

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická
Katedra zadávající téma diplomové práce

BAKALÁŘSKÁ PRÁCE

Elektronický fonendoskop

Autor práce: **Vojtěch Bílek**
Vedoucí práce: **Prof. Ing. Milan Štork, CSc**

2023

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Vojtěch BÍLEK**
Osobní číslo: **E19B0077P**
Studijní program: **B2612 Elektrotechnika a informatika**
Téma práce: **Elektronický fonendoskop**
Zadávací katedra: **Katedra elektroniky a informačních technologií**

Zásady pro vypracování

Navrhněte a realizujte elektronický fonendoskop pro poslech pacienta. Systém by se skládal z miniaturního mikrofону, zesilovače a mikrokontroléru, který by umožňoval také grafické zobrazení vzorkovaného signálu. Akustický výstup by měl být též zachován.

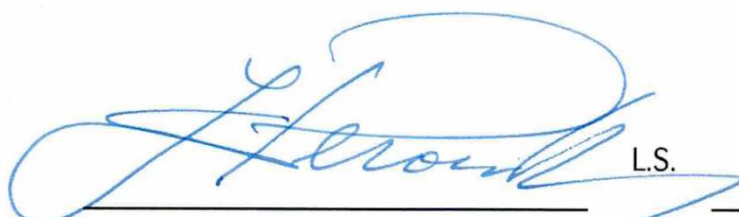
Rozsah bakalářské práce: **30 až 40**
Rozsah grafických prací: **5**
Forma zpracování bakalářské práce: **elektronická**

Seznam doporučené literatury:

1. Mikrofon [online], <http://cs.wikipedia.org/wiki/Mikrofon> .
2. IMRAMOVSKÝ, M. Zdravotnické elektronické přístroje I. Ostrava: VŠB skriptum, 2007.
3. Ru San Tan, Kevin Tshun Chuan Chai, Chao, The electronic stethoscope.
4. Supreeya Swarup, Amgad N Makaryus, Digital stethoscope technology update, https://www.researchgate.net/publication/322246148_Digital_stethoscope_Technology_update .

Vedoucí bakalářské práce: **Prof. Ing. Milan Štork, CSc.**
Katedra elektroniky a informačních technologií

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

Srdeční choroby jsou stále aktuálním problémem lidstva. Podstatná je jejich včasná detekce, která je již možná i pomocí levných moderních zařízení. Elektronický fonendoskop postavený na základě levného mikrokontroleru umožňující digitální zpracování signálu je hlavní úděl této práce.

Mikrokontroler ESP-32 zpracovává již digitální signál z inter integrated circuit kompatibilního mikrofону na bázi mikro elektromechanické struktury, který upraví pomocí digitálního filtru. Pracuje s 10místným průměrovým oknem, po součtu hodnot jednotlivých vzorků jej vydělí jejich počtem. Výsledkem je v případě okna dlouhého 10 vzorků dolní propust s mezi frekvencí na 2250 Hz. Srdeční signál je omezen na nižší frekvence, zbylé rušivé nad 2250 Hz jsou odstraněny.

Zaznamenaný signál je pomocí inter integrated circuit kompatibilního zesilovače přehráván přes pasivní reproduktor. Připojená SD karta pomocí sériového periferního interface umožňuje uložení 10 sekund záznamu. Aktivní wifi server mikrokontroleru pak soubor pomocí webové stránky umožňuje stáhnout.

Získaný signál svými parametry připomíná profesionální záznam fonokardiogramu, avšak jeho kvalita nemůže být bez odborníka v medicíně plně vyhodnocena. I přesto lze říct, že výsledný signál je úspěchem jak z důvodu zadání, tak tím, že i přes malou cenu součástek lze zaznamenat laicky použitelný signál.

Klíčová slova

Elektronický fonendoskop, digitální filtr, mikrokontroler, srdeční choroby, auskultace

Abstract

Heart diseases remain a pressing issue in human civilization, emphasizing the importance of early detection. Current technologies, even the cheap ones, are already capable of detecting such diseases. Electronical phonendoscope built on a microcontroller unit that digitalizes the sound of human heart is the main idea of this thesis.

The microcontroller esp32 obtains a digital audio signal from the inter-integrated circuit microphone that works on the principle of micro electromechanical structure and uses a digital filter on it. The filter works on an averaging window the size of 10 samples. After all the samples are summed up, they are divided by 10. The output filter with parameters of lowpass with the upper limit of 2250 Hz. Heart produces deeper frequencies than that so only the rest of the signal is filtered out.

The recorded signal is played back through inter-integrated circuit amplifier with a passive speaker connected to it. Serial peripheral interface mounted SD card adds the function of saving the signal in a 10 second wave file. The Wi-Fi server running on the microcontroller's second core lets the user show the SD card directory and download a selected file.

The output signal has similar parameters of a professional phonocardiogram recording, although the quality cannot be fully evaluated without the presence of an expert in this field. Even though it can be said that the recorded signal is a success of this thesis as it got a somewhat useful signal with the low cost of used electronics.

Key Words

Electronic phonendoscope, digital filter, microcontroller, heart disease, auscultation

Poděkování

Rád bych poděkoval svému vedoucímu bakalářské práce panu profesoru Milanu Štorkovi za pomoc při kontrole smysluplnosti obsahu a detekci nesprávných úvah, které by jinak práci ubraly na ucelenosti.

Obsah

1	Úvod.....	- 1 -
2	Kardiologická část	- 2 -
2.1	Činnost lidského srdce.....	- 2 -
2.2	Měření lidského srdce.....	- 4 -
2.2.1	Elektrokardiogram ECG.....	- 4 -
2.2.2	Fotopletysmografie PPG	- 5 -
2.2.3	Fonokardiogram PCG	- 6 -
2.2.1	Lékařský fonendoskop	- 7 -
2.2.2	Porovnání elektronických fonendoskopů na trhu.....	- 9 -
2.2.3	Průběh signálů akustického záznamu.....	- 11 -
3	Elektronická část.....	- 12 -
3.1	Mikrokontroler ESP-32	- 12 -
3.2	Komunikační protokoly.....	- 13 -
3.2.1	I ² S	- 13 -
3.2.2	Direct memory access DMA.....	- 14 -
3.2.3	Serial Peripheral Interface SPI.....	- 15 -
4	Zpracování elektronické části	- 16 -
4.1	Mikrofon.....	- 16 -
4.1.1	Volba mikrofonu	- 16 -
4.1.2	Konfigurace periferie pro použití MEMS mikrofonu	- 20 -
4.2	Efektivní bitová hloubka	- 23 -
4.3	Problém hlasitosti signálu.....	- 24 -
4.4	Digitální filtry	- 26 -
4.4.1	Zpoždění způsobené výpočtem při filtrování a zesílení.....	- 26 -
4.4.2	Zpracování signálu ve funkci	- 31 -
4.4.3	Průměrový filtr	- 33 -
4.5	Průběhy zvukových záznamů	- 36 -
4.6	Záznam akustického výstupu.....	- 38 -
4.6.1	Zvukový soubor WAVE.....	- 38 -

