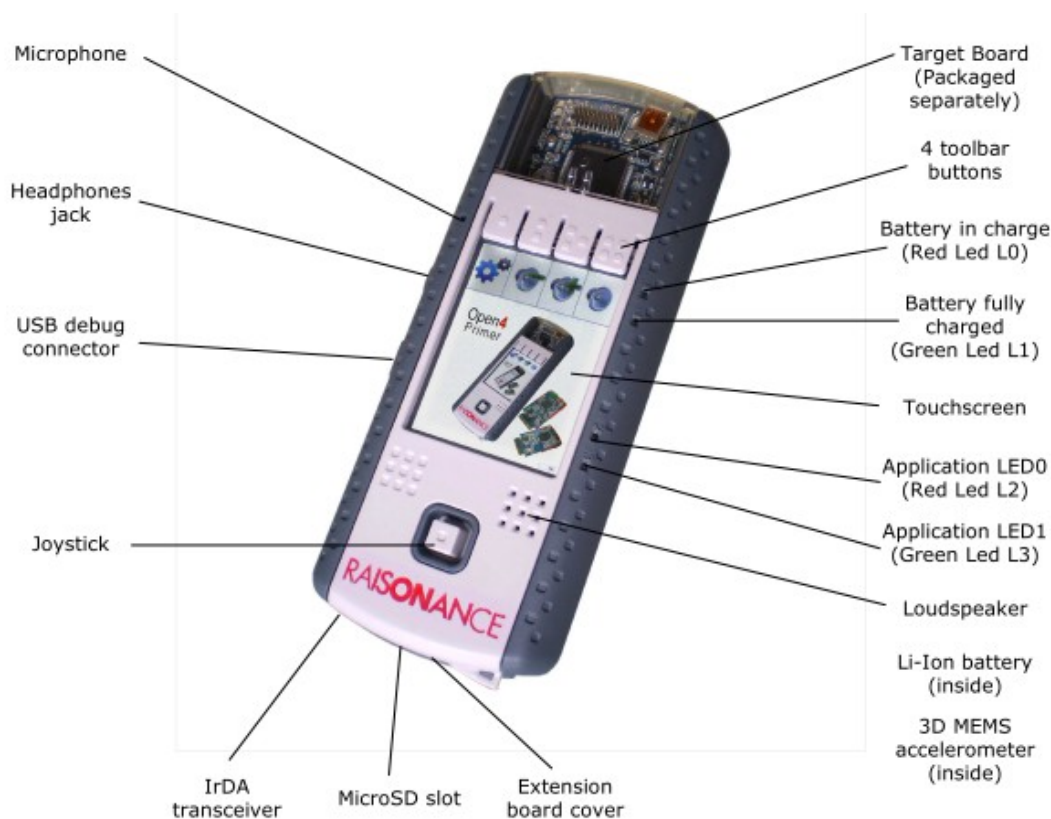
3.2 Základní informace o vývojovém kitu EvoPrimer

EvoPrimer je vývojový kit pro mikrokontroléry STM8 a STM32, vyvinutý firmou Raisonance. EvoPrimer navazuje na předchozí platformy Primer a Primer2 a skládá se ze dvou hlavních částí – ze základní desky (base platform) a z procesorového modulu (target board). Po připojení procesorového modulu k základní desce tento modul ovládá hardwarové vybavení desky. Při připojení základní desky k PC je možné nahrávat programy do paměti na procesorovém modulu.

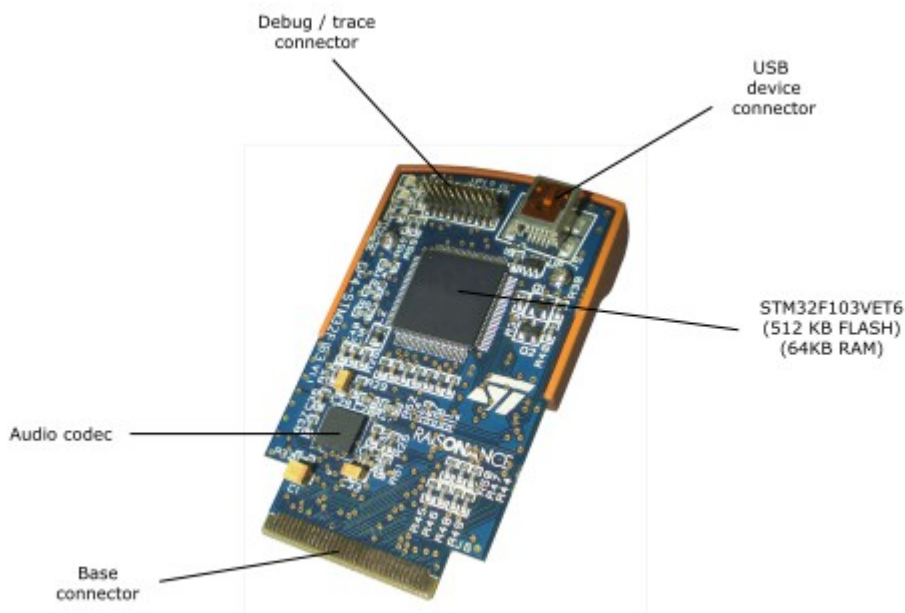
Základní deska disponuje USB rozhraním pro nahrávání a ladění programů, dotykovým barevným LCD displejem o rozlišení QVGA (320x240), joystickem fungujícím zároveň jako tlačítko, 4 samostatnými tlačítky, slotem pro micro SD kartu, akcelerometrem, Li-Ion baterií, mikrofonom a reproduktorem. Kromě toho je součástí základní desky také rozšiřující konektor sloužící pro využití rozhraní jako jsou například USART, SPI, I2C nebo ADC.

Vybavení procesorového modulu se liší v závislosti na jeho typu, vždy ale obsahuje mikrokontrolér z řady STM8 nebo STM32 a USB rozhraní pro komunikaci s PC. V našem případě byl vybrán procesorový modul STM32F103VE [11]. Tento procesorový modul disponuje mikrokontrolérem STM30F103VET6 s jádrem ARM Cortex-M3 a je vybaven 512 KB FLASH paměti a 64 KB RAM a ladícím rozhraním SWD.

3.2 Základní informace o vývojovém kitu EvoPrimer



Obr 3.1 – Popis hlavních částí vývojového kitu EvoPrimer [12]



Obr 3.2 – Procesorový modul STM32F103VET6 [11]

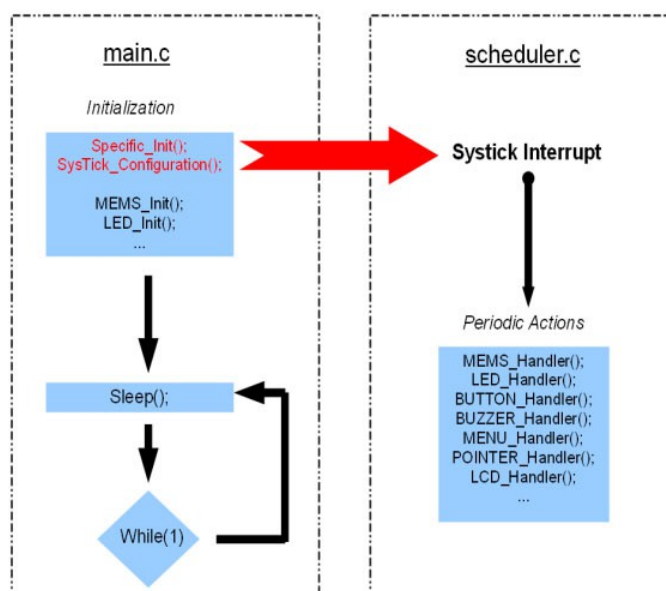
K dispozici jsou i procesorové moduly s vyšším výkonem, například STM32F407 [13] vybavený mikrokontrolérem STM32F407IGH6 s jádrem ARM Cortex-M4F s frekvencí 168MHz a FPU jednotkou. STM32F407 má také 1024 KB FLASH paměti, 192 KB RAM, 2 MB paměti typu PSRAM (pseudostatic RAM) a fotoaparát s rozlišením 1,3 Mpixel.

3.3 Operační systém CircleOS

CircleOS je jednoduchý operační systém, který poskytuje obslužné funkce (handlers) pro hardware vývojového kitu EvoPrimer, jednoduchý plánovací systém a API pro uživatelské aplikace, které zjednodušuje přístup k hardwaru.

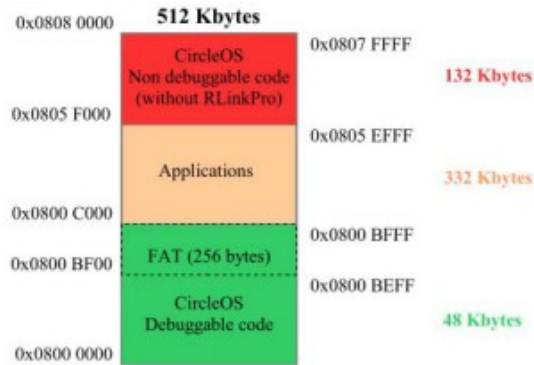
V CircleOS je funkce main (v souboru main.c) určena pouze k inicializaci zařízení a nastavení přerušení, poté už jsou volány jednotlivé handlers v tabulce handlerů. Nejvýše může být 16 handlerů, jejich volání plánuje SysTickHandler, který je volán přibližně každou milisekundu, pokud uživatel nenastaví frekvenci jeho volání jinak. Frekvence volání handlerů může být nastavena uživatelem – je možné handler volat při každém SysTicku, každých N volání Systicku, případně je možné volání daného handleru úplně zakázat.

Uživatelské aplikace mohou být linkovány s CircleOS, nahrány do EvoPrimeru a mohou používat API poskytovaná CircleOS pro virtualizaci přístupu k hardwaru.

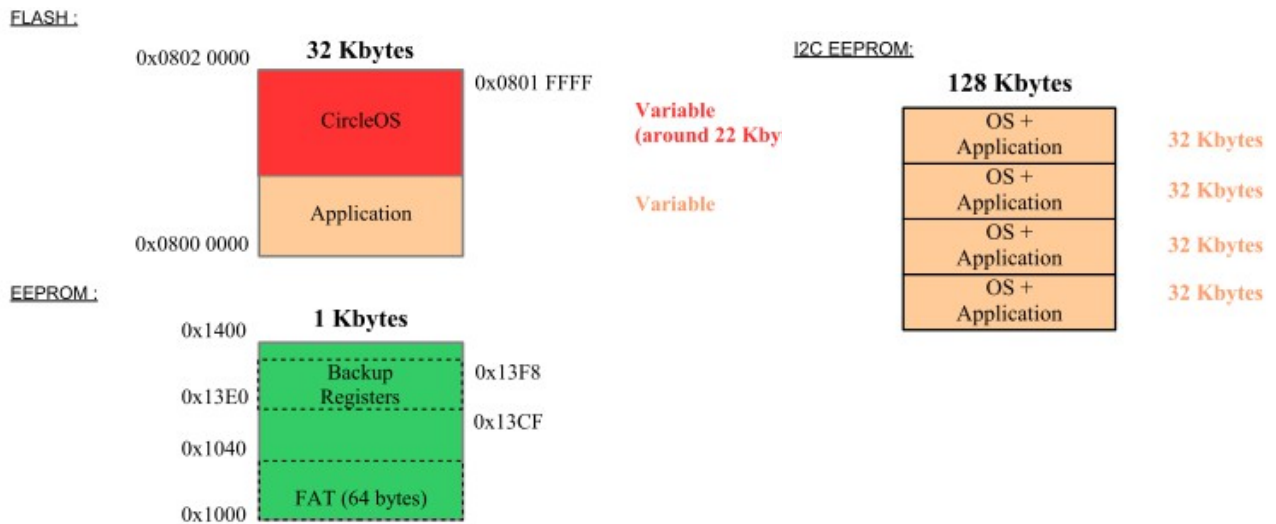


Obr 3.3 – Princip inicializace handlerů v CircleOS a jejich volání plánovačem při SysTick přerušení

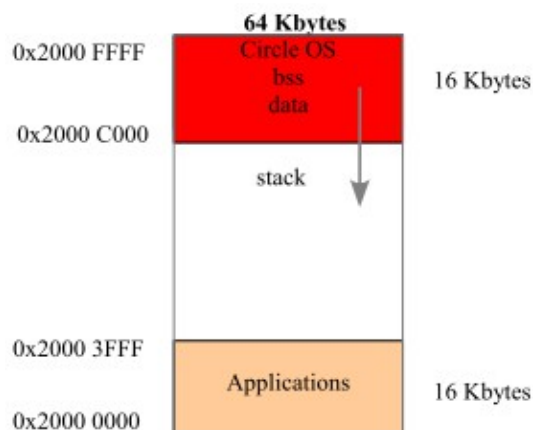
Organizace paměti závisí na použitém procesorovém modulu, moduly založené na STM32 mají paměti typu FLASH a RAM, moduly založené na STM8 mají RAM a ROM. ROM je zde rozdělena mezi paměti typu FLASH, vnitřní EEPROM a vnější I2C EEPROM. K ladění a programování slouží rozhraní Rlink, dostupné také ve verzi Rlink Pro s rozšířenými možnostmi ladění.



Obr 3.4 -Organizace paměti FLASH u typu EvoPrimer STM32F103VE. Paměť je rozdělena na dvě části pro CircleOS, přičemž jedna umožňuje ladění pouze s RlinkPro, část pro FAT tabulku a část pro uživatelské aplikace.



Obr 3.5 – Organizace paměti ROM při použití procesorových modulů založených na STM8.



Obr 3.6 - RAM u typu STM32F103VE, paměť je rozdělena mezi část pro CircleOS, část pro aplikace a část pro zásobník.

3 EvoPrimer a koncepce CircleOS

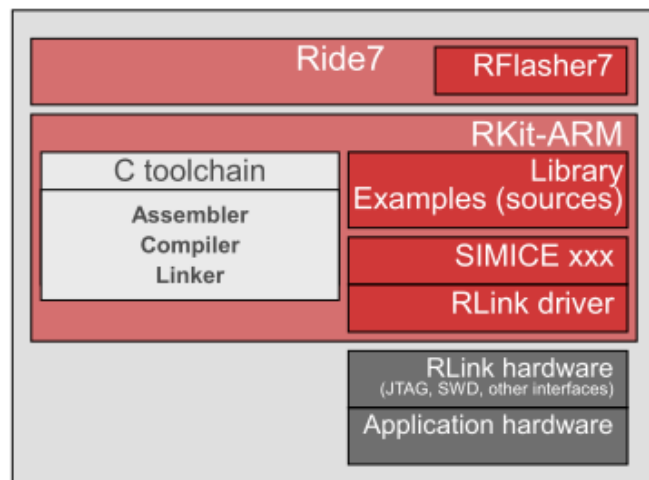
API operačního systému CircleOS poskytuje mimo jiné funkce pro použití LCD displeje, nastavení zvuku, dotykové obrazovky a tlačítek. Poskytuje také funkce pro využití souborového systému FAT32 a tedy přístup k micro SD kartě. Dřívější verze CircleOS používaly knihovnu DosFs, od verze 4.5 se ale přešlo na knihovnu FatFs. Poměrně novou součástí API je pak rozšíření Cx, sloužící k jednoduchému nastavení rozhraní, která jsou dostupná na rozšiřujícím konektoru základní desky – jedná se například o USART, SPI nebo ADC. Naopak například rozhraní I2C zde chybí a je tedy nutné používat přímo funkce knihovny STM32Fxxx. Dále API poskytuje funkce pro konfiguraci systému, PLL, plánovače a nastavení handlerů přerušení. Koncepce CircleOS je podrobně popsána v [14].