4.6.2	Kódové řešení záznamu	- 39 -
4.7	Wifi server	- 40 -
4.8	Srovnání získaných průběhů.....	- 41 -
5	Zhodnocení a závěr	- 44 -
	Literatura.....	- 46 -

Seznam symbolů a zkratek

Elektronické zkratky

MCU	<i>jednotka mikrokontroleru</i>
SOC	<i>system-on-chip</i>
I²S	<i>inter-integrated circuit sound</i>
SPI	<i>sériový periferní interface</i>
DMA	<i>přímý přístup do paměti</i>
MEMS	<i>mikro elektro-mechanická struktura</i>

Kardiologické zkratky

S1, S2	<i>srdeční ozvy</i>
EKG	<i>elektro kardiogram</i>
PPG	<i>foto pletysmografie</i>
PCG	<i>fono kardiogram</i>
Auskultace	<i>poslech vnitřních zvuků těla</i>

Akustické pole

SQNR_{dB}	dB	<i>poměr signál šum</i>
--------------------------	-----------	-------------------------

Elektrické veličiny

U	V	<i>elektrické napětí</i>
I	A	<i>elektrický proud</i>

Časová a frekvenční oblast

T_r	s	<i>perioda čtení</i>
T_{pr}	s	<i>perioda zpracování</i>
f_s	Hz	<i>vzorkovací frekvence</i>
T_{pr}	s	<i>perioda zpracování dat</i>

1 Úvod

Náplní práce je vývoj a implementace elektronického fonendoskopu na bázi mikrokontroleru. Fonendoskop je základním nástrojem pro auskultaci neboli poslech srdce, plic a dalších tělesných zvuků, který je využíván při diagnostice a hodnocení stavu pacientů. V současné době existují ve většině tradiční akustické fonendoskopy, které vyžadují zkušeného lékaře nebo zdravotního pracovníka s dobrou sluchovou schopností. Elektronický fonendoskop představuje moderní technologický pokrok, který umožňuje snadnější poslech zvuků a přesnější diagnostiku.

Hlavní problém, který se v práci řeší, je omezená slyšitelnost a rozpoznatelnost tělesných zvuků pomocí tradičních akustických fonendoskopů. Některé tělesné zvuky mohou být velmi slabé nebo obtížně rozpoznatelné a jejich správná interpretace může být pro lékaře náročná. Elektronický fonendoskop se zaměřuje na tyto problémy a snaží se poskytnout lékařům a zdravotním pracovníkům nástroj, který zlepší kvalitu a přesnost jejich diagnostiky.

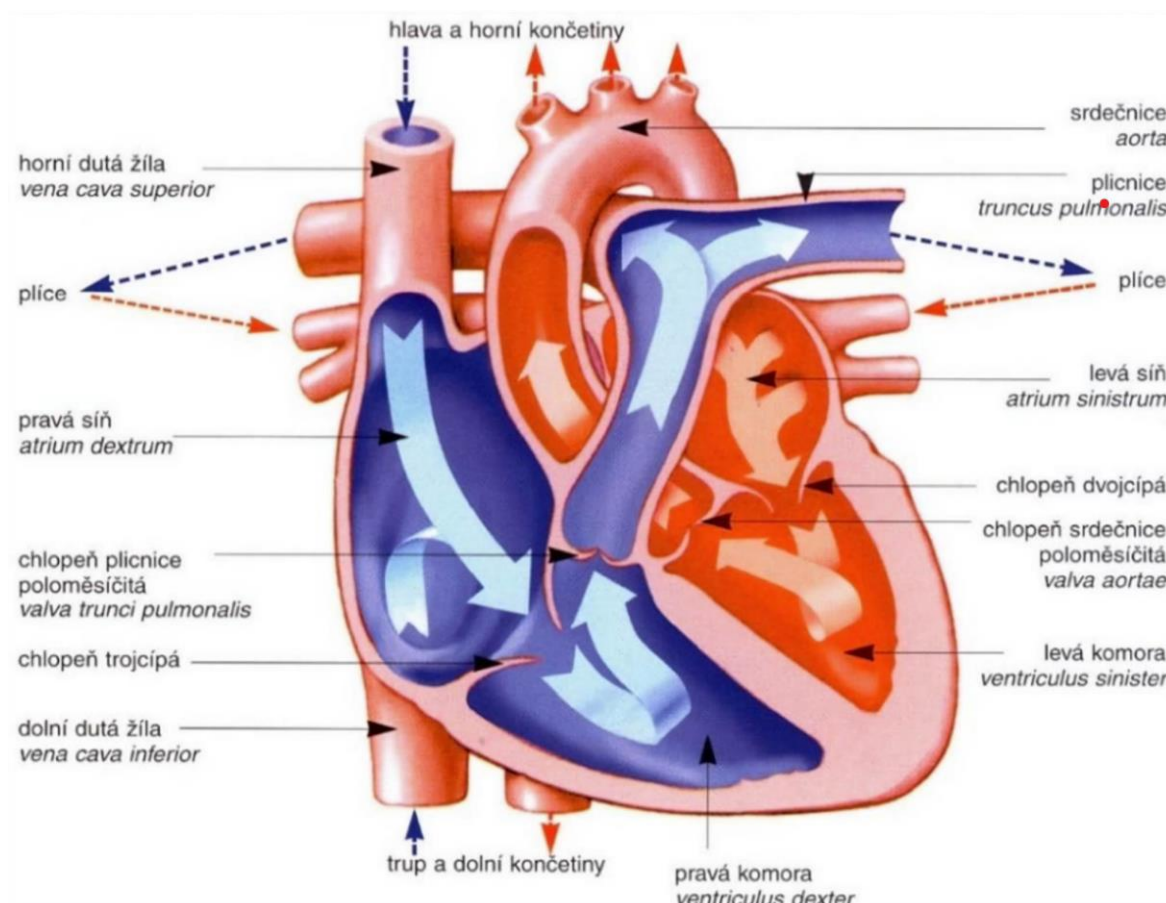
Záměrem této práce je inovace v oblasti auskultace a využití moderních technologií v lékařství, která se i přes několik málo dostupných produktů na trhu nedostává k masovému použití. Elektronický fonendoskop přináší možnost zaznamenávat a analyzovat zvukové signály, poskytovat vizualizaci zvuků a umožnit jejich další zpracování a interpretaci. Cílem je tedy navrhnout a implementovat elektronický fonendoskop na bázi mikrokontroleru, který bude přesný, snadno ovladatelný a přizpůsobitelný potřebám lékařů.