3.4 Vývojové prostředí Ride7 a nástroje Rkit-ARM

Ride7 je integrované vývojové prostředí vyvinuté firmou Raisonance, které zahrnuje editor, debugger, správce projektů a rozhraní RFlasher7 sloužící k nahrávání programů do paměti mikrokontroléru.

Rkit-ARM je sada nástrojů dostupná jako doplněk IDE Ride7 zahrnující mimo jiné kompilátor GCC pro architekturu ARM ovládaný přímo z Ride7, ladící rozhraní a simulátor procesorů architektury ARM. Sada je dostupná ve dvou licencích – Lite a Enterprise, Enterprise podporuje použití překladačů třetích stran a programování v C++.

Rlink je JTAG/SWD emulátor používající rozhraní USB, který umožňuje nahrávat a ladit uživatelské aplikace. O vývojovém prostředí i nástrojích je možné se podrobněji dočíst v [15].



Obr 3.7 – Uspořádání jednotlivých komponent vývojového prostředí

4 Popis funkce vyvinutých knihoven

4.1 Knihovna pro rychlý zápis na micro SD kartu

4.1.1 Úvod

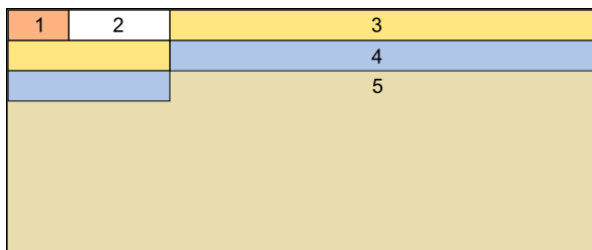
Jedním z cílů vyvíjené uživatelské aplikace mělo být kromě výpočtu srdeční frekvence také zaznamenávání naměřených a vypočtených dat na micro SD kartu, a to při frekvenci alespoň 200 zápisů za sekundu, přičemž jeden zápis má velikost do 64 bytů. Vzhledem k tomu, že operační systém CircleOS poskytuje API pro práci se soubory, byl vyzkoušen nejdříve tento způsob záznamu dat. Operační systém CircleOS používal ve starších verzích pro souborový systém knihovnu DosFs podporující souborové systémy FAT12, FAT16 a FAT32, od verze 4.5 ale používá knihovnu FatFs, podporující stejné souborové systémy.

4.1.2 Souborový systém FAT

Souborový systém FAT (File Allocation Table) [16], [17] vznikl na přelomu 70. a 80. let, vyvinutý původně pro použití na disketách. Tento souborový systém se v dnešní době už většinou nepoužívá pro harddisky, kde byl nahrazen souborovými systémy NTFS, Ext2, Ext4 a dalšími, ale stále se používá například pro paměti typu FLASH, přenosné disky a paměťové karty. Používá se také ve vestavěných systémech kde jsme omezeni výpočetním výkonem a menší kapacitou disků, protože umožňuje poměrně dobrý výkon i při jednoduché implementaci a má velmi malou režii (overhead). V dnešní době rozlišujeme 3 typy souborového systému FAT – FAT12, FAT16 a FAT32. Číslo znamená počet bitů každého záznamu v tabulce FAT a ovlivňuje nejvyšší možnou velikost oddílu. Dále se zaměříme pouze na FAT32.

Souborový systém FAT32 má následující strukturu: V prvním sektoru se nachází Volume ID, kde je uveden mimo jiné například počet bytů v sektoru (téměř vždy 512), počet sektorů v clusteru, počet rezervovaných sektorů, počet kopií FAT, velikost každé FAT v sektorech a číslo prvního clusteru kořenové složky. Samotná FAT je pole 32 bitových hodnot, kde index této hodnoty reprezentuje příslušný cluster a hodnota reprezentuje následující cluster souboru. V případě, že je daný cluster posledním clusterem souboru, je hodnota v tabulce nastavena na značku značící konec souboru – EOF. Ta bývá obvykle 0xFFFFFFFF, ale může to být jakékoli číslo vyšší než 0x8FFFFFFF.

4 Popis funkce vyvinutých knihoven



Obr 4.1 – Struktura souborového systému FAT32. Volume ID (1), rezervované sektory (2), první kopie FAT (3), druhá kopie FAT (4), datové clustery (5).

XXXXXXXXXX	XXXXXXXXXX	0xFFFFFFFF	0x00000004
0x00000006	0x0000000A	0xFFFFFFFF	0x00000000
0x00000000	0x00000000	0x0000000B	0x0000000C
0xFFFFFFFF	0x00000000	0x00000000	0x00000000
0x00000000	0x00000000	0x00000000	0x00000000

Obr 4.2 – Část tabulky souborového systému FAT32. První dva záznamy jsou nepoužité (clustery se nečíslují od 0 ale od 2), za nimi následuje záznam kořenové složky o velikosti jeden cluster (zeleně) a soubor používající clustery 3,4 a 6 (oranžově). Další soubor ve FAT je umístěn v clusterech 5,10,11,12 (žlutě).

4.1.3 DosFs

DosFs je open-source FAT-kompatibilní souborový systém určený pro vestavěné systémy. Podporuje podsložky a vyžaduje pouze jeden buffer o velikosti jednoho sektoru (512 bytů). DosFs nastavuje čas u souborů vždy na stejnou hodnotu a umožňuje pouze krátká jména souborů – 8 znaků pro samotný název, 3 znaky pro příponu. DosFs je vhodný pro nenáročné aplikace a v situacích, kdy jsme omezeni velikostí paměti, trpí ovšem častými přístupy k disku, a to jak pro čtení, tak i pro zápis. Tyto přístupy jsou způsobené tím, že DosFs nepoužívá vyrovnávací paměť pro části FAT. Tím je sice dosaženo nízké paměťové náročnosti, ovšem za cenu snížení výkonnosti. Nejvíce znatelné jsou tyto nadbytečné přístupy při nastavování nových clusterů ve FAT, kdy dochází ke čtení i zápisu při každém nastavení clusteru. Výhodou tohoto systému pak je, že není nutné soubory zavírat a nemůže tak dojít k poškození souboru – vzhledem k tomu, že neexistuje žádná vyrovnávací paměť pro FAT, je vždy vše zapsáno přímo na disk.

4.1.4 FatFs

FatFs [18] je souborový systém určený pro vestavěné systémy, který je zcela oddělený od diskové I/O vrstvy, takže nezávisí na hardwarovém vybavení. Umožňuje používat dlouhé názvy souborů a vytvářet soubory o velikosti až 4 GB. Nejvýznamnějším rozdílem oproti DosFs je, že FatFs využívá vyrovnávací paměť pro části FAT, a má tedy výrazně nižší počet přístupů k disku.

4.1.5 Volba vhodného souborového systému

Pro záznam dat byla vytvořena jednoduchá aplikace pro EvoPrimer, zapisující data periodicky na micro SD kartu a bylo testováno, zda je možné dosáhnout požadované frekvence zápisů 200 Hz. Nejprve byl testován souborový systém DosFs, který byl tou dobou používán CircleOS. S DosFs se podařilo dosáhnout frekvence záznamů zhruba 72 Hz, více zápisů za vteřinu nebylo možno dosáhnout. Důvodem pro tyto výsledky byly právě časté přístupy ke kartě, kdy na zápis jednoho sektoru mohlo připadnout až 12 operací pro čtení a zápis dohromady. Následně byl vyzkoušen systém FatFs používaný novou verzí CircleOS. U FatFs byla frekvence záznamů proměnlivá a pohybovala se mezi 70 Hz až 100 Hz. Způsobeno to bylo tím, že ačkoli ukládání částí FAT do vyrovnávací paměti celkově snížilo počet přístupů ke kartě, bylo po několika zápisech nutné tuto paměť zapsat na kartu a naplnit novými daty, což si vyžádalo další přístupy.

Vzhledem k těmto výsledkům bylo rozhodnuto vyvinout vlastní FAT32-kompatibilní knihovnu umožňující zápisy na kartu v dostatečné frekvenci. Alternativním řešením by bylo zaznamenávat data rovnou na kartu bez jakéhokoli souborového systému a vyvinout aplikaci pro PC, pomocí které by se karta četla, nevýhodou by ovšem byla nečitelnost karty pro počítače bez této aplikace.

4.1.6 Optimalizace DosFs pro rychlý záznam dat

Jako základ souborového systému byl zvolen systém DosFs a cílem optimalizací bylo odstranit nadbytečné operace čtení a zápisu a tím dosáhnout požadované frekvence záznamu 200 Hz. Tyto operace se týkaly v DosFs úprav FAT a kořenové složky, které se prováděly při každém zápisu dat. Aby se co nejvíce omezil počet operací čtení a zápisu během zaznamenávání dat, je se soubory zacházeno následujícím způsobem. Místo přístupů k FAT a kořenové složce během zápisu dat se tyto operace provedou při otevírání souboru, kdy je zároveň specifikována maximální možná velikost souboru v clusterech. Poté co je tato velikost specifikována, je ve FAT hledán dostatečně velký celistvý blok volného místa pro nový soubor. Pokud je takový blok nalezen, je upraven záznam složky a je vytvořen prázdný soubor požadované velikosti. Výsledkem je, že ačkoli jsme ještě nezapsali žádná data, máme už přednastavenou FAT i záznam složky, takže během zápisu dat do souboru do nich již nemusíme zapisovat ani je číst. Když je po provedení zápisu dat soubor uzavírán, jsou FAT a záznam složky pouze upraveny tak, aby odpovídaly skutečné velikosti souboru. Tím je soubor zkrácen tak, aby obsahoval pouze platná data. Bylo by také možné zcela vynechat počáteční nastavení FAT a složky a provést jejich nastavení až na konci při zavírání souboru. To by ale zároveň znamenalo, že při nesprávném ukončení souboru (například zamrznutí programu a následný reset, vypnutí systému) by byla ztracena veškerá data zapsaná do tohoto souboru, protože by pro soubor neexistoval záznam ve FAT. Pokud je nastavení FAT a složky provedeno předem, dojde v takovém případě pouze k tomu, že soubor není zkrácen a za platnými daty tak navazují neplatná data, nedojde ale ke ztrátě žádných dat kromě posledního obsahu bufferu o velikosti jednoho sektoru, který ještě nebyl zapsán. Tímto způsobem je dosaženo odstranění nadbytečných přístupů ke kartě během zápisu dat. Data se zapisují po sektorech, tedy každých 512 bytů se provede jedna operace zápisu.

4 Popis funkce vyvinutých knihoven

Stejně jako DosFs, i jeho optimalizovaná verze využívá pro operace s diskem buffer o velikosti jednoho sektoru, tedy 512 bytů. Při rychlém zápisu dat se postupuje tak, že se data nezapisují hned, ale pouze se ukládají do bufferu. K jejich zápisu dojde až při naplnění bufferu, tedy vždy po zapsání 512 B. Po posledním zápisu je nutné provést ještě dodatečnou operaci pro zapsání posledního sektoru, protože ne zcela zaplněný buffer ještě nebyl zapsán na disk. Stejně tak je tuto operaci nutné provést, pokud se mezi dvěma zápisy provádí disková operace, která přepíše buffer a dojde tak ke ztrátě dosud nezapsaných dat.

XXXXXXXX	XXXXXXXX	0xFFFFFFFF	0x00000000
0x00000000	0x00000000	0x00000000	0x00000000
0x00000000	0x00000000	0x00000000	0x00000000
0x00000000	0x00000000	0x00000000	0x00000000

Obr 4.3 – FAT před vytvořením souboru, obsahuje pouze záznam kořenové složky.

XXXXXXXX	XXXXXXXX	0xFFFFFFFF	0x00000004
0x00000005	0x00000006	0x00000007	0x00000008
0x00000009	0xFFFFFFFF	0x00000000	0x00000000
0x00000000	0x00000000	0x00000000	0x00000000

Obr 4.4 – FAT po vytvoření prázdného souboru o velikosti 7 clusterů.

XXXXXXXX	XXXXXXXX	0xFFFFFFFF	0x00000004
0x00000005	0x00000006	0xFFFFFFFF	0x00000000
0x00000000	0x00000000	0x00000000	0x00000000
0x00000000	0x00000000	0x00000000	0x00000000

Obr 4.5 – FAT po uzavření souboru. Protože byly zapsány pouze 4 clustery, zbylé clustery byly označeny jako volné místo a do posledního použitého záznamu clusteru byla zapsána značka EOF značící konec souboru.

XXXXXXXX	XXXXXXXX	0xFFFFFFFF	0x00000000
0x00000000	0x00000006	0x00000007	0x00000008
0x00000009	0x0000000A	0xFFFFFFFF	0x00000000
0x00000000	0x00000000	0x00000000	0x00000000

Obr 4.6 – Založení nového souboru o velikosti 7 clusterů v tomto případě selže, protože ačkoli je zde dostatek místa, netvoří volné místo celistvý blok o požadované velikosti.

4.1 Knihovna pro rychlý zápis na micro SD kartu

I když se podařilo výše zmíněnými postupy odstranit nadbytečné přístupy ke kartě, potřeba celistvého volného bloku o požadované velikosti komplikuje zakládání nových souborů a může snadno dojít k tomu, že ačkoli bude na kartě dostatek volného místa, bude fragmentováno tak, že nebude možné vytvořit výše zmíněným postupem nový soubor. Nabízí se tedy možnost uložit celou FAT do pole, takže by se při každém zápisu místo přístupu k FAT na kartě přistupovalo k její lokální kopii. Tento způsob by ovšem vyžadoval příliš velké množství paměti.

Řešením tohoto problému je bitmapa volného místa (free space bitmap), tedy pole ve kterém bit nastavený na 0 odpovídá volnému místu a bit nastavený na 1 zapsanému místu. Opět vzhledem k nedostatku paměti není možné udělat bitmapu tak rozsáhlou, aby každý bit odpovídal jednomu clusteru, ale je možné ji vytvořit tak, že jedno místo bude odpovídat určitému počtu clusterů. Pokud je daný blok clusterů celý volný, nastaví se příslušný bit na 0, v opačném případě se bit nastaví na 1. Tímto způsobem se umožní alespoň částečná fragmentace s fragmenty o minimální rovné bloku clusterů. Použitím bitmapy se nijak nezvýší počet přístupů ke kartě během zápisů a vzhledem k její relativně malé velikosti ani samotné její procházení výrazně nezvýší dobu potřebnou pro jeden záznam. Pro micro SD kartu o velikosti 4 GB s velikostí clusteru 4096 B a bitmapou o velikosti 256 B má každý blok clusterů velikost zhruba 16 MB.

4.1.7 Výsledky optimalizace

Za účelem otestování provedených optimalizací byla vyvinuta testovací aplikace pro EvoPrimer STM32F103VE s operačním systémem CircleOS, která provádí v MENU handleru zápisy na mikro SD kartu. Frekvence volání handleru byla volena co nejvyšší a bylo zjišťováno, při jaké frekvenci volání už daný souborový systém nebude schopný zaznamenat všechna data. Výsledky jsou zobrazeny v tabulce 4.1.

Souborový systém	Počet bytů při jednom zápisu	Nejvyšší dosažená frekvence zápisu	Objem dat zapsaný za sekundu v KB
DosFs	16B	72Hz	1,152KB/s
FatFs	16B	<100Hz	<1,6KB/s
Optimalizovaný DosFs	16B	2300Hz	36,8KB/s
	32B	2060Hz	65,92KB/s
	64B	1850Hz	118,4KB/s
	128B	1240Hz	158,72KB/s
	256B	590Hz	151,04KB/s
	512B	380Hz	194,56KB/s

Tabulka 4.1 – Výsledky testování souborových systémů DosFs, FatFs a optimalizovaného DosFs.

Testování bylo prováděno na vývojovém kitu EvoPrimer STM32F103VE vybaveném 512KB FLASH paměti a 64KB RAM s operačním systémem CircleOS verze 4.6.1. Výsledky testování jasně ukazují, že při zápisu dat má optimalizovaný DosFs značně vyšší výkonnost než DosFs a FatFs, frekvence zápisu několikanásobně převyšuje požadovanou mez 200 Hz. Vzhledem k tomu, že optimalizované funkce slouží především pro rychlé zápisy dat například během logování naměřených údajů, byly v této verzi zároveň zachovány původní funkce DosFs pro standardní zacházení se soubory.

4 Popis funkce vyvinutých knihoven

Další zrychlení zápisu dat snižováním počtu operací již není dále možné, protože při samotném zápisu se provádí nyní pouze jeden přístup ke kartě každých 512 bytů. Další optimalizace by se proto musely týkat ovladačů používaných pro operace s kartou. Vzhledem k tomu, že optimalizovaná knihovna splňovala zadané požadavky – tedy zápis s frekvencí alespoň 200 Hz při velikosti dat v jednom zápisu do 64 bytů, nebyly již další optimalizace provedeny.

4.2 Knihovna pro použití akcelerometru

4.2.1 Úvod

Použití akcelerometru ve vyvíjené aplikaci bylo nutné z toho důvodu, abychom získali informace o pohybu pacienta – zda je v klidu a nebo méně či více intenzivním pohybu a na základě získaných dat nastavili úroveň prahování koeficientů při filtraci založené na vlnkové transformaci. Byl vybrán akcelerometr typu LIS331DLH [19], který měří ve všech třech osách a má 16 bitový výstup. Tento akcelerometr podporuje rozhraní I2C a SPI, a to jak SPI s využitím 4 vodičů, tak SPI pouze se třemi vodiči, kdy je pro signály MISO a MOSI použit pouze jeden vodič. V této práci se používá čtyřvodičové zapojení.

4.2.2 Popis implementace

Po provedení inicializace knihovna čte hodnoty z akcelerometru a ukládá je do kruhového bufferu jako hodnoty typu float. Knihovna může být nakonfigurována v několika režimech, podle vybraného režimu jsou pak data ukládána do bufferu.

- Režim záznamu jedné osy:

Zaznamenává se pouze jedna z os x, y, z , zaznamenávání všech tří os najednou by bylo příliš paměťově náročné. Nevýhodou tohoto režimu je, že pokud je akcelerometr nevhodně orientován, může se zaznamenávaná osa lišit od osy nejvýznamnějšího rušení.

- Režim záznamu maximální výchylky:

Zaznamenávají se všechny tři osy, každá do samostatného bufferu. Při čtení zaznamenaných dat jsou vrácena data z bufferu, který má v danou chvíli nejvyšší RMS a rušení v této ose je tedy nejvýznamnější.

- Režim záznamu velikosti vektoru:

Zaznamenává se velikost vektoru složeného ze všech tří os.

- Režim gradient:

Data jsou zaznamenávána stejně jako při záznamu velikosti vektoru, ale při jejich čtení je místo velikosti vektoru vypočten gradient v daném bodě.

Při testování se nejvíce osvědčil režim záznamu maximální výchylky, protože je na rozdíl od režimu záznamu jedné osy nezávislý na orientaci akcelerometru. Režim gradient a režim velikosti vektoru se ukázaly jako nevhodné, pokud měla být se signálem prováděna frekvenční analýza.

Knihovna také umožňuje průměrovat naměřená data přes posledních N vzorků, implementace je založena na MEMS funkcích operačního systému CircleOS, které slouží pro komunikaci s vestavěným akcelerometrem.

Cílem knihovny je zjistit nejvýznamnější frekvenci obsaženou v naměřených datech. K tomuto účelu slouží funkce, která provede nad naměřenými Fourierovu transformaci a následně vyhledá index s nejvyšší absolutní hodnotou. Z indexu je pak vypočtena příslušná frekvence na základě znalosti vzorkovací frekvence dat akcelerometru. Je přitom možné specifikovat úsek prohledávaných frekvencí, například pokud hledáme frekvenci pohybu pacienta, lze očekávat že se bude pohybovat v rozmezí nula až jednotky hertzů, nemá proto smysl prohledávat celé frekvenční spektrum. Tím, že prohledáme pouze tento úsek také zamezíme tomu, abychom detekovali například frekvenci, která je sice nejvíce zastoupena, ale neodpovídá pohybu pacienta a je způsobena například vysokofrekvenčním rušením.

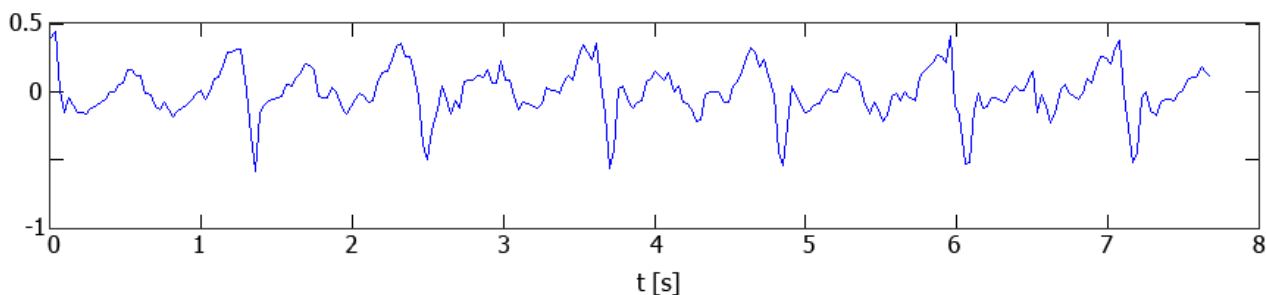
Dále je nutné ošetřit situaci, kdy je pacient v klidu a z akcelerometru tak nepřichází žádná užitečná data. Pokud by v takovém případě proběhla detekce frekvence pouze výše zmíněnou metodou, získali bychom frekvenci šumu akcelerometru, ačkoli frekvence pohybu je nulová. Toto je ošetřeno tak, že se po nalezení frekvence pomocí Fourierovy transformace ještě provede volání funkce hledající vrcholy v časové oblasti. Pokud je nalezeno dostatek vrcholů s danou minimální výškou vzdálených od sebe alespoň N vzorků, je nalezená frekvence ponechána beze změny. Pokud ovšem není nalezen dostatek vrcholů, znamená to, že se pacient nepohybuje a nalezená frekvence je nastavena na 0 Hz.

Vzhledem k tomu že například rychlá chůze a pomalejší běh mohou mít shodnou frekvenci zhruba 2Hz, bylo nutné tyto dvě situace rozlišit. Důvodem je, že ačkoli je frekvence v obou případech stejná, při běhu dochází k větším otřesům zařízení a je tedy nutné nastavit úroveň prahování koeficientů u SWT jinak. Při rozlišení těchto dvou situací bylo využito toho, že při běhu je intenzita otřesů vyšší než při chůzi. Rozlišení se provádí pomocí výpočtu efektivní hodnoty (RMS) naměřených dat. Výpočet probíhá tak, že jsou sečteny druhé mocniny všech vzorků a výsledek je vydělen počtem vzorků. Jediný rozdíl oproti efektivní hodnotě je zde v tom, že výsledek není odmocněn, protože operace pro odmocnění je výpočetně náročná. Pokud pak frekvence překročí nějakou mez a zároveň je vypočtená efektivní hodnota vyšší než definovaný práh, je situace vyhodnocena jako běh. V opačném případě se jedná o chůzi.

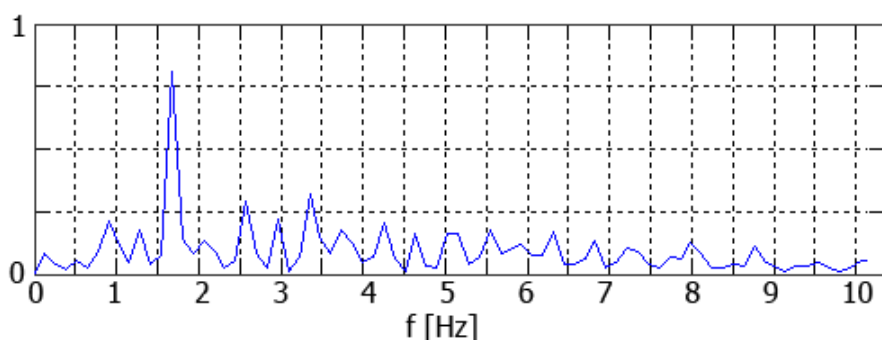
Kromě frekvenční analýzy by bylo také možné hledat frekvenci rušení v časové oblasti pomocí hledání vrcholů. Nevýhodou této metody je, že je nutné specifikovat minimální vzdálenost mezi sousedními vrcholy. Vzhledem k tomu, že vzdálenost vrcholů se mění v závislosti na frekvenci pohybu, byla vybrána metoda detekce pomocí frekvenční analýzy. Pro výpočet samotné Fourierovy transformace byly původně s úpravami převzaty funkce z open-source C++ knihovny Wavelet Library, kterou vyvinul Rafat Hussain. Vzhledem k tomu, že výpočet probíhal příliš pomalu, bylo od této implementace upuštěno a byly využity funkce DSP knihovny vyvinuté firmou STMicroelectronics. Výhodou této knihovny je, že využívá dopředu vypočítané tabulky pro goniometrické funkce a také dopředu vypočítanou tabulku pro bitovou reverzaci během Fourierovy transformace.

4 Popis funkce vyvinutých knihoven

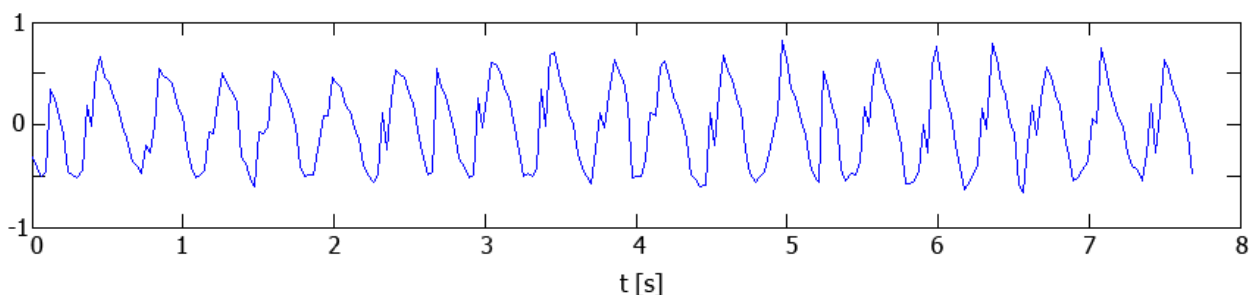
Pro snížení výpočetní náročnosti bylo zvažováno pro data z akcelerometru použít celá čísla místo čísel s plovoucí řádovou čárkou (float). Použití 32 bitových celých čísel by snížilo výpočetní složitost, použití 16 bitových čísel navíc i paměťovou náročnost. Toto řešení bylo zvažováno hlavně z toho důvodu, že procesor Cortex-M3 použitý v EvoPrimeru s procesorovým modulem STM32F103VE nemá matematický koprocessor pro výpočty v plovoucí čárce (floating-point unit - FPU) a výpočty v plovoucí řádové čárce tak probíhají pomalu. Od této myšlenky bylo ale nakonec upuštěno, protože by nastávaly problémy při použití Fourierovy transformace, kde by snadno mohlo dojít k nepřesnostem a přetečení.



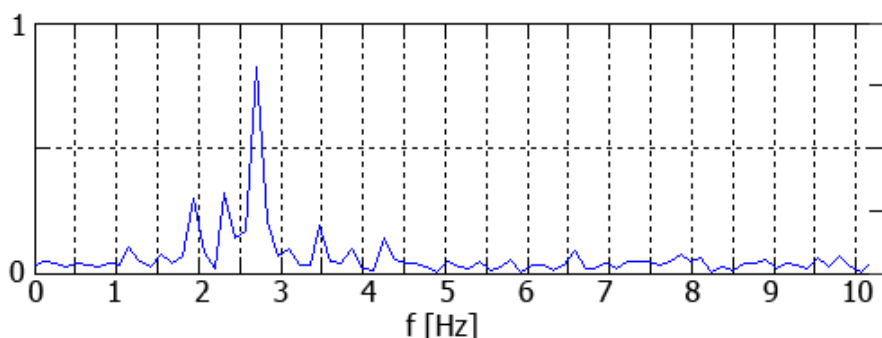
Obr 4.7 – Data naměřená akcelerometrem během chůze rychlostí 4 km za hodinu. Vzorkovací frekvence 33 Hz, délka záznamu 256 hodnot tedy odpovídá zhruba 7,8 sekundám.



Obr 4.8 Frekvenční analýza signálu z obrázku 4.7. Jasně zde vidíme vrchol u frekvence 1,6 Hz reprezentující chůzi.



Obr 4.9 – Data naměřená akcelerometrem během běhu rychlostí 8 km za hodinu.