2 Kardiologická část

2.1 Činnost lidského srdce

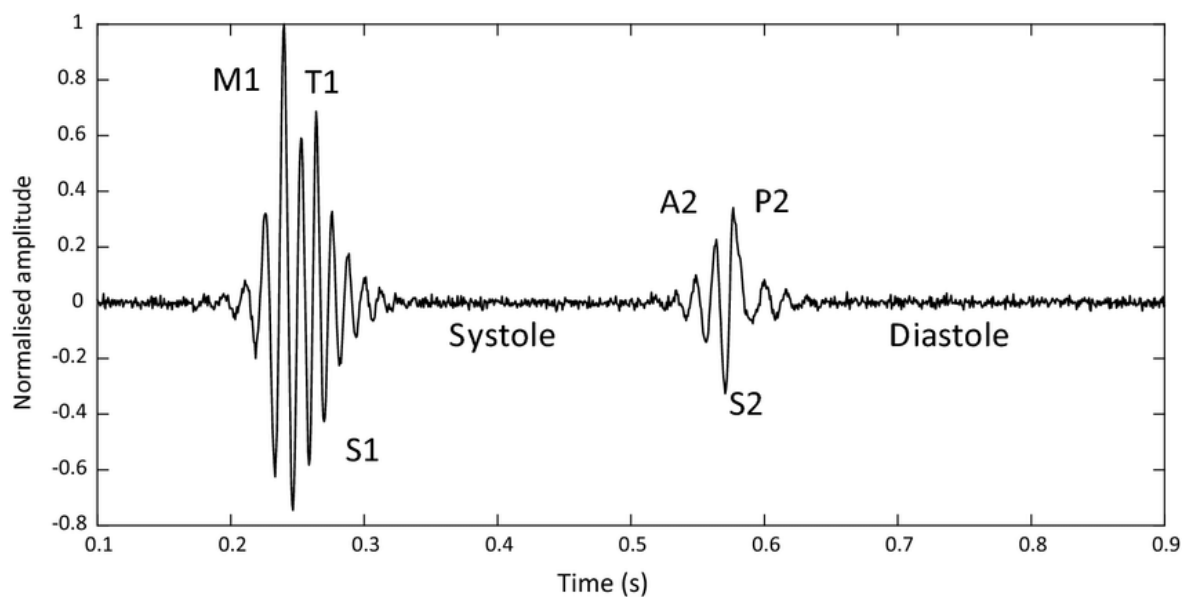
Lidské srdce je sval, který je řízen vlastními impulsy generovanými v sinoatriálním uzlu a vedených přes speciální typ kardiomyocytů, buněk uvnitř srdeční svaloviny a hlavní části srdeční stěny – myokardu. Ten obklopují ještě další 2 vnější vrstvy tkáně – epikard a perikard, zevnitř pak jedna vrstva – endokard.

Celkem má tento orgán 4 vnitřní dutiny, a to levou a pravou komoru a síň. Odkysličená krev se dostane do pravé síně, odtud se při diastole (uvolnění srdce) vlije do pravé komory skrze trojcípou chlopeň. Systolou (kontrakcí srdce) se díky uzavření chlopně pod tlakem vlije dále do plicní tepny a následně v plicích okyslíčí. Nově okysličená krev se plicní žilou dostává zpět do srdce, a to konkrétně do levé síně, odtud se stejně jako v pravé straně při diastole vlije do levé komory odkud je následně při systole pod tlakem vháněna do aorty, která krev rozvádí do celého těla. Mezi jednotlivými komorami jsou srdeční chlopně, které při zpětném tlaku vzájemně dolehnou a uzavřou otvor, kterým by v opačném směru krev proudila. Právě díky těmto prudkým domykáním chlopní a rychlým prouděním krve kolem stěn je srdeční činnost slyšitelná a zle pomocí srdečních ozev diagnostikovat případné neregulérnosti srdeční činnosti.



Obrázek 1 Průtok krve srdcem

Primárně lze rozlišit celkem 4 srdeční ozvy. Ve chvíli uzavření obou cípatých chlopní (téměř najednou) se charakterizuje jako S1. Druhá ozva S2 opět vzniká dovřením dvou měsíčkovitých chlopní, a to konkrétně na konci systolické fáze. Dále jsou dvě velmi jemné ozvy, které plynou u S3 z oscilace stěn komor po jejich prudkém naplnění. A S4 jako kontrakci síní. Pro poslední dvě ozvy je z důvodu jejich poměrově malé hlasitosti potřeba lokalizovat ideální poslechový bod na hrudníku, případně využít fonograf pro zobrazení průběhu a graficky je lokalizovat.



Obrázek 2 Typický průběh akustického záznamu se srdečními ozvy S1 a S2, mezi nimi systola a diastola

2.2 Měření lidského srdce

2.2.1 Elektrokardiogram ECG

Jedním z hlavních signálů, které srdce generuje je elektrický potenciál napříč jeho svalovou tkání při jejím stahu. Svalová buňka, tak jako mnoho dalších buněk v těle, funguje na principu membránového potenciálu vyvolaného polarizací biologické membrány buňky. Na jedné straně membrány dojde k nastřádání kationtů a na druhé aniontů skrze kanály v její stěně. Z fyzikálního hlediska jde o elektrické pole vyvolané elektrochemickým gradientem. Takto polarizovaná, negativně nabitá buňka je připravena na depolarizaci, a tedy v případě srdečního svalu na stah. Při šíření depolarizační vlny podél svalu je možné si sval představit jako elektrický dipól, který na polarizované straně obsahuje stále nabitě anionty a jeví se tak jako negativní pól. Druhá, již depolarizovaná část svalu uvolnila anionty a svůj el. náboj zvýšila oproti protějšku, působí tak jako kladný pól dipólu. V moment šíření vlny depolarizace je tedy možné pomocí elektrod vhodně rozmístěných po těle odečíst elektrické napětí a v časové ose zobrazit jeho intenzitu. Opačným dějem-repolarizací, je buňka opět nabita el. nábojem a připravena k dalšímu cyklu. Stejně jako u depolarizace, i zde dojde k jevu elektrického dipólu, avšak s opačným směrem el. pole. Pro nejsilnější signál je třeba elektrody umístit tak, aby přímka mezi nimi byla rovnoběžná s vektorem elektrické intenzity. Jelikož každá část srdečního svalu má šíření depolarizační vlny v jiném směru, je

třeba umístit až 12 elektrod, poté každá dvojice snímá depolarizační vlny jiného směru a tím i jiné části srdce. Výsledným spojením signálů získáme průběh celého srdečního cyklu.

Měření probíhá neinvazivní metodou – do těla není nijak zasahováno, ale přesto se jedná o koordinačně složitější postup. Čtyři elektrody se umísťují na končetiny, zbytek k hrudníku v oblasti srdce. Každou elektrodu je nutné vyvést do elektrokardiografu, vyšetření však trvá jen několik sekund. Nejčastější vyšetření prováděná ECG jsou z důvodu dušnosti, bolesti na hrudi, pocitu bušení srdce, případně i při zvětšení srdečních oddílů.