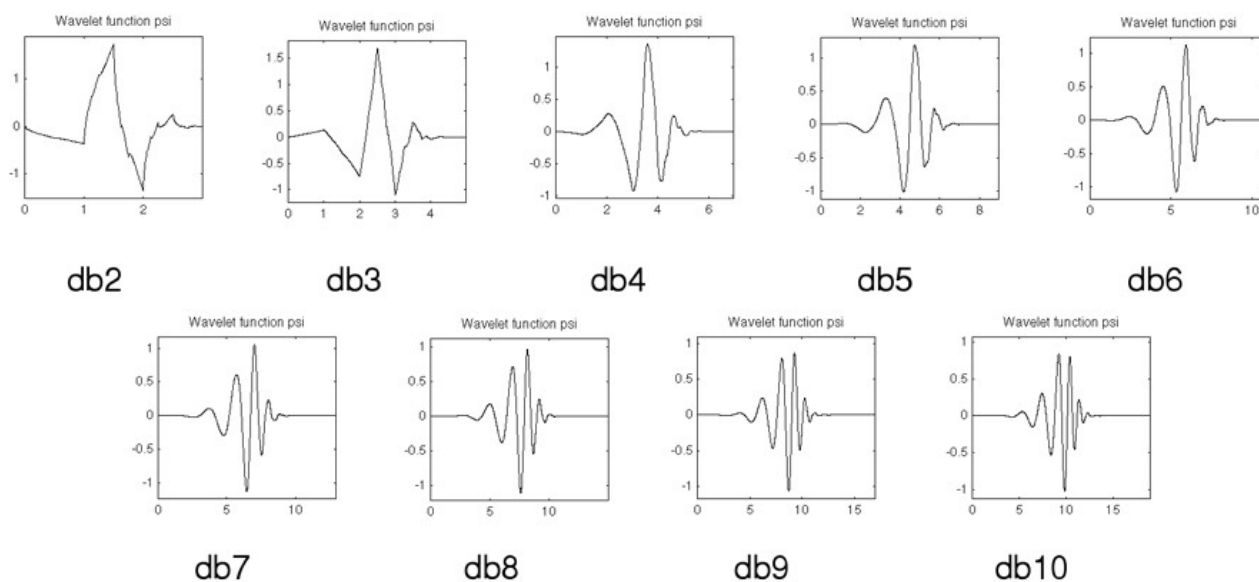


Obr 4.10 – Frekvenční analýza signálu z obrázku 4.9, frekvence rušení způsobeného během je zhruba 2,7 Hz.

4.3 Knihovna pro filtraci založenou na SWT

4.3.1 Úvod

V kapitole 2 jsme probrali principy a použití jak vlnkové transformace, tak i stacionární vlnkové transformace (SWT) a popsali způsob, jak je možné využít SWT pro filtraci signálů. Dále jsme pak probrali filtraci za pomoci SWT s adaptivním nastavením prahování koeficientů. Nyní se detailněji zaměříme na provedení filtrace a na využití dat získaných z akcelerometru při nastavení koeficientů prahování. Na základě výsledků prezentovaných v diplomové práci Lukáše Paroulka [4] byla zvolena pro vlnkovou transformaci vlnka z rodiny Daubechies, konkrétně Daubechies db4.



Obr 4.11 – Vlnky z rodiny Daubechies, číslo vlnky značí počet jejich koeficientů. O vlnkách Daubechies je možné se dočíst více v [20].

4.3.2 Popis implementace

Cílem implementovaných funkcí je získat data z PPG senzoru a odstranit z nich pohybové artefakty. Po odstranění pohybových artefaktů následuje výpočet srdeční frekvence. Filtrace je rozdělena do několika iterací, například každá úroveň rozkladu signálu pomocí SWT probíhá v samostatné iteraci.

Čtení dat z PPG senzoru se provádí za použití rozšíření Cx operačního systému CircleOS, za pomoci kterého čteme signál z PPG senzoru připojeného na AD převodník (ADC), přečtená data jsou ukládána do kruhového bufferu.

Při zahájení samotné filtrace a výpočtu srdeční frekvence se nejprve zpracují data z akcelerometru, tedy zavolá se funkce pro detekci nejvýznamnější frekvence v rušení pomocí Fourierovy transformace popsaná v podkapitole věnované knihovně pro práci s akcelerometrem. Po zjištění této frekvence se ještě vypočte efektivní hodnota dat z akcelerometru. Pokud efektivní hodnota překročí daný práh, je pohyb vyhodnocen jako běh, v opačném případě jako chůze. Frekvence pohybu je průměrována přes 4 poslední výsledky. Na základě zjištěné frekvence a vyhodnocení zda se jedná o chůzi či běh je vybrána jedna ze sad úrovní prahování koeficientů. Celkem se zde rozlišuje 5 sad. Jedná se o sadu pro klid nebo pomalý pohyb, pro středně rychlý pohyb, rychlejší pohyb - chůzi, rychlejší pohyb – běh a pro rychlý pohyb.

Následuje příprava výpočtu stacionární vlnkové transformace, ve které se data naměřená PPG senzorem přepírají do pracovního bufferu – vzhledem k tomu, že transformace používá komplexní čísla, jsou data nakopírována v komplexním prokládaném tvaru, kdy sudé indexy obsahují reálnou část a liché indexy imaginární část.

Protože užitečný signál v datech získaných z PPG senzoru má úzké frekvenční spektrum, jsou neúžitečné frekvence ze signálu odfiltrovány filtrem typu pásmová propust' s dolní hranicí 0,2Hz a horní hranicí 5Hz. Tím se zároveň odstraní ze signálu stejnosměrná složka.

Po provedení této filtrace začíná výpočet stacionární vlnkové transformace, jehož výstupem je pracovní buffer naplněný detailními koeficienty swd všech úrovní a aproximačním koeficientem swa z poslední úrovně. Implementace je založena na C++ open-source knihovně Wavelet Library, pro výpočet Fourierovy transformace jsou ovšem používány funkce převzaté z DSP knihovny vyvinuté STMicroelectronics.

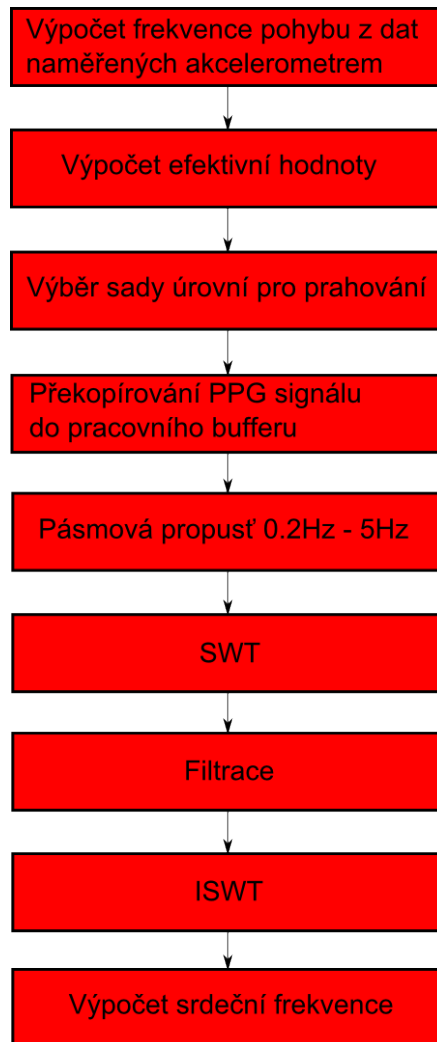
Dalším krokem je provedení samotné filtrace, kdy jsou detailní koeficienty prahovány podle sady úrovní prahování koeficientů vybrané v počátku výpočtu na základě analýzy dat získaných z akcelerometru. Společně s prahováním je možné také aplikovat na libovolné koeficienty detailů filtr typu pásmová propust'.

Po provedení filtrace je nutné provést zpětnou stacionární vlnkovou transformaci (ISWT). Jejím výstupem je PPG signál zbavený pohybových artefaktů, který je už možné použít pro výpočet srdeční frekvence.

Protože SWT a ISWT jsou výpočetně i paměťově (vypočtené koeficienty je nutné ukládat) náročné operace, bylo nutné provést několik úprav pro urychlení výpočtu. Během pozorování průběhů PPG signálu bylo zjištěno, že koeficienty swd6 a vyšší neobsahují žádnou informaci o srdečním tepu, ale pouze artefakty. Knihovna může být proto nakonfigurována tak, aby posledních N detailních koeficientů nepočítala ani neukládala, ale počítala pouze aproximační koeficient. Tím je snížena jak paměťová tak výpočetní složitost. Výpočetní složitost byla ještě dále snížena tím, že se daný detailní koeficient nepočítá pokud je jeho úroveň prahování nastavena na 1 – tedy že má být vše odstraněno.

4.3 Knihovna pro filtraci založenou na SWT

Posledním krokem je výpočet samotné srdeční frekvence. Pro její výpočet jsou dostupné dvě funkce – jedna založená na frekvenční analýze, druhá založená na hledání vrcholů. Funkce založená na frekvenční analýze vezme filtrovaný PPG signál a vypočte Fourierovu transformaci. Srdeční frekvence je pak vypočtena z indexu nejvíce zastoupené frekvence. Funkce používající hledání vrcholů je založena na funkci `findpeaks` dostupné v MATLABu, srdeční frekvence je vypočtena z počtu nalezených vrcholů. Při použití této funkce je stejně jako v MATLABu nutné zadat minimální vzdálenost mezi dvěma vrcholy. Výsledná srdeční frekvence je průměrována přes 4 poslední vzorky. Celý postup filtrace je znázorněn na následující straně na obrázku 4.12.



Obr 4.12 – Postup při filtraci PPG signálu

5 Uživatelská aplikace HR Monitor

5.1 Cíle uživatelské aplikace

Cílem uživatelské aplikace Heart Rate Monitor (zkráceně HR Monitor) vyvinuté pro EvoPrimer s procesorovým modulem STM30F103VET6 je provádět dlouhodobý monitoring srdeční činnosti pacienta. To zahrnuje získávání signálu z akcelerometru, PPG senzoru a EKG modulu, eliminaci pohybových artefaktů v PPG signálu pomocí výše zmíněných knihoven pro práci s akcelerometrem a filtraci PPG signálu a výpočet srdeční frekvence pacienta na základě filtrovaného PPG signálu. Signál z EKG modulu slouží pro ověření korektnosti výpočtů. Zároveň je potřeba naměřená a vypočtená data pravidelně zaznamenávat v dostatečné frekvenci na micro SD kartu.

5.2 Koncepce aplikace

Aplikace využívá operační systém CircleOS a její základ stejně jako u ostatních aplikací pro tento operační systém tvoří inicializační funkce a aplikační handler.

5.2.1 Inicializační funkce

Na počátku inicializační funkce je nutné zkontrolovat, zda je používaná verze CircleOS vyšší nebo rovná než minimální verze požadovaná aplikací. Následně je nastavena frekvence volání aplikačního handleru. Frekvence volání určuje, jak často v závislosti na přerušení SysTicku (systémového časovače) se bude aplikační handler volat – zda při každém přerušení SysTicku, při každém druhém atd. Zde je pro aplikační handler nastaveno, aby se volal při každém SysTicku. Samotná konfigurace SysTicku a ostatních přerušení bude probrána později. Poté se provede nastavení prioritních skupin NVIC (Nested Vector Interrupt Controller) [22], [23] a konfigurace PLL. Tím inicializační funkce končí a je spuštěn aplikační handler.

5.2.2 Aplikační handler

Aplikační handler obvykle představuje hlavní aplikační smyčku u aplikací vytvořených pro CircleOS. Nejdříve je provedena inicializace micro SD karty a založení nového souboru o zvolené velikosti. Nyní je velikost souboru nastavena na 8192 clusterů, což při velikosti clusteru 4096B odpovídá velikosti souboru 32 MB. Založení probíhá tak jak bylo popsáno v kapitole věnované optimalizaci DosFs, je tedy vytvořen prázdný soubor o požadované velikosti a nastavena bitmapa volného místa. Následuje inicializace akcelerometru, PPG senzoru a EKG modulu. Akcelerometr je připojen pomocí rozhraní SPI, signál z PPG a EKG se získává pomocí AD převodníků.

Vzhledem k tomu, že je nutné v aplikaci pravidelně číst data a zaznamenávat je na micro SD kartu a zároveň provádět výpočetně náročnou eliminaci pohybových artefaktů spolu s výpočtem srdeční frekvence, bylo nutné kromě aplikačního handleru využívat přerušení od časovače. Časovač je použit pro výpočet zmíněné eliminace artefaktů a výpočet srdeční frekvence, zatímco aplikační handler slouží k získávání dat z akcelerometru, PPG senzoru, EKG modulu a záznamu dat na micro SD kartu.

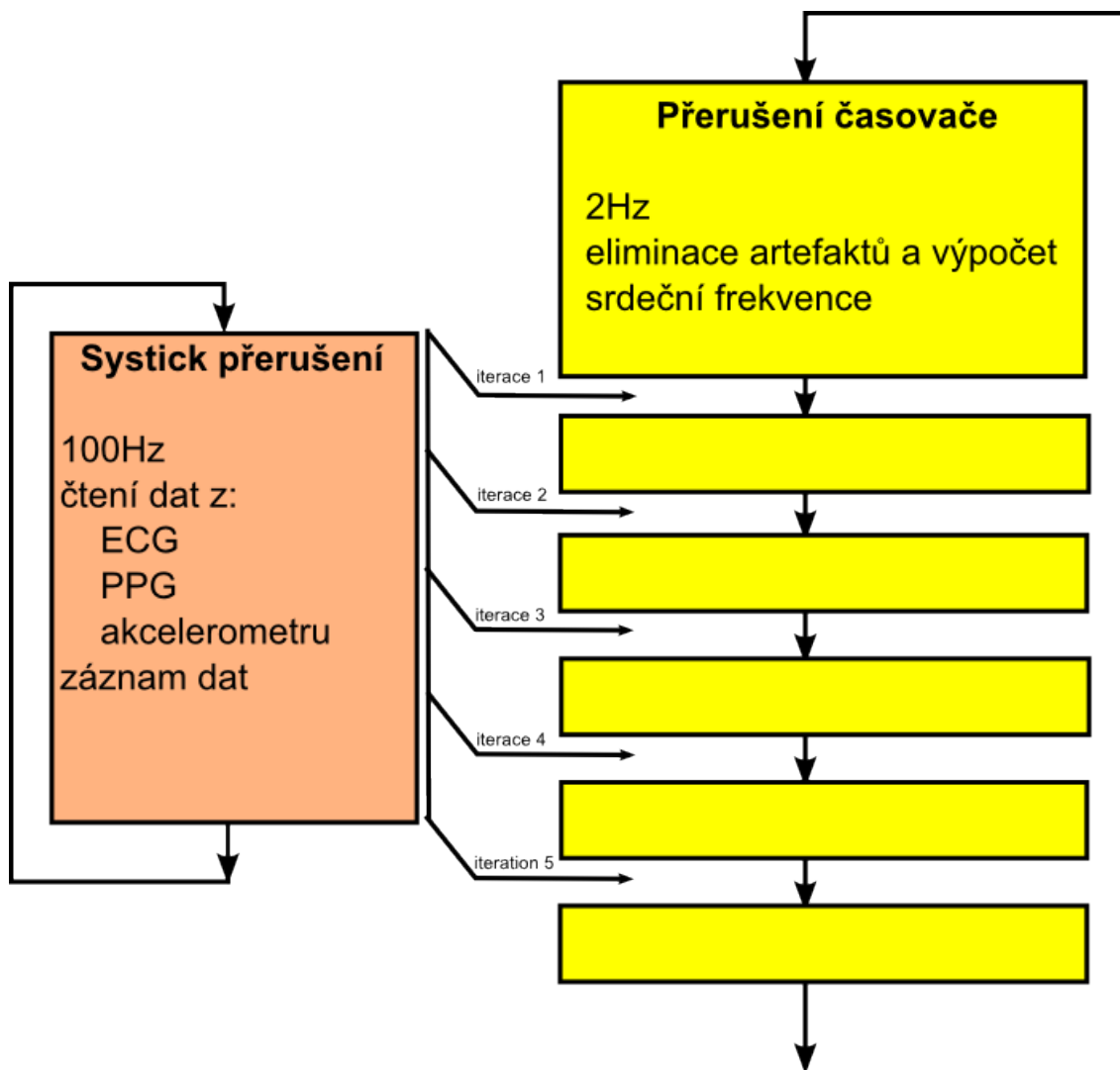
Priority jsou nastaveny tak, aby mohl aplikační handler přerušovat přerušení od časovače. Tím je dosaženo toho, že jsou data získávána i zapisována pravidelně.

Frekvence volání handleru je 100 Hz, nicméně PPG senzor a akcelerometr jsou čteny pouze každé třetí volání, tedy s frekvencí zhruba 33,33 Hz. Důvodem je to, že jak u dat z akcelerometru, která reprezentují pohyb pacienta, tak i u PPG signálu není žádný důvod pro vyšší vzorkovací frekvenci, protože oba signály ve vyšších frekvencích již neobsahují žádná užitečná data.

Oproti tomu data z EKG modulu jsou získávána při každém volání aplikačního handleru, tedy s frekvencí 100 Hz. Důvodem je to, že ačkoli srdeční frekvence je mnohem nižší, špičky EKG signálu značící srdeční tep jsou velmi úzké a například při použití vzorkovací frekvence 33,33 Hz často docházelo k jejich výpadkům. Data na micro SD kartu se zaznamenávají s frekvencí 100 Hz. Zaznamenávají jsou následující údaje: čas záznamu, poslední hodnota získaná z akcelerometru, PPG senzoru a ECG modulu, poslední vypočtená frekvence pohybu a poslední vypočtená srdeční frekvence.

Jak už bylo napsáno, výpočet srdeční frekvence probíhá v přerušení časovače a na zvoleném vývojovém kitu trvá 5 až 8 vteřin v závislosti na konfiguraci knihovny pro eliminaci pohybových artefaktů. To je způsobeno z velké části chybějící FPU jednotkou použitého procesoru Cortex-M3, protože většina výpočtů zde probíhá v plovoucí čárce.

Aplikace se ukončuje stiskem tlačítka. Při ukončování aplikace je uzavřen soubor se zaznamenanými daty a zkrácen na svou skutečnou délku, zároveň jsou obnoveny původní nastavení priorit přerušení a frekvence volání SysTick přerušení. Struktura aplikace je popsána na následující straně na obrázku 5.1.



Obr 5.1 – Struktura aplikace HR Monitor

6 Ověření funkce systému a výsledky měření

6.1 Popis průběhu měření

Cílem měření je ověřit schopnost aplikace měřit srdeční puls pacienta pomocí PPG senzoru a po provedení filtrace korektně detekovat tepovou frekvenci. Vzhledem k zaměření filtrace na odstraňování pohybových artefaktů z PPG signálu bylo zvoleno použití běžícího pásu. Měření probíhalo po úsecích délky 4 minuty a to v klidu, při chůzi na páse a nakonec při běhu. Rychlost chůze byla 3 km/h, 4 km/h a 6 km/h, běh na páse probíhal rychlostí 8 km/h. Během měření byl PPG senzor umístěn v uchu a zároveň byly připojeny EKG elektrody pro ověření správnosti měření. Akcelerometr byl přilepen k PPG senzoru, aby bylo zajištěno, že pohyby PPG senzoru způsobující rušení budou akcelerometrem přesně zaznamenány. Po ukončení měření byla naměřená data analyzována v prostředí MATLAB a bylo zjišťováno, zda srdeční frekvence naměřená vyvinutým systémem odpovídá srdeční frekvenci vypočtené v MATLABU z EKG signálu.

6.2 Konfigurace aplikace

Na základě předchozích pokusů s měřením srdeční frekvence byly nastaveny koeficienty prahování tak, jak ukazuje tabulka 6.1. Úroveň 0 zde znamená, že daný koeficient byl ponechán beze změny, úroveň 1 naopak značí, že byl zcela odstraněn. Úroveň zadaná desetinným číslem značí, že byly ponechány pouze hodnoty vyšší a rovné danému procentu maxima koeficientu. Například úroveň koeficientu 0,75 znamená, že všechny hodnoty koeficientu nižší než 75 procent maximální absolutní hodnoty v koeficientu byly prahovány a nastaveny na 0. V některých případech je po prahování navíc na koeficienty swd3 a swd4 aplikován bandpass filtr. Důvodem pro použití tohoto filtru je, že srdeční frekvence očekáváme v určitém rozmezí a takto je možné odstranit z koeficientů frekvence, které potenciálně mohou reprezentovat pomalý pohyb (pod 1 Hz), velmi rychlý pohyb (například nad 2,5 Hz) a nebo vyšší harmonické frekvence pohybu. Detekce srdeční frekvence se provádí ve frekvenční oblasti za použití Fourierovy transformace.

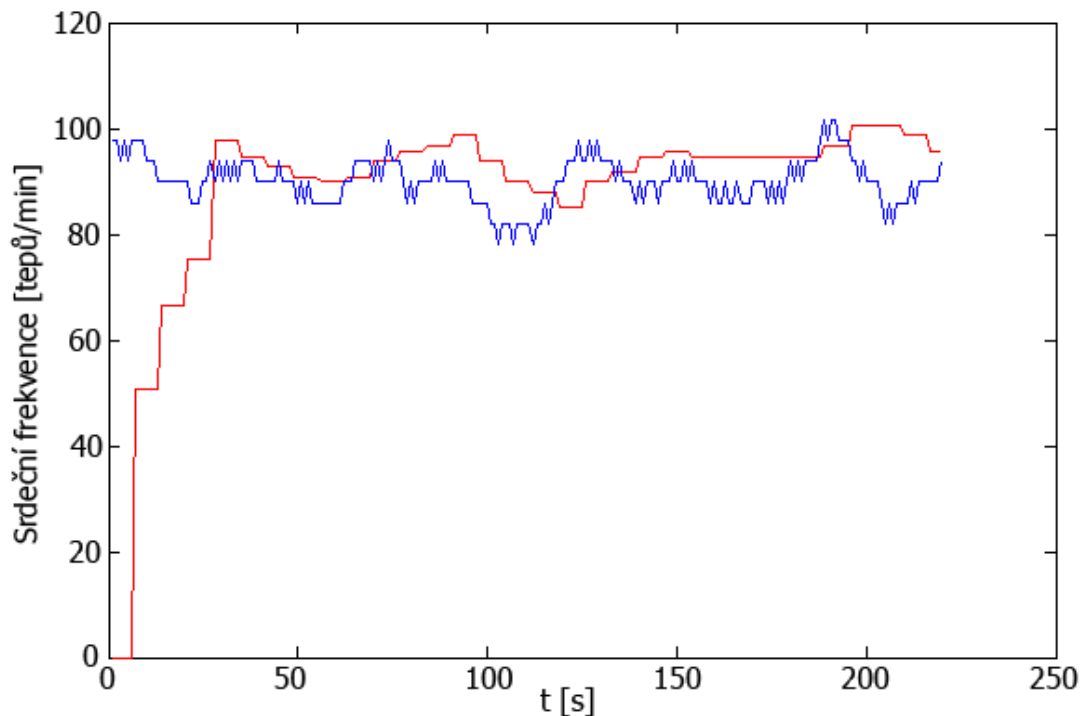
6 Ověření funkce systému a výsledky měření

Druh pohybu	Rozsah frekvencí pohybu	Nastavené prahování od swd1 do swd8	Bandpass
Klid	0 Hz – 0,65 Hz	1, 1, 0,5, 0, 0,75, 1, 1, 1	není
Pomalý pohyb	0,65 Hz – 1,1 Hz	1, 1, 0,5, 0, 0,75, 1, 1, 1	není
Středně rychlý pohyb	1,1 Hz – 1,5 Hz	1, 1, 0, 0,5, 1, 1, 1, 1	0,9 Hz – 2,5 Hz
Rychlejší pohyb	1,5 Hz – 1,75 Hz	1, 1, 0, 0,5, 1, 1, 1, 1	0,9 Hz – 2,5 Hz
Rychlý pohyb	> 1,75 Hz	1, 1, 0, 0,5, 1, 1, 1, 1	1 Hz – 3,3 Hz

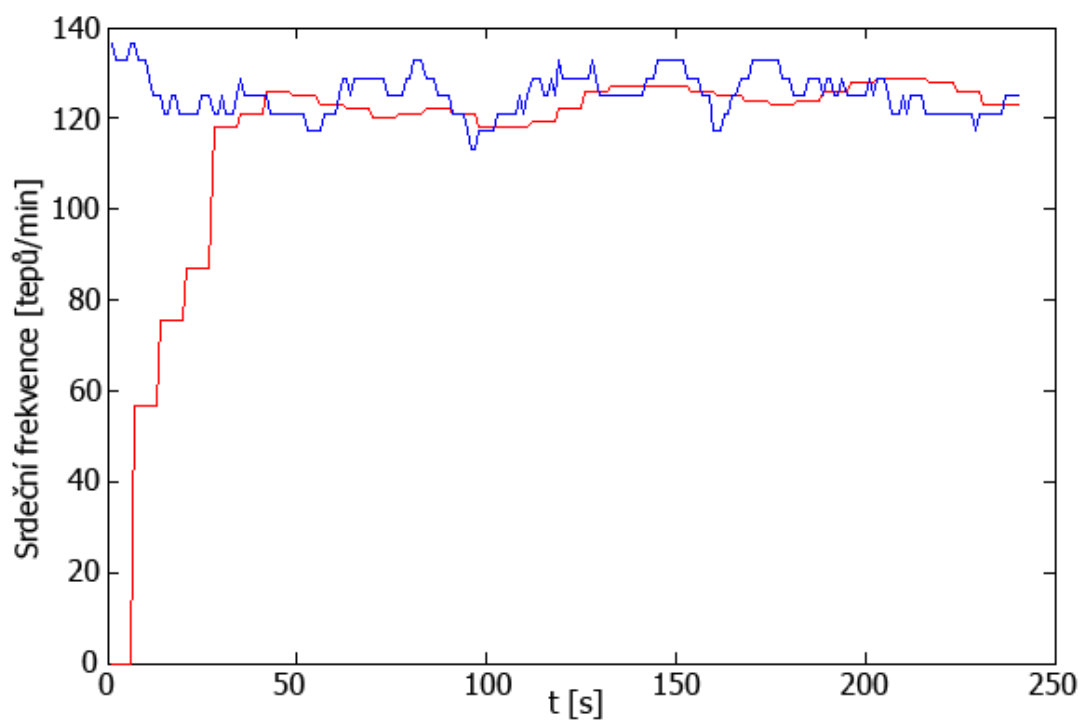
Tabulka 6.1 – Nastavení prahování koeficientů

6.3 Výsledky měření

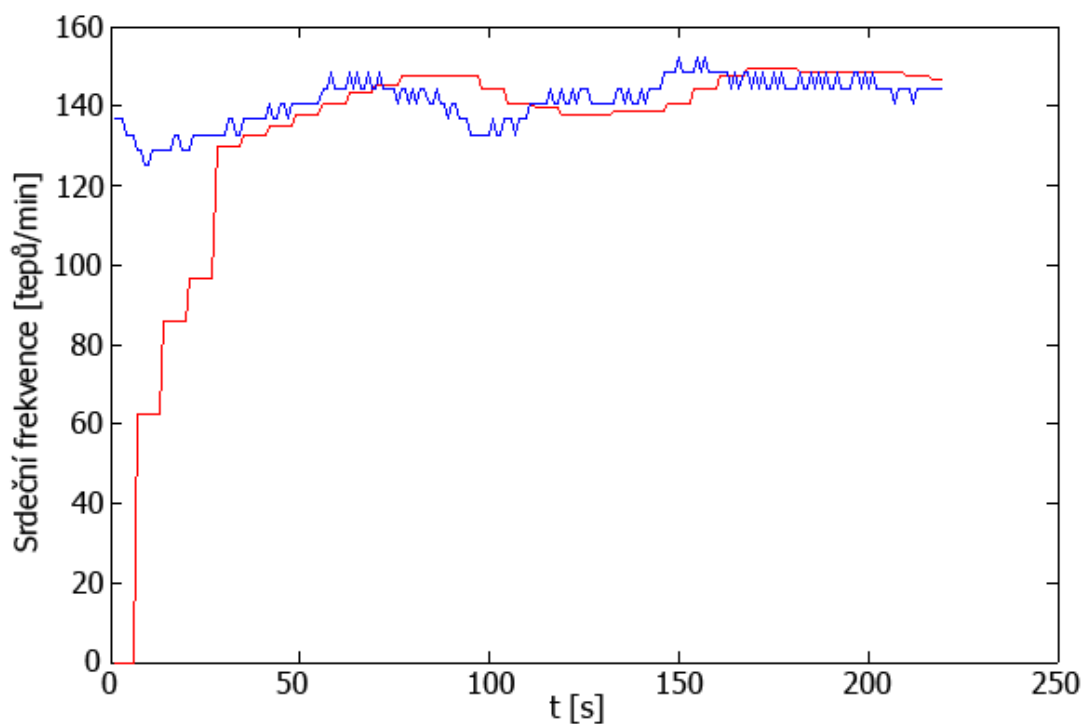
Výsledky měření v klidu, při chůzi a při běhu jsou zobrazeny na následujících obrázcích. V grafech je srdeční frekvence vypočtená z PPG zobrazena červeně, frekvence vypočtená z referenčního EKG signálu je zobrazena modře.