2.2.2 Fotopletysmografie PPG

Metoda měření pulsu pomocí relativně intenzivního zdroje světla, které musí dostatečně hluboko prosvítit pokožku v místě měření. Princip spočívá v detekci změn odraženého světla, které se vrátí do senzoru, typicky umístěného těsně vedle zdroje, případně naproti němu v závislosti na aplikaci (u hrudního snímače to není možné, u prstu naopak je). Díky tomu, že krev pohlcuje světlo více, než běžná tkáň, je možné poznat tlakovou vlnu vyvolanou stahem srdečních svalů (systola) šířící se krevním oběhem. V moment, kdy se cévy roztáhnou při nárustu tlaku krve naroste i objem a klesne snímaná intenzita odraženého světla. Naopak při druhé části činnosti srdce (diastola) je pokles tlaku, tedy i objemu krve v cévách, a tedy i větší odražená intenzita do světelného snímače. Pomocí této metody lze kontrolovat srdeční činnost velmi jednoduchým optoelektronickým zařízením. Nutno dodat, že pro kvalitní měření například i při pohybu je velmi důležité zpracování odchylek signálu způsobených změnou tlaku při pohybech nebo vliv okolního světla. Pro měření tepu lze použít červené světlo, ale také více používané zelené, které má sice menší hloubku průniku, ale krev je pro něj prakticky neprůsvitná. V případě měření SpO_2 okysličení krve je důležité použít infračervené světlo, které se odráží v závislosti na krevní saturaci kyslíkem. Vhodné je mluvit spíše o odhadu koncentrace než přímo o jejím měření, ale pro rychlé, neinvazivní určení přibližné hodnoty je velmi používané.

Odvozená varianta interpretace dat APG (acceleration PPG) je pomocí druhé derivace signálu. Průběh poté v maximech a minimech zobrazuje místa kde se původní signál mění z konkávního tvaru na konvexní. Akcelerační PPG tedy zobrazuje akceleraci krve, dynamiku změn tepu. Díky této metodě lze určit srdeční choroby, které by bez derivací signálu nemusely být patrné.

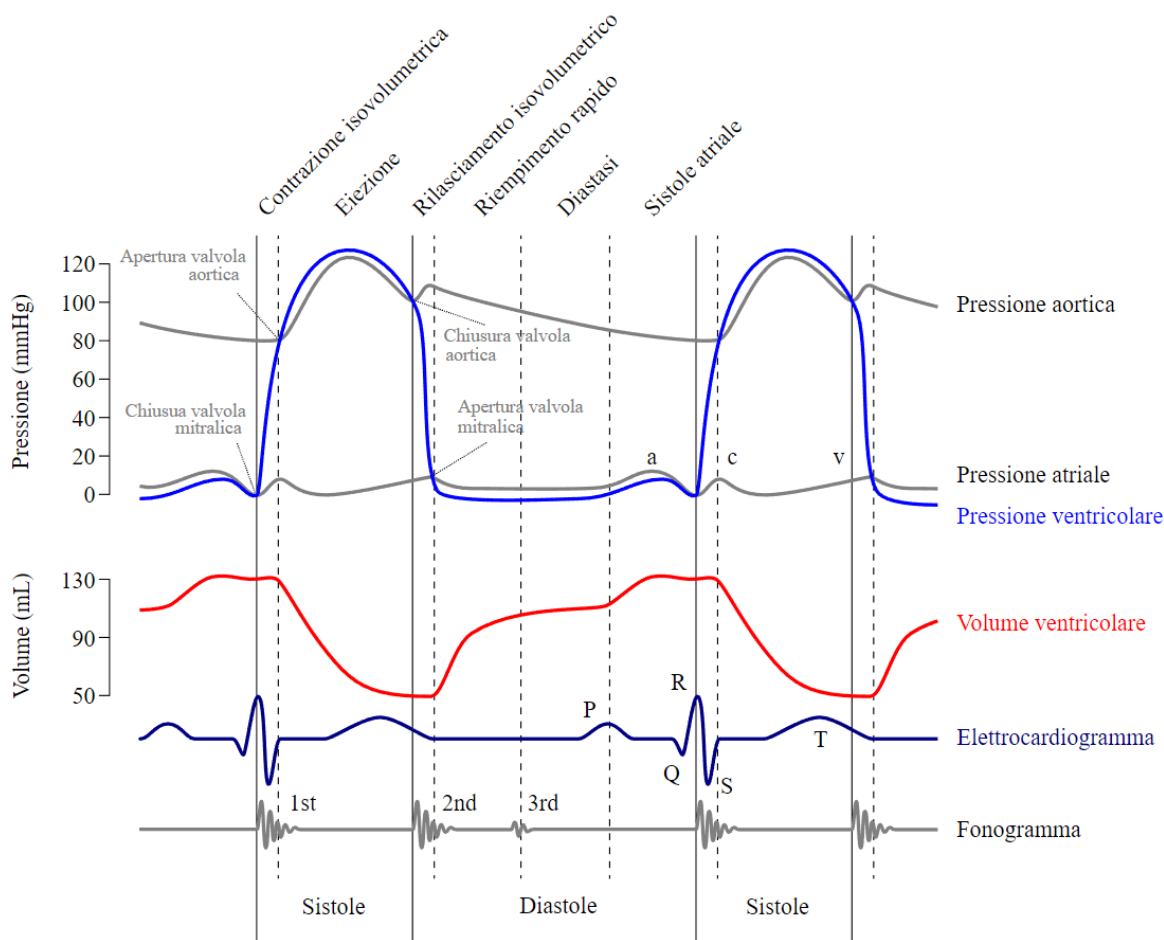
2.2.3 Fonokardiogram PCG

Fonokardiogram je zařízení pro snímání a záznam zvuků produkovaných srdcem. Tyto zvuky vznikají prouděním krve, a to při změně její rychlosti nebo směru, při zavírání a otevírání chlopní, jednotlivým přeléváním mezi komorami a zbylými turbulentními dynamickými toky. Zaznamenaný průběh zvuku se zobrazuje v grafu s časovou vodorovnou osou a amplitudou signálu. V RAW režimu se přehrává veškerý zaznamenaný plno spektrální signál. Pro čitelnější zobrazení se však používá signálové zpracování procesorem, který v závislosti na různých parametrech reprezentuje signál.

Základním typem úpravy signálu může být pouze vyčištění šumu, které vzniká během konverze akustického signálu na elektronický ale také během průchodu signálovou cestou. Dále je vhodné izolování pouze těch zvuků, které jsou generovány srdeční činností. Toho lze dosáhnout vhodným frekvenčním omezením v pásmu, kde zvuky vydávají ostatní orgány. Například 20-1000 Hz, je však nutné uvažovat s možnou ztrátou informací při abnormální srdeční činnosti, například šelestech. Je tak nutné najít správný balanc, kdy propouštěné pásmo odpovídá tomu, na co se zaměřujeme. Proto je možné filtr frekvenční propustit ladit za chodu tak, aby neomezoval rozlišovací schopnost, ale na druhou stranu nepropouštěl příliš mnoho zvukových ruchů. Praktickou možností je například simulovat funkci hlavy fonendoskopu. Ta z jedné strany přes membránu přijímá širší pásmo z vyšších frekvencí a z druhé strany přes konický tvar propouští frekvence nižší.

Další možnosti zpracování signálu se hluboce zaobírají jejich komplexním matematickým rozborem. Porovnávají jednotlivé amplitudy různých složek, vyhodnocují průměrné trvání polarit v jedné složce a podobně. Jedním z případů je empirická modální kompozice, která se vzdáleně podobá Fourierově transformaci, avšak složky signálů nerozkládá na čisté sinové signály – není to předpoklad rozkladu periodického signálu. Modální funkce, které tímto rozkladem vznikají jsou založeny na maximech a minimech a jejich vzájemném časovém odstupu.