Obr 6.1 – Porovnání srdečních frekvencí z PPG a EKG v klidu. Je vidět, že zde se srdeční frekvence počítá správně a víceméně kopíruje frekvenci vypočtenou z EKG. Zpoždění signálu je způsobeno především dobou výpočtu zhruba 7 vteřin. Na zpoždění se podílí také průměrování srdeční frekvence přes 4 poslední výsledky a zpoždění samotného PPG signálu oproti EKG.

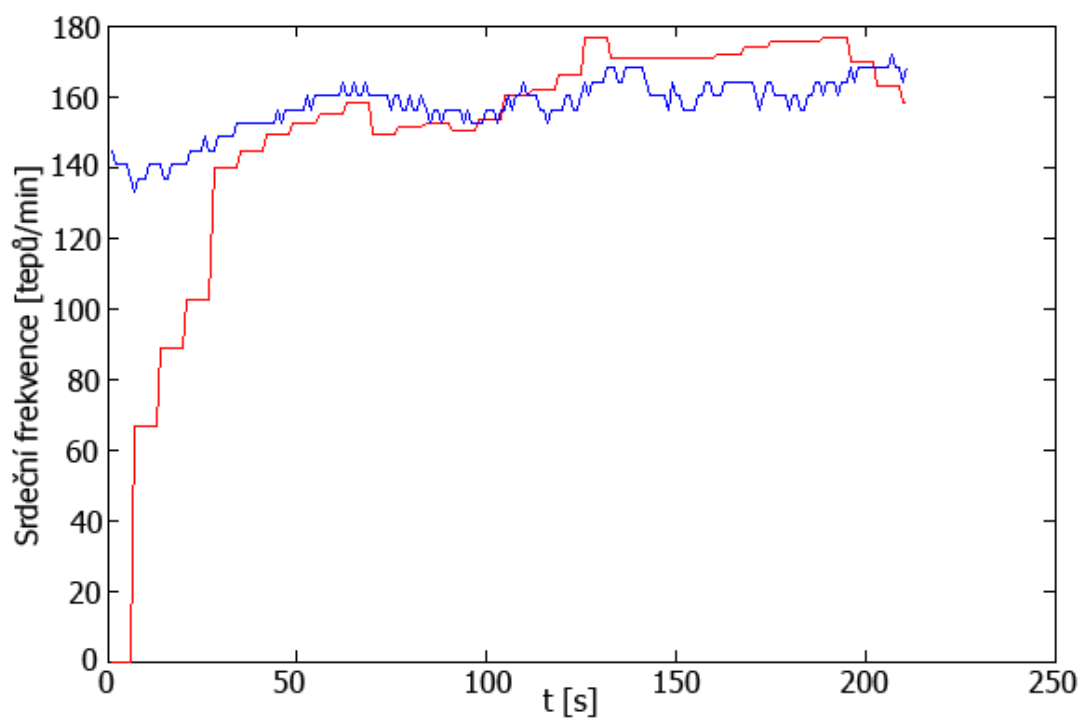


Obr 6.2 – Porovnání při chůzi rychlostí 3 km/h. Detekovaná frekvence rušení se pohybovala okolo 1.43Hz. Srdceční frekvence se ve většině případů počítá opět správně.

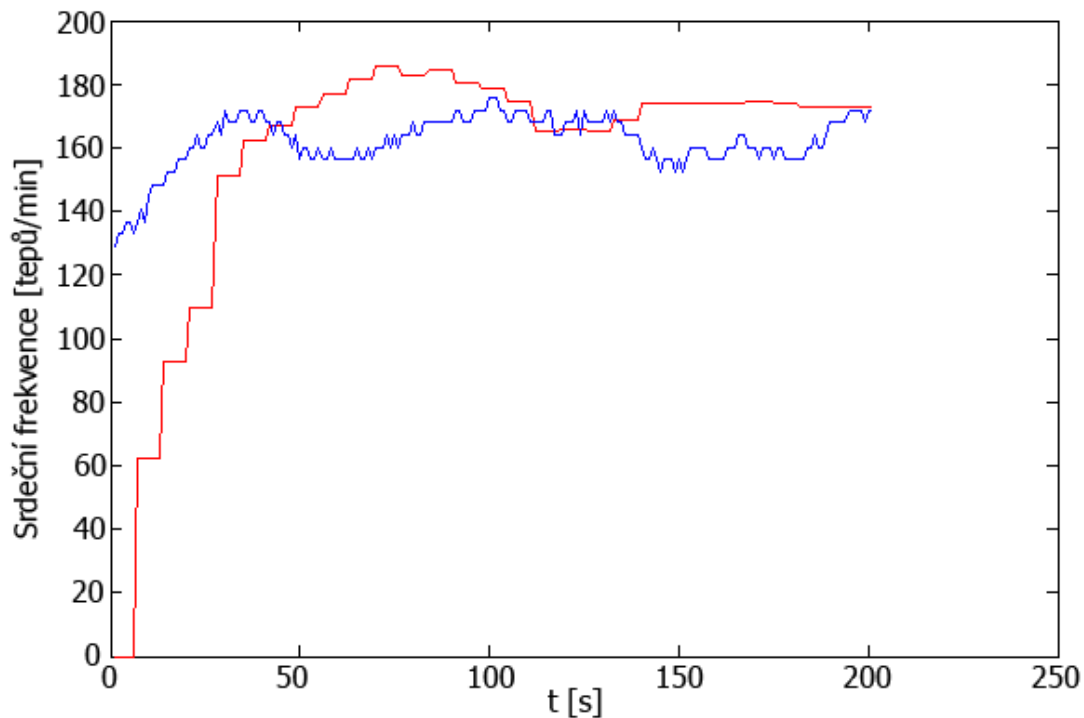


Obr 6.3 – Srdceční frekvence při chůzi rychlostí 4 km/h. Výpočet frekvence probíhá ve většině případů správně. Frekvence rušení se pohybovala mezi 1,69 Hz a 1,82 Hz.

6 Ověření funkce systému a výsledky měření



Obr 6.4 – Srdeční frekvence při chůzi rychlostí 6 km/h. Vidíme zde, že ve většině případů je frekvence detekována korektně, někde ale dochází k odchylce až o 20 tepů za minutu. Frekvence rušení se pohybovala od 2,08 Hz do 2,3 Hz.



Obr 6.5 – Srdceční frekvence při běhu rychlostí 8 km/h. Frekvence rušení byla 2,6 Hz až 2,9 Hz. Vidíme zde, že vypočtená frekvence se sice pohybuje poměrně blízko frekvenci vypočtené z EKG, ale nekopíruje EKG ani se zpožděním, naopak zůstává v jedné úrovni. To může značit, že není ve skutečnosti detekována srdceční frekvence, ale frekvence pohybu. Tomu by odpovídalo i to, že detekovaná srdceční frekvence je zhruba 170 tepů za minutu, tedy asi 2,9 Hz, což je přesně frekvence rušení. Na konci měření došlo k odpadnutí EKG elektrody a následnému vyjmutí PPG senzoru z ucha a přerušeni měření, posledních 20 vteřin proto neobsahuje platná data.

6.4 Shrnutí výsledků měření

Při měření bylo zjištěno, že výpočet srdceční frekvence probíhá poměrně přesně v klidu a při chůzi rychlostí do 4 km/h. I při chůzi rychlostí 6 km/h se naměřené hodnoty pohybovaly v rozumných mezích. Při běhu rychlostí 8 km/h ale již docházelo ke značné odchylce od skutečné srdceční frekvence a navíc byla pravděpodobně frekvence rušení chybně vyhodnocena jako srdceční frekvence. Dalším problémem je poměrně dlouhá doba výpočtu pohybující se okolo 7 vteřin, která způsobuje zpoždění výpočtu, nízkou frekvenci výpočtů a dlouhý rozjezd výpočtu kdy jsou platná data dostupná až po zhruba 30 vteřinách.

Během měření bylo také zjištěno, že užitečný signál obsahují prakticky výhradně koeficienty swd3 a swd4, kterým při vzorkovací frekvenci 33,33 Hz odpovídají frekvenční rozsahy zhruba 2,08 Hz – 4,16 Hz, respektive 1,04 Hz až 2,08 Hz. Ostatní koeficienty neobsahují data nezbytně nutná pro detekci srdceční frekvence. Také bylo zjištěno, že amplituda koeficientu swd3 je výrazně nižší než u swd4. Vzhledem k tomu, že vyšší srdceční frekvence patří do frekvenčního spektra koeficientu swd3, je jejich detekce obtížnější, protože z výše zmíněného důvodu snadno zaniknou v porovnání s swd4.

6.5 Úprava filtrační metody

Na základě poznatků získaných při měření srdeční frekvence byla filtrační metoda upravena tak, aby se zlepšila detekce vyšších srdečních frekvencí.

6.5.1 Zkrácení výpočtu SWT

Jak již bylo uvedeno, užitečné informace obsahují pouze koeficienty swd3 a swd4. Koeficienty swd1 a swd2 obsahují šum a neužitečné detaily PPG signálu, koeficienty swd5 a dále naopak reprezentují příliš nízké frekvence. Vzhledem k tomu, že ani aproximační koeficient swa8 neobsahuje žádná užitečná data, je zbytečné počítat SWT až do osmé úrovně. Místo toho byl výpočet zkrácen pouze do čtvrté úrovně, přičemž jsou ponechány pouze koeficienty swd3 a swd4, aproximační koeficient swa4 je odstraněn. Touto úpravou došlo k výraznému zrychlení výpočtu.

6.5.2 Zlepšení detekce vyšších frekvencí a potlačení rušení

V kapitole 6.4 byl zmíněn problém s detekcí vyšších srdečních frekvencí (nad 2 Hz), které patří do frekvenčního spektra koeficientu swd3, způsobený výrazně nižší amplitudou koeficientu swd3 oproti swd4, který reprezentuje nižší frekvence. V důsledku toho může srdeční frekvence nad 2 Hz snadno zaniknout, čímž dojde k chybnému výpočtu.

K odstranění tohoto problému jsme použili výpočet efektivní hodnoty (RMS). Nejprve jsou vypočteny efektivní hodnoty swd3 a swd4, pak je koeficient swd4 vynásoben hodnotou $\text{rms}(\text{swd3}) / \text{rms}(\text{swd4})$. Tím jsou amplitudy swd3 a swd4 vyrovnány. Poté se ještě swd4 násobí parametrem závislým na detekované frekvenci pohybu, který se pohybuje od 0,25 do 1,25. Při nižší frekvenci pohybu očekáváme nižší srdeční frekvenci, proto je parametr nastaven na hodnotu vyšší než 1, čímž swd4 získává větší váhu ve výsledné funkci. Naopak při vyšší frekvenci pohybu očekáváme vyšší srdeční frekvenci. V takovém případě má parametr hodnotu nižší než 1 a vyšší váhu tak získává koeficient swd3.

Dále byla ještě zavedeno potlačení frekvencí rovných nebo blízkých frekvenci pohybu. Tyto frekvence není možné odstranit, protože by mohlo dojít k odstranění užitečných dat, kdyby byla srdeční frekvence shodná s frekvencí pohybu. Proto jsou pouze při Fourierově transformaci zčásti utlumeny, aby se snížil jejich význam ve výsledném signálu.

6.5.3 Dvoufázový výpočet srdeční frekvence

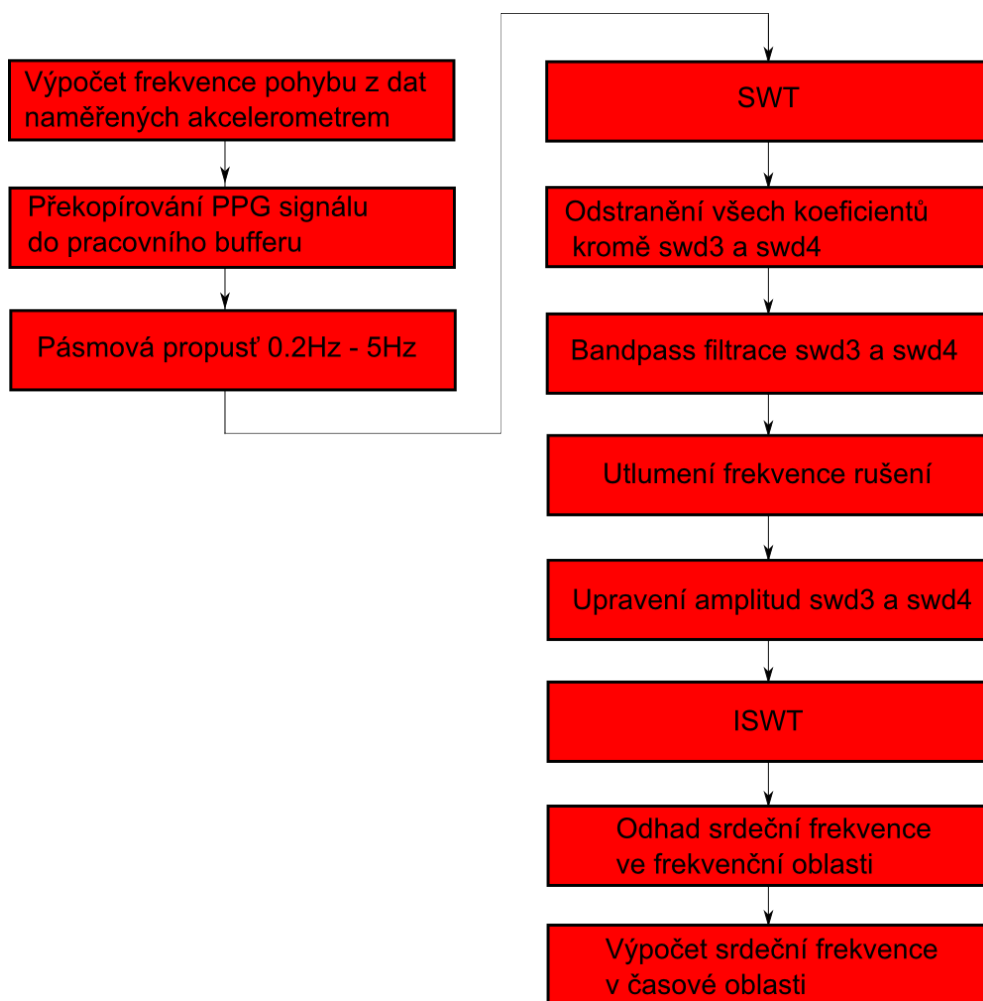
Při testování bylo také odhaleno, že detekce srdeční frekvence ve frekvenční oblasti je náchylná k náhlým propadům či naopak špičkám způsobeným rušením signálu. Tento problém není pozorovatelný na zde zobrazených grafech, nicméně se během testování objevoval. Z toho důvodu byla vyzkoušena detekce srdeční frekvence v časové oblasti za použití funkce pro hledání vrcholů, implementované podle funkce findpeaks z MATLABu. Nevýhodou tohoto řešení naopak je, že pro správnou funkci je nutné zadat minimální vzdálenost mezi vrcholy. Vzhledem k tomu, že se srdeční frekvence může měnit v rozsahu od 1 Hz až po více než 3 Hz, není možné použít jednu univerzální vzdálenost mezi vrcholy, protože by pak docházelo buď k detekci vrcholů tam kde nejsou, nebo naopak k přeskočení některých vrcholů.

Tento problém byl vyřešen tak, že se nejprve provede odhad srdeční frekvence ve frekvenční oblasti a na jeho základě se stanoví minimální vzdálenost mezi vrcholy. Pak je provedena detekce v časové oblasti s takto nastavenou minimální vzdáleností mezi vrcholy. Na základě počtu nalezených

vrcholů se vypočte srdeční frekvence. Jak srdeční frekvence tak i její odhad jsou průměrovány přes 4 vzorky.

6.5.4 Shrnutí provedených úprav

Úpravy popsané v minulých podkapitolách by měly zajistit účinnější filtraci pohybových artefaktů a tím také přesnější stanovení srdeční frekvence. Zároveň se podařilo celý výpočet značně urychlit díky zkrácení výpočtu SWT. Zatímco původně trval výpočet 5 až 8 vteřin, nyní zabere pouze 3,2 vteřiny. Obrázek 6.6 znázorňuje postup výpočtu při filtraci s výše popsanými úpravami.



Obr 6.6 – Postup při filtraci po provedení úprav popsaných v kapitole 6.5

6.6 Konfigurace upravené aplikace

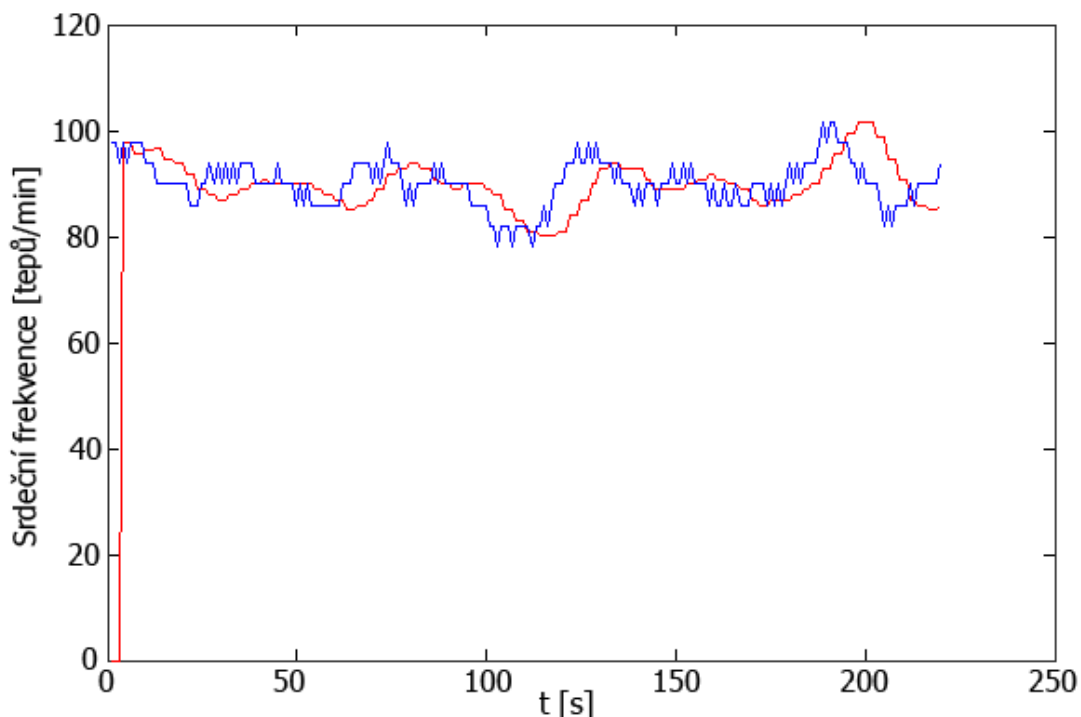
Na rozdíl od původní aplikace se zde již nenastavuje prahování koeficientů, které je trvale nastaveno na zachování koeficientů swd3 a swd4, ale konfigurace se týká bandpass filtrů, úpravy amplitudy koeficientu swd4 a frekvenčních rozsahů pro jednotlivé intenzity pohybu. Použitá nastavení jsou zobrazena v tabulce 6.2.

Druh pohybu	Rozsah frekvencí pohybu	Bandpass	Parametr násobící swd4
Klid	0 Hz – 0,65 Hz	1 Hz – 2,7 Hz	1,25
Pomalý pohyb	0,65 Hz – 1,1 Hz	1,2 Hz – 2,7 Hz	0,75
Středně rychlý pohyb	1,1 Hz – 1,5 Hz	1,3 Hz – 3,3 Hz	0,45
Rychlejší pohyb	1,5 Hz – 2 Hz	1,4 Hz – 3,3 Hz	0,35
Rychlý pohyb	> 2 Hz	1,6 Hz – 3,3 Hz	0,25

Tabulka 6.2 – nastavení upravené aplikace

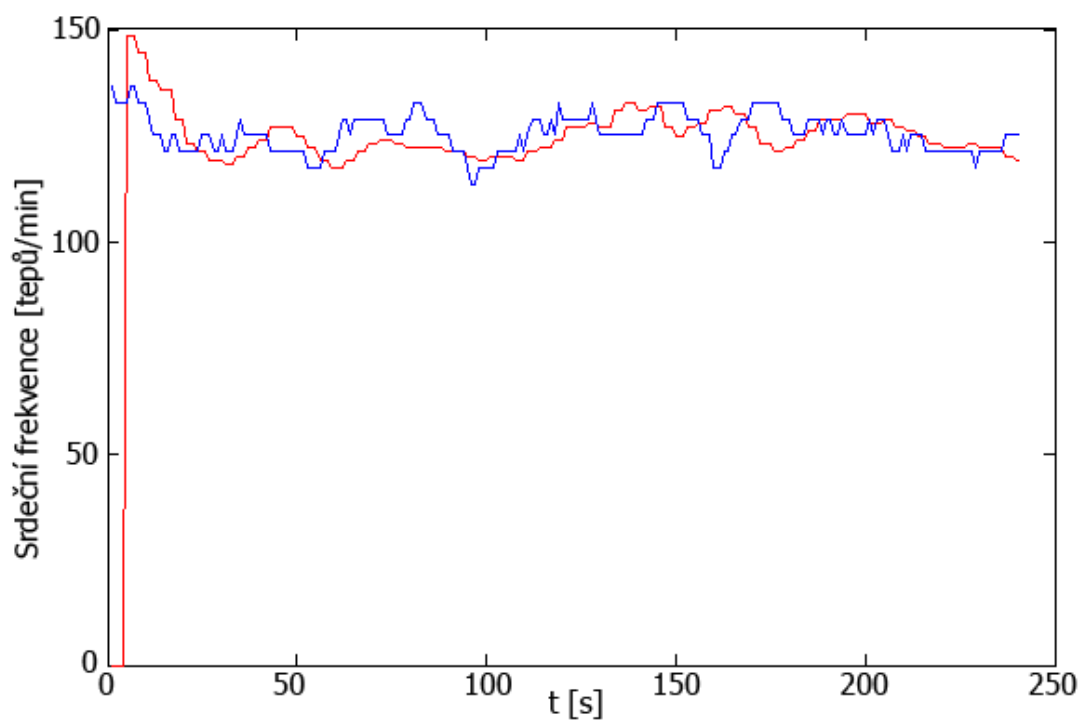
6.7 Výsledky měření při použití upravené aplikace

Výsledky měření v klidu, při chůzi a při běhu jsou zobrazeny na následujících obrázcích tak jako u výsledků měření s původní aplikací. V grafech je srdeční frekvence vypočtená z PPG zobrazena červeně, frekvence vypočtená z referenčního EKG signálu je zobrazena modře.

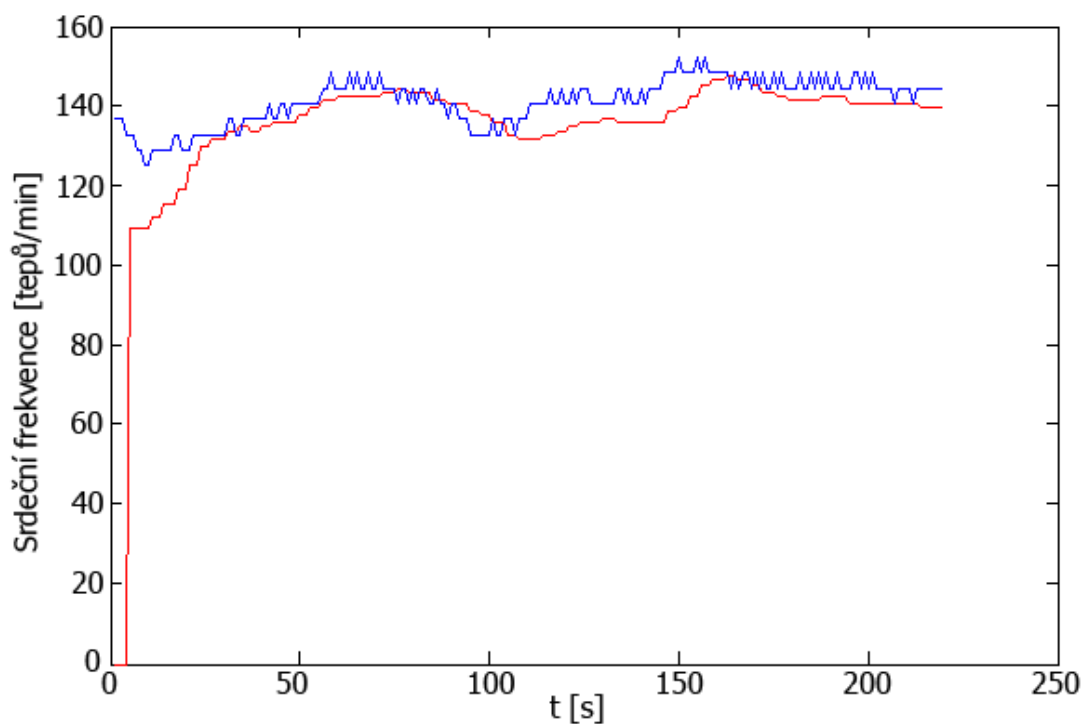


Obr 6.7 – Srdeční frekvence v klidu, vypočtená frekvence z PPG kopíruje frekvenci získanou z EKG přesně, ale se zpožděním. Zpoždění je způsobeno dobou výpočtu zhruba 3 vteřiny a zpožděním PPG signálu za EKG. Vliv na zpoždění má také dvoufázový výpočet srdeční frekvence s průměrováním hodnot. Zpoždění je nicméně výrazně nižší než u původní metody filtrace.

6.7 Výsledky měření při použití upravené aplikace

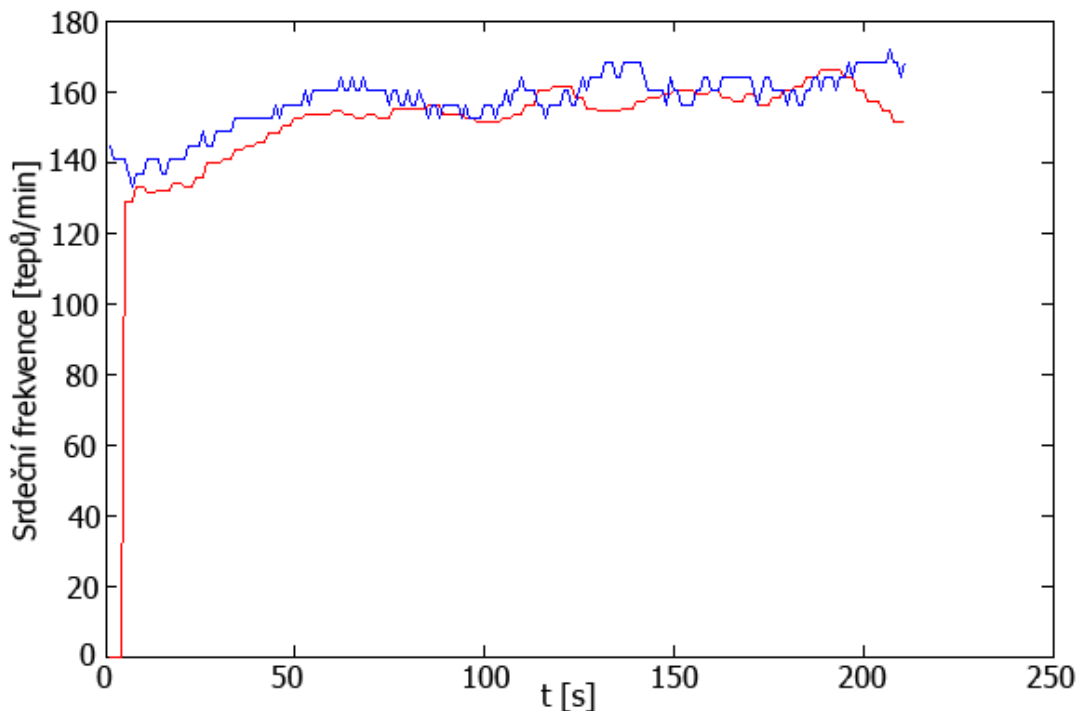


Obr 6.8 – Srdční frekvence při chůzi rychlostí 3 km/h. Hodnoty z PPG odpovídají hodnotám z EKG, výpočet probíhá korektně.