Pro detekci srdečních onemocnění pomocí analyzačního programu lze také využít neuronové sítě, které si z již vyhodnocených dostupných dat vytvoří síť různých parametrů, kterým přiřadí adekvátně naučené významové hodnoty a dle nich potom vyhodnocují měřené signály. Obdobnou metodou strojového učení může být metoda podpůrných vektorů-SVM, která však pracuje i s nelineárními prostory a jejím úkolem je nalézt podpůrné vektory, které ohraničují optimální nadrovinu. Ta by měla rozdělovat trénovací data tak, že parametry odlišné třídy leží v opačném poloprostoru. Tato SVM metoda vyniká možností obecné klasifikace biomedicínských signálů.



Obrázek 3 Porovnání čtyř výstupů srdečních cyklů

2.2.1 Lékařský fonendoskop

Stetoskopem se dříve nazývalo zvukovodné ústrojí, kterým se zvuk pouze přiváděl k uživateli, fonendoskopem se označovalo zařízení s membránou, které se přikládá (mimo jiné) na kůži pacienta a efektivně vyvádí zvukové vlny ven z těla a trubicí až do uší uživatele. Tento jemný rozdíl se však dnes neuvažuje a setkáme se v běžné praxi s oběma slovy – dnes mají jeden a ten samý význam.

Již zmíněná část s membránou se nazývá hlava a zastává hlavní roli. Kvůli nutnosti poslechu více druhů zvuků těla pacienta bylo třeba zařídit získání a izolování správného frekvenčního spektra, které izoluje jednotlivé zdroje zvuku. Proto má dnes fonendoskop jednu větší membránu – bubínek, která zajišťuje dobrý přenos středních a vyšších frekvencí a druhou stranu – zvon, který pouze směřuje nižší frekvence dovnitř dutiny fonendoskopu bez požití membrány. Moderní zařízení disponují pouze jednou poslechovou stranou, která pomocí variabilního napnutí membrány modulací přtlaku zajišťuje změnu frekvenčního

pásma, které se přenáší skrze hlavici dále do zvukovodné trubice. Typicky má fonendoskop dvě olivky, které se jako sluchátka zasunou do uší. Pro správný přenos zvuku, hlavně spodní části spektra, je třeba zajistit správné utěsnění v uchu.

Klasická analogová varianta je dnes stále nejpoužívanější, hlavně díky své jednoduchosti a malé údržbě. Jsou ale případy, kdy je elektronická varianta lepší volbou. Zvláště pro lékaře se zhoršeným sluchem je dodatečné zesílení signálu klíčové. Další varianta je například při nutnosti poslechu mimo oddělení, tedy mimo (jinak potřebné) ticho. Díky aktivnímu potlačení hluku je možné provést kvalitní poslech například v rušném místě akutního zásahu. Úplnou novinkou je pro elektronické fonendoskopy možnost sdílení nahrávek. V případě, že si pacient sám provede záznam může nahrávku sdílet s lékařem. Tato varianta telemedicíny se v dnešním světě čím dál více rozvíjí a může dále navazovat na již zavedené domácí měření pomocí zapůjčeného přístroje, který sbírá data a po určitém časovém období jsou při vrácení analyzována. V případě použití například fonendoskopu s připojením na internet by mohl pacient svůj pořízený záznam rovnou odeslat na server přístupný lékaři a ten by tak mohl rovnou vyhodnotit případnou komplikaci.



Obrázek 4 Lékařský fonendoskop

2.2.2 Porovnání elektronických fonendoskopů na trhu

Velmi moderní a úsporné řešení elektronického fonendoskopu nabízí firma Littmann. Zachovala všechny typické prvky analogového zařízení a k již hotovému fonendoskopu pouze přidala elektronický mezičlánek, který se stará o veškeré zpracování a následnou reprodukci získaného signálu, který posílá dále skrze zvukovodné trubice do binaurálního nástavce. Zároveň ale neblokuje případný analogový chod fonendoskopu. Zařízení dokáže zesílit získaný zvuk až 40krát, a zároveň poskytnout aktivní potlačení okolního hluku. Integrací Bluetooth dovoluje živý přenos zvuku do chytrého telefonu, který lze přehrát v Bluetooth sluchátcích, a zobrazení průběhu amplitudy zvuku v podobě fonokardiogramu PCG na jeho displeji. Dále umožňuje nahrávání až 2minutových nahrávek, které je možné později lékařsky evaluovat, nebo rovnou odeslat lékaři v případě akutní nevolnosti. Implementace AI v aplikaci telefonu dovoluje automatické testování na srdeční šelest. Díky klasické analogové hlavě fonendoskopu zajišťuje frekvenční laditelnost poslechu pomocí variabilního přitlaku membrány. Připojený telefon tak nahrazuje nedostatky samotného jádra, které disponuje pouze 4 fyzickými tlačítky a USB konektorem. Veškeré nastavení zpracování zvuku tedy probíhá skrze telefon, který se stará o maximální využití potenciálu elektronického fonendoskopu.



Obrázek 5 Digitální stetoskop firmy Littman

Opačný přístup zvolila firma ThinkLabs, která zastává čistě elektronický postup a z analogového fonendoskopu si ponechala pouze tvar hlavy, která se přikládá na tělo pacienta. Dokonce i samotnou membránu přepracovala tak, že její průhyb přímo produkuje elektrický signál. Jedná se o princip kondenzátorového mikrofону, přičemž vnější poslechová membrána je přímo spojena s jednou z jeho elektrod. Jakákoliv změna vyvolaná silou působící na membránu, vyvolá určitý signál na nabitých elektrodách díky změně kapacity mezi nimi. Dochází tedy k přímému převedení zvuků z těla na elektrický signál, zařízení proto nelze používat v analogovém režimu jako předchozí popsanou variantu. Zdánlivou nevýhodu však kompenzuje svojí integrací v kruhové kapsli, jež není o moc větší než běžná hlava fonendoskopu. Z druhé strany kapsle je malý displej, který umožňuje kontrolu nastavených parametrů poslechu. K jejich ovládání slouží 4 mechanická tlačítka. Jelikož se membrána chová frekvenčně stále, zařízení umožňuje elektronické nastavení poslechového frekvenčního spektra. Pomocí ukazatele lze posouvat frekvenční propust napříč frekvenčním spektrem a případně přepnout na celé spektrum. Oproti předchozímu příkladu nedisponuje bezdrátovým přenosem, a tak je využití displeje telefonu podmíněno kabelovým propojením přes jack konektor. Aplikace pro chytrý telefon opět dovoluje zobrazení přehrávaného zvuku stylem PCG, dále také záznam a jeho sdílení s doktorem-telemedicína. Pro reprodukci zvuku používá dva nezávislé mono zesilovače pro každý kanál, ačkoliv se jedná o mono záznam. Zvuk pak pomocí jack konektoru lze přehrát v libovolných kabelových sluchátcích. Pro simultánní poslech i záznam je nutné využít přibalenou jack to jack rozdvojku.