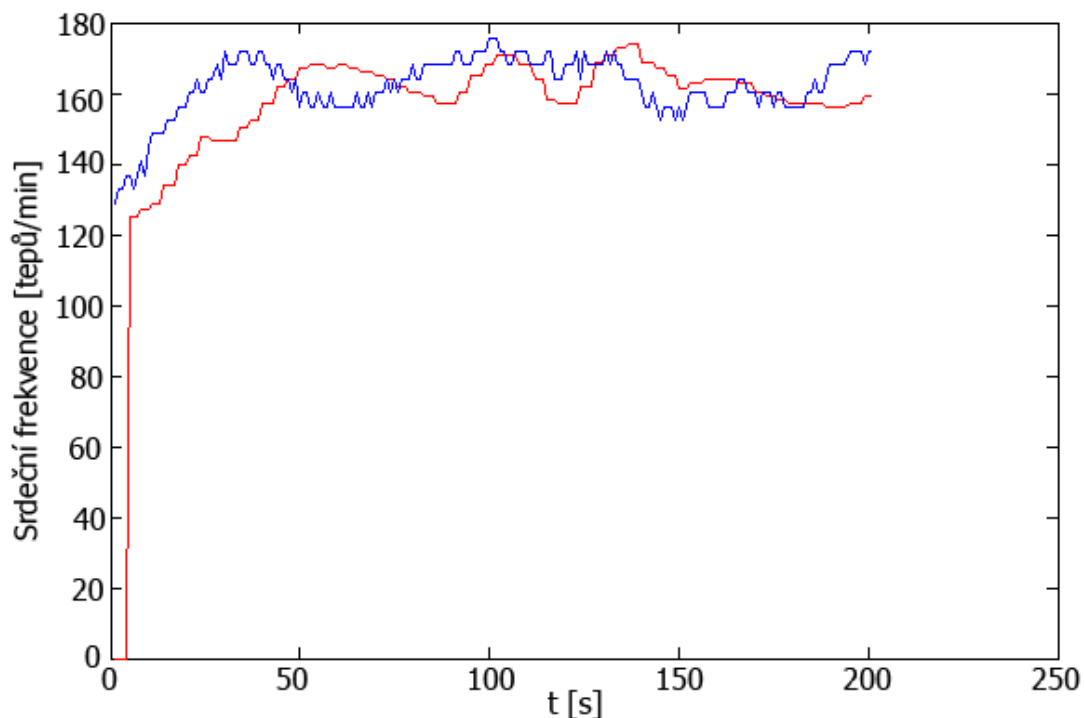


Obr 6.9 – Srdční frekvence při chůzi rychlostí 4 km/h. Frekvence je opět detekována správně.

6 Ověření funkce systému a výsledky měření



Obr 6.10 – Srdeční frekvence při chůzi rychlostí 6 km/h. Výsledky jsou výrazně lepší než při použití původní filtrační metody.



Obr 6.11 – Srdeční frekvence při běhu rychlostí 8 km/h. I když dochází k nepřesnostem, výsledky jsou opět výrazně lepší než při použití původní filtrační metody. Je zde také jasně vidět, že je opravdu detekována srdeční frekvence a ne frekvence pohybu, protože srdeční frekvence se zpožděním kopíruje frekvenci vypočtenou z EKG. Na konci měření došlo k odpadnutí elektrody a následnému přerušení měření, posledních zhruba 20 vteřin tedy představuje neplatná data.

7 Shrnutí výsledků testování

V předchozí kapitole byl popsán průběh testování systému, zahrnující detekci srdeční frekvence v klidu, při chůzi rychlostí 3 km/h, 4 km/h a 6 km/h a při běhu rychlostí 8 km/h. Nejprve byla testována původní varianta filtrace a na základě výsledků tohoto testování byly provedeny úpravy filtrace a detekce srdeční frekvence. Po úpravách byl systém opět testován ve stejných podmínkách jako původní varianta systému.

Při testování původní varianty systému bylo zjištěno, že detekce srdeční frekvence probíhá korektně v klidu a při chůzi rychlostí 3 km/h a 4 km/h. I při rychlosti chůze 6 km/h se odchylka detekované srdeční frekvence stále pohybovala v rozumných mezích od skutečné frekvence vypočtené z EKG. K problémům s detekcí ovšem docházelo při běhu rychlostí 8 km/h. Detekovaná srdeční frekvence se sice opět pohybovala v hodnotách blízkých skutečné frekvenci, její průběh ale ani se zpožděním nekopíroval průběh skutečné frekvence. To bylo pravděpodobně způsobeno tím, že nebyla ve skutečnosti detekována srdeční frekvence ale frekvence pohybu, která se pohybovala okolo 2,9 Hz, což odpovídá detekované tepové frekvenci zhruba 170 tepů za minutu.

Nevýhodou této původní varianty systému byla také poměrně dlouhá doba výpočtu pohybující se okolo 7 vteřin, která způsobovala zaprvé značné zpoždění detekované frekvence za hodnotami vypočtenými z EKG, zadruhé pomalý náběh filtrace trvající okolo 30 vteřin. Další nevýhodou plynoucí z dlouhé doby výpočtu pak byla nízká frekvence výpočtu nových hodnot srdeční frekvence.

Aby se tyto nedostatky odstranily, byla filtrace přepracována a nová verze systému byla opět otestována. Výsledky tohoto testování ukázaly, že nová verze je schopná korektně detekovat srdeční frekvenci nejen v klidu a při chůzi rychlostí 3 km/h a 4 km/h, ale i při chůzi rychlostí 6 km/h či běhu rychlostí 8 km/h. Zvláště u běhu došlo k výraznému zlepšení detekce, kdy již byla detekována srdeční frekvence namísto frekvence rušení.

Nová verze systému také umožnila provádět výpočet výrazně rychleji, původní dobu výpočtu okolo 7 vteřin se podařilo snížit na 3,2 vteřiny. Tím se snížilo zpoždění vypočtené srdeční frekvence za skutečnou srdeční frekvenci a také se výrazně urychlil náběh filtrace, který se podařilo snížit ze 30 vteřin na méně než 10 vteřin.

8 Závěr

Cílem diplomové práce bylo vyvinout systém sloužící k odstranění pohybových artefaktů z PPG signálu, výpočtu srdeční frekvence a záznamu naměřených a vypočtených údajů na micro SD kartu a funkčnost celého systému následně otestovat.

Úvodní kapitoly práce byly věnované jak krátkému seznámení s EKG, tak především photoplethysmografii. Byl probrán princip funkce PPG senzoru a varianty jeho umístění, tvar PPG křivky a způsoby využití photoplethysmografie v lékařství. Dále jsme se věnovali vzniku pohybových artefaktů v PPG signálu a metodám jejich odstranění. Na základě předchozích prací zabývajících se touto problematikou bylo rozhodnuto použít filtraci založenou na stacionární vlnkové transformaci. Vzhledem k tomu, že hlavním zdrojem rušení PPG signálu je periodické rušení vzniklé pohyby pacienta jako je například chůze či běh, bylo rozhodnuto o využití akcelerometru k získání dat o pohybu pacienta a výpočtu frekvence tohoto pohybu. Na základě takto získaných údajů jsou pak nastavovány parametry filtrace. Bylo rozhodnuto umístit PPG senzor do ucha a připevnit k němu akcelerometr, aby se zajistilo, že rušení působící na PPG senzor bude zachyceno akcelerometrem.

Hlavní částí diplomové práce byl vývoje knihoven pro vývojový kit EvoPrimer a následně také vývoj samotné aplikace pro detekci srdeční frekvence. Vzhledem k tomu, že bylo nutné zaznamenávat naměřená data s dostatečnou vzorkovací frekvencí, bylo nejprve nutné vyřešit záznam na micro SD kartu. Protože dostupná řešení v podobě souborových systémů DosFs a FatFs neumožňovala dosáhnout požadované frekvence záznamu, bylo nutné přijít s jiným řešením. Bylo zvažováno několik řešení tohoto problému jako například záznam s vlastním jednoduchým souborovým systémem a PC aplikací sloužící ke čtení tohoto systému nebo záznam zcela bez souborového systému opět doplněný aplikací určenou pro PC. Nakonec byl ale vyvinut souborový systém optimalizovaný pro rychlý záznam dat vycházející z DosFs. Toto řešení umožnilo dosáhnout požadované frekvence záznamu, přičemž nebylo nutné vyvíjet dodatečnou PC aplikaci, protože tento souborový systém byl FAT-kompatibilní a tedy čitelný v podstatě na libovolném PC.

Dalším krokem byl vývoj knihovny pro získávání dat z akcelerometru a následné využití těchto dat k detekci nejvýznamnější frekvence rušení. Bylo zvažováno, zda detekovat frekvenci rušení ve frekvenční oblasti pomocí Fourierovy transformace nebo použít hledání vrcholů v časové oblasti. Po otestování obou variant byla zvolena detekce ve frekvenční oblasti, protože poskytovala spolehlivější výsledky. Po vyřešení záznamu na micro SD kartu a návrhu a implementaci knihovny pro využití akcelerometru bylo nutné implementovat samotnou filtraci PPG signálu založenou na stacionární vlnkové transformaci. Postup při filtraci je podrobně popsán v příslušné kapitole. Parametry filtrace jsou voleny na základě frekvence rušení vypočtené z dat získaných akcelerometrem.

Při testování funkčnosti vyvinutého systému bylo dosaženo poměrně dobrých výsledků při pohybu rychlostí do 6 kilometrů za hodinu, detekovaná srdeční frekvence se i při rychlosti 6 km/h lišila ve většině případů nejvýše o 10 pulzů. Při běhu bylo rušení již příliš intenzivní a srdeční frekvence se vyhodnocovala chybně. Na základě těchto zjištění byla filtrační metoda upravena a opět proběhlo testování, které podalo výrazně lepší výsledky zejména při detekci srdeční frekvence za rychlé chůze a běhu, kde se odchylka opět pohybovala ve valné většině případů do 10 pulzů.

Největší problémy při filtraci představovaly situace, kdy se kromě základní frekvence pohybu začaly projevovat také vyšší harmonické frekvence tohoto pohybu. Problémy ve výpočtu také nastávaly, pokud se srdeční frekvence blížila frekvenci rušení, nebo jeho harmonické frekvenci. U původní varianty filtrace se pak stávalo, že byla frekvence rušení vyhodnocena jako srdeční frekvence a došlo k chybnému výpočtu. Výskyt obou těchto problémů se podařilo potlačit po úpravě filtrační metody, nicméně se stále mohou za některé situace projevovat.

Hlavními omezeními při vývoji systému byl zaprvé nedostatek operační paměti na vývojovém kitu, zadruhé absence matematického koprocesoru. Je třeba si také uvědomit, že vyvinutý systém sice dokázal poměrně dobře detekovat srdeční frekvenci, není ho ale možné použít pro vyhodnocení parametrů závislých na tvaru PPG křivky, protože rušením a následným odfiltrováním co největšího množství nadbytečných informací ze signálu je zcela změněn tvar PPG křivky.

9 Literatura

- [1] Joseph D. Bronzino, *The Biomedical Engineering Handbook, Second Edition*, CRC Press LLC, 2000
- [2] Simon Arberet, Mathieu Lemay, Philippe Renevey, Josep Sola, Olivier Grossenbacher, Daniela Andries, Claudio Sartori, Mattia Bertschi, *Photoplethysmography-Based Ambulatory Heartbeat Monitoring Embedded into a Dedicated Bracelet*, Swiss Center for Electronics and Microtechnology (CSEM), Neuchatel, Switzerland, University Hospital Center of Lausanne (CHUV), Lausanne, Switzerland, 2013
- [3] Susana Aguiar Santos, Boudewijn Venema, *Accelerometer assisted PPG Measurement during Physical Exercise using the LAVIMO Sensor-System*, Philips Chair for Medical Information Technology, Helmholtz-Institute for Biomedical Engineering, RWTH Aachen University, Pauwelsstrasse 20, 52074 Aachen, Germany, 2012
- [4] Lukáš Paroulek, *Dlouhodobý monitoring srdečního tepu s využitím fotopletysmografického snímače*, ČVUT, 2013
- [5] R. J. E. Merry, *Wavelet Theory and Applications*, Eindhoven University of Technology, 2005
- [6] Liu Chun-Lin, *A Tutorial of the Wavelet Transform*, 2010
- [7] Ivan W. Selesnick, *Wavelet Transforms — A Quick Study*, Polytechnic University Brooklyn, NY, 2007
- [8] Dan Ellis, *Digital Signal Processing*, 2013
- [9] G. P. Nason, B. W. Silverman, *The Stationary Wavelet Transform and some Statistical Applications*, University of Bristol, 1995
- [10] YADHURAJ S. R, HARSHA. H, Dr. PADMAJA K. V, *REMOVAL OF NOISE IN PPG SIGNALS USING WAVELETS*, IJCSMC, 2013
- [11] RAISONANCE, *EvoPrimer for STM32F Performance Line User Manual*, RAISONANCE, 2012

- [12] RAISONANCE, *Open4 Base Manual*, RAISONANCE, 2010
- [13] RAISONANCE, *EvoPrimer for STM3240G Performance Line User Manual*, RAISONANCE, 2012
- [14] RAISONANCE, *CircleOS V4.6, Conception document*, RAISONANCE, 2013
- [15] RAISONANCE, *Raisonance Tools for ARM core-based microcontrollers*, RAISONANCE, 2013
- [16] Microsoft, *Microsoft Extensible Firmware Initiative FAT32 File System Specification*, Microsoft, 2000
- [17] Priscilla Oppenheimer, *File System Forensics*
- [18] Erich Styger, *FatFs: An Open Source File System*, Lucerne University of Applied Sciences and Arts, 2011
- [19] STMicroelectronics, *LIS331DLH MEMS digital output motion sensor — ultra low-power high performance 3-axes “nano” accelerometer*, STMicroelectronics, 2009
- [20] Ingrid Daubechies, *ORTHONORMAL BASES OF COMPACTLY SUPPORTED WAVELETS*, Society for Industrial and Applied Mathematics, 1993
- [21] Václav Hlaváč, Miloš Sedláček, *Zpracování signálu a obrazu*, Elektrotechnická fakulta ČVUT v Praze, 1999

Internetové stránky

- [22] Cortex™-M3 Technical Reference Manual - 5.3.2. Priority grouping, [dostupné 8.5.2014] [〈http://infocenter.arm.com〉](http://infocenter.arm.com)
- [23] How to config interrupts priorities for STM32, [dostupné 8.5.2014] [〈http://micromouseusa.com〉](http://micromouseusa.com)
- [24] Understanding FAT32 Filesystems [dostupné 9.5.2014] [〈http://www.pjrc.com/tech/8051/ide/fat32.html〉](http://www.pjrc.com/tech/8051/ide/fat32.html)

Použitý software

- [25] MATLAB, ver. 8.2.0.701, R2013b, The MathWorks, 2013
- [26] PDF-XCHANGE VIEWER, ver. 2.5, Tracker Software Products, 2012
- [27] Ride7, ver. 7.42.12.0305 Enterprise version, KEOLABS S.A.S, 2012
- [28] Rkit-ARM for Ride7, ver. 1.48.13.0029, Enterprise Suite, KEOLABS S.A.S, 2012
- [29] Inkscape, ver. 0.48.4 r9939, Open source
- [30] GIMP, ver. 2.8.10, Open source, 2014
- [31] WinHex, ver. 17.4, X-Ways Software Technology AG, 2013
- [32] Git Extensions, ver. 2.47.3, Henk Westhuis, 2013
- [33] Microsoft Visual Studio 2010 Professional, ver. 10.0.30319.1, Microsoft Corporation, 2010

Příloha A

Obsah přiloženého CD

K této práci je přiloženo CD, které obsahuje pdf soubor se samotnou prací a dále následující adresáře:

adresář HR_Monitor:	Původní verze filtrační aplikace
adresář HR_Monitor2:	Přepracovaná verze filtrační aplikace
adresář Matlab:	MATLAB skripty používané během vývoje a k ověření naměřených hodnot
adresář Data:	Průběhy naměřené během vývoje a testování systému