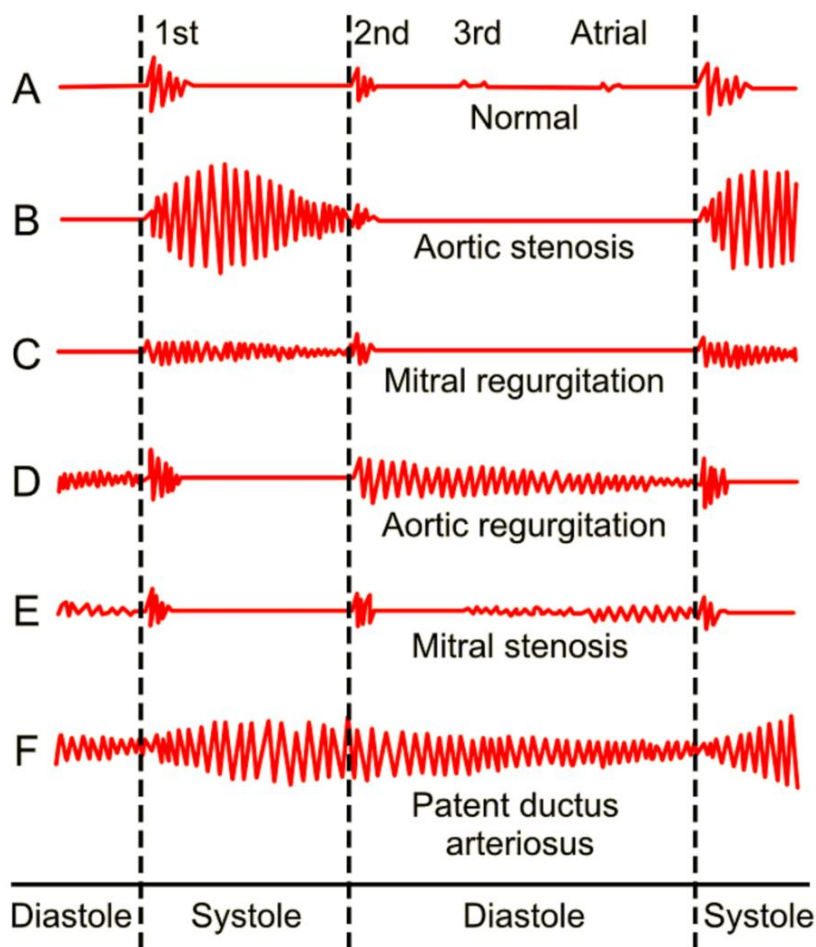
Obrázek 6 Digitální stetoskop ThinkLabs

2.2.3 Průběh signálů akustického záznamu

Z Průběhu zaznamenaného signálu můžeme opticky rozlišit, zda se jedná o typický průběh, nebo o podezření na srdeční chorobu. V případě zdravého srdce jsou obě srdeční ozvy S1 a S2 (1st a 2nd) odděleny časovým úsekem bez signálu. V reálném případě je signál stále přítomen, avšak jeho výkmit je vůči srdečním ozvám minimální.

V případě srdečního onemocnění dochází k produkci zvuku i mezi oběma ozvami, případy jsou delší, mají deformovaný průběh, nebo nejsou vůbec odděleny.

- **Aortální stenóza**, zúžení aortální chlopně
- **Mitrální regurgitace**, nedomykavost mitrální chlopně
- **Aortální regurgitace**, nedomykavost aortální chlopně
- **Mitrální stenóza**, zúžení mitrální chlopně
- **Otevřená Botallova tepenná dučej**, špatné poporodní uzavření cévy novorozence spojující plicnici a aortu která zkracuje cestu okysličené krvi dodávané plodu pupeční šňůrou.



Obrázek 7 Průběhy signálu srdečních chorob ve fonokardiogramu

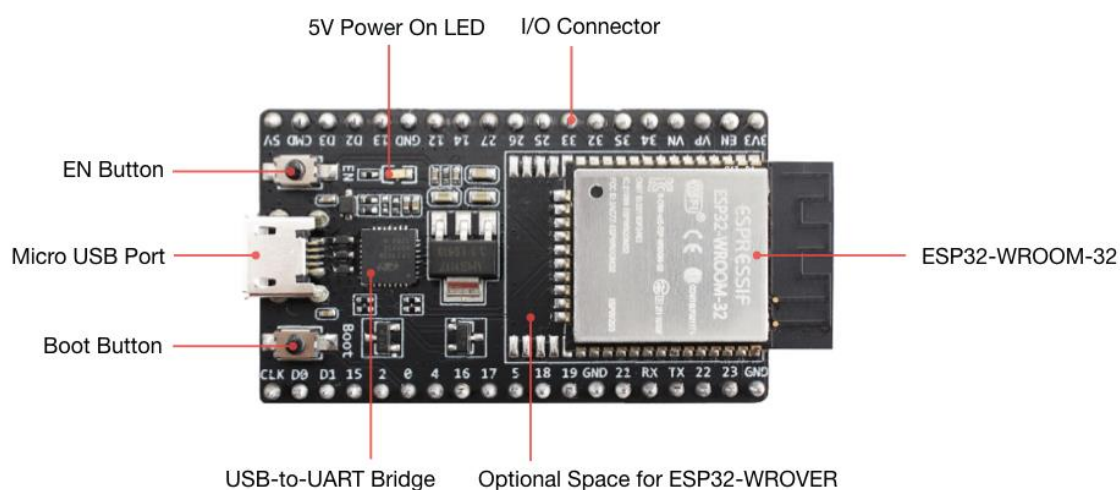
3 Elektronická část

3.1 Mikrokontroler ESP-32

ESP32 je moderní SoC vyráběný firmou Espressif, který integruje mnoho různých funkcí a periférií do jednoho malého modulu velikosti mince. Jedná se o mikrokontroler s 32 bit architekturou 40nm technologií dvoujádrového procesoru s pracovní frekvencí až 240Mhz a 512KB RAM. Hlavní předností je bezdrátová komunikace, kterou zvládá jak přes Bluetooth, tak i Wi-Fi, takže se výborně hodí pro IoT projekty (Internet of Things).

Jako nástupce předchozího modelu ESP8266 nabízí nejen větší počet GPIO pinů - 40, ale také přidává nové integrované funkce, jako například analogové měření na 18 kanálech, druhé logické jádro nebo Bluetooth komunikaci. S novými funkcemi se zvětšila i cena, ale i tak se stále jedná o zařízení s cenou okolo dvou set korun. Díky většímu počtu pinů se také rozrostli možnosti vývojových desek, které uživatelům usnadňují práci s mikrokontrolerem a zjednodušují vývoj zařízení.

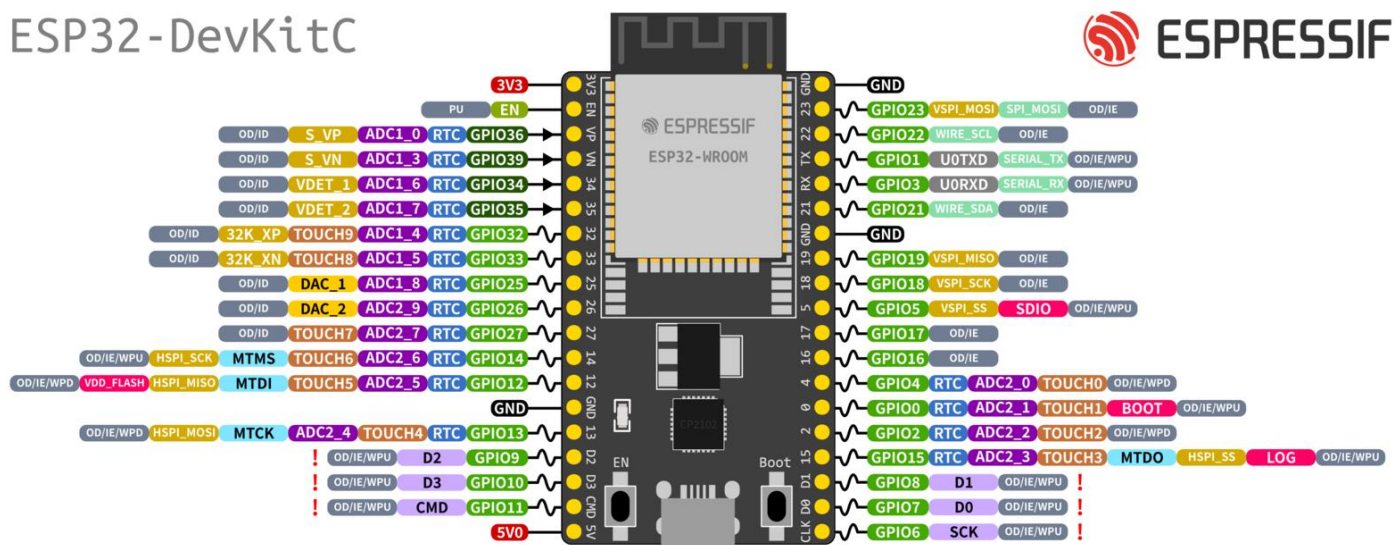
Espressif nabízí desítky variant čipů s různými parametry a také desítky vývojových modulů, které rovnou implementují potřebné periferní obvody, např. zabudované displeje, kamery, audio zpracování, nebo IoT interface. Většina variant modulů má napájení přes USB port, nějaké moduly však rovnou obsahují konektor na baterii, takže se výborně hodí na přenosné zařízení. Modul obsahuje i regulátor napětí na 3,3 V, takže je možné dále napájet různorodé moduly.



Obrázek 8 Vývojový modul eps32 wroom

ESP32 nabízí širokou škálu možností skrze GPIO piny. Výhodou je také možnost multiplexu u pinu, tedy na jednom GPIO kontaktu lze připojit více zařízení současně.

ESP32-DevKitC

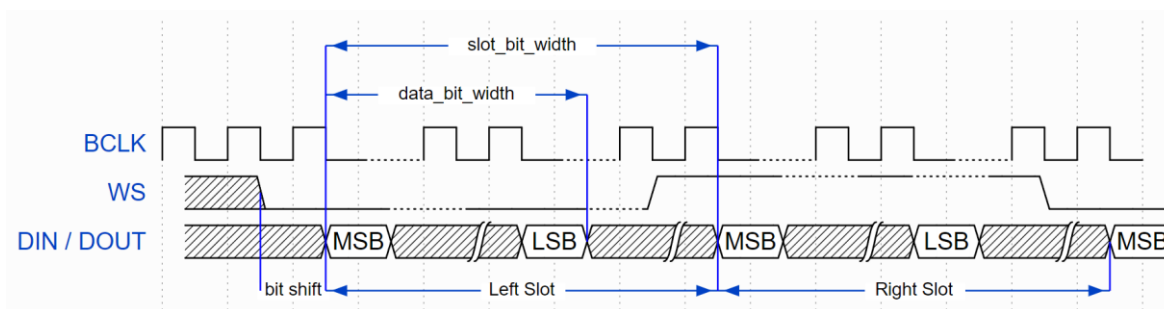


Obrázek 9 Piny vývojové desky DevkitC s mikrokontrolerem ESP-32

3.2 Komunikační protokoly

3.2.1 I²S

I²S neboli inter-IC sound, je typ sériové sběrnice, která se používá pro přenos audio dat v digitální podobě mezi zařízeními. Synchronní komunikace v režimu Standart nebo TDM probíhá přes tři vodiče. Hodinový impuls BCLK – bit clock line dává informaci o každém datovém bitu dostupném na datovém pinu. Je generovaný z MCLK – master clock frequency a jeho frekvence závisí na bitové hloubce vzorků, například pro 16bit rozlišení bude 16 tiků v jednom datovém 16bit slotu. WS – word select line slouží pro určení začátku bitového slova a pro selekci mezi zvukovými kanály. Poslední je datová linka, a to DIN - data in nebo DOUT-data out po které se přenáší samotná zvuková data. Jak I²S, tak I²C-inter integrated circuit jsou protokoly vyvinuté firmou Philips Semiconductors, stejný název nese i základní režim přenosu dat-Standart Phillips format, který má datový slot posunutý o jeden takt vůči hraně WS se střídou 50:50, tedy polovičním rozdělením pro každý ze stereo kanálů.



Obrázek 10 Časování i2s komunikačního protokolu Phillips

I2S nabízí i další formáty pro přenos zvukových dat: MSB, PCM Short a PDM v režimu TX i RX. Dále ale také nabízí režim LCD/Camera který přes paralelní sběrnici poskytuje komunikaci pro přenos obrazových dat.

ESP32 nabízí také subrežim ADC/DAC, který vyhází z LCD/Camera režimu. Ten dovoluje přímé propojení na interní obousměrný DAC, a tak mohou periferie s DAC nebo ADC převodníkem přímo zapisovat nebo číst z I2S kanálu skrze přímý přístup do paměti DMA.

3.2.2 Direct memory access DMA

DMA je základem transportu dat I2S protokolu. Jedná se o kanál, který umožňuje zrychlené operace přesunu dat mezi dvěma datovými prostory, ať už uvnitř jedné paměti, nebo mezi dvěma zařízeními. Veškerý přenos zajišťují hardwarové prostředky a procesor se tím tak nezatěžuje. V opačném případě, využití procesoru a zajištění operace přesunu pomocí kódu, by doba zpracování znatelně narostla. Pro úspěšný DMA přenos je nutná rezervace registrů v procesoru, které pracují s adresy zdroje a cíle, počty přenosů a dočasným uložením dat. V případě konfigurace několika kanálů do jednoho bloku má každý kanál vlastní 4 registry. Celek se všemi kanály i jejich řízením nazýváme řadič DMA.

ESP32 nabízí dva vzájemně nezávislé periferní I2S kanály – I2S_NUM0 a I2S_NUM1 které se pro jednotlivé použití definují na začátku kódu. To dovoluje simultánní komunikaci mezi kontrolérem a vysílačem i přijímačem. V jednoduché komunikaci s jedním je role zdroje a přijímače jasná, přičemž mikrokontroler může být kterýkoliv z nich. V případě mikrokontroleru a dvou zařízení je možné ho použít pouze jako řízení provozu, a tedy generace signálu BCLK a WS pro obě zařízení zároveň, přičemž data jdou přímo mezi zařízeními – není do nich „vidět“. Druhá možnost konfigurace je datový tok i sběrnici rozdělit – mikrokontroler postaví mezi oba přenosy a generovat všechny signály pro obě zařízení separátně a chovat se jako prostředník, který má k datům přístup – předává je, ale

současně i čte. Součtem dvou I2S sběrnic bude tedy 6 separátních pinů, přičemž každý I2S kanál obstarává vlastní sběrnici. Tato možnost zmenšuje nároky na přesnost generace hodin a WS, kanály tedy nemusí nutně být vzájemně synchronní a pomocí virtuálního programového bufferu uvnitř kontroléru dochází k bezproblémovému předávání dat mezi kanály.

3.2.3 Serial Peripheral Interface SPI

Jedná o základní typ sériové komunikace, která je použita pro většinu mikrořadičů, které se starají o datový tok na krátkou vzdálenost. Nejen paměti, ale i veškeré senzory, převodníky a jiné lokální zdroje signálů komunikují přes tuto čtyř vodičovou sběrnici. Rozvod hodin z Master obvodu – SCK je pro všechny připojené Slave obvody společný, stejně tak jako datový vodič na vstup dat SDI a výstup dat SDO. Čtvrtý vodič výběrového signálu CS je pro každý připojený obvod zvlášť a aktivuje ten, se kterým je třeba komunikovat, zbylé obvody jsou neaktivní. Díky dvěma datovým vodičům je možná full-duplex komunikace, což znatelně urychluje provoz. Moderní úprava sběrnice dovoluje přepínání rolí Master a Slave při běhu. Datové piny se tak mění na MISO (Master in Slave out) a MOSI (Master out Slave in). Výběrový signál CS je změněn na SS (Slave select). V případě změny Slave na Master je třeba přehodit smysl vodičů z Master in na Slave out a Master out na Slave in, obdobně pak u Master na Slave. Důležité je dodržet v jednom čase pouze jeden obvod Master, což je úkol správně nastaveného softwaru. V případě použití komunikace SPI pro modul s SD kartou není třeba změny Master obvodu. Inicializaci komunikace zahajuje mikrokontroler a modul s kartou plní požadavek čtení či zápisu.

ESP32 disponuje dvěma nezávislými uživatelsky přístupnými SPI sběrnicemi, a to VSPI a HSPI, zbylé dvě SPI0 a SPI1 jsou použity interně pro vlastní řízení a neměly by být uživatelsky použity. Každá sběrnice může obsloužit až 3 Slave periferie. Maximální frekvence je 40Mhz za předpokladu, že připojený Slave obvod pouze posílá data (half-duplex), je potlačen DMA přístup a k přenosu signálu se používá GPIO matice. V případě použití pinů s multiplexory místo matice je možné dosáhnout až 80MHz.

4 Zpracování elektronické části

4.1 Mikrofon

Při snaze dosáhnout co nejlepšího akustického záznamu, je třeba nahlížet hned na několik parametrů mikrofону. Primární důraz je kladen na frekvenční odezvu, tedy bylo třeba volit takový mikrofon, který je relativně schopný zaznamenat nízké frekvence. Konkrétně srdce je z principu činnosti kontrakcí zdrojem velmi nízkých frekvencí. I přes to, že je málo zdrojů uvádějící kompletní frekvenční spektrum, lze zhruba říct, že se jedná již o ultrazvuk od 1 Hz až po zvuk do 200 Hz. Zde však velmi záleží na způsobu snímání (viz obrázek 22). V případě anomálií srdeční činnosti může být spektrum širší.

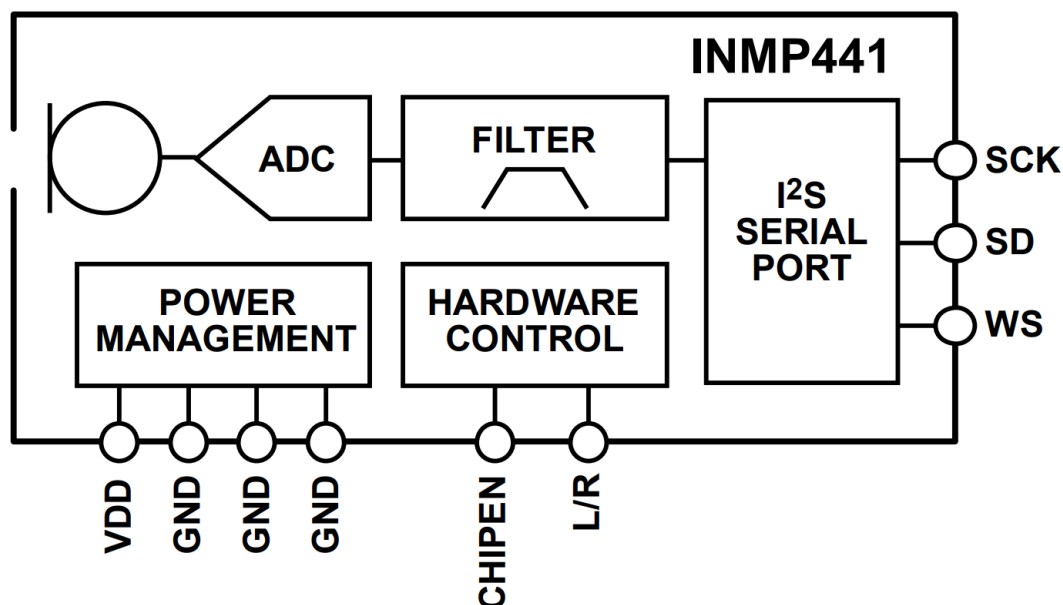
Problém s určením přesného spektra je jednak z důvodu přechodu prostředí, tělo – vzduch, jednak princip snímání zvuku.

V případě, že zvuk snímáme klasickým mikrofónem, musí akustická vibrace vyvolaná srdcem projít skrz různé vrstvy lidské tkáně, přičemž každá z nich utlumí jinou část spektra. Dále se musí vyzářit pokožkou, která oproti tuhé membráně reproduktoru způsobí další nerovnoměrný útlum v signálu. Poté lze zvuk poměrně bez problému zachytit pomocí nízkofrekvenčního mikrofónu. Pokud však zvuk snímáme pomocí kontaktu s tělem, lze použít například piezoelektrický měnič, který buď přímo, nebo přes membránu přiloženou na tělo, prohybem generuje elektrický signál. Obecně lze problém shrnout jako problém impedančního přizpůsobení mezi vysílačem a přijímačem.

4.1.1 Volba mikrofónu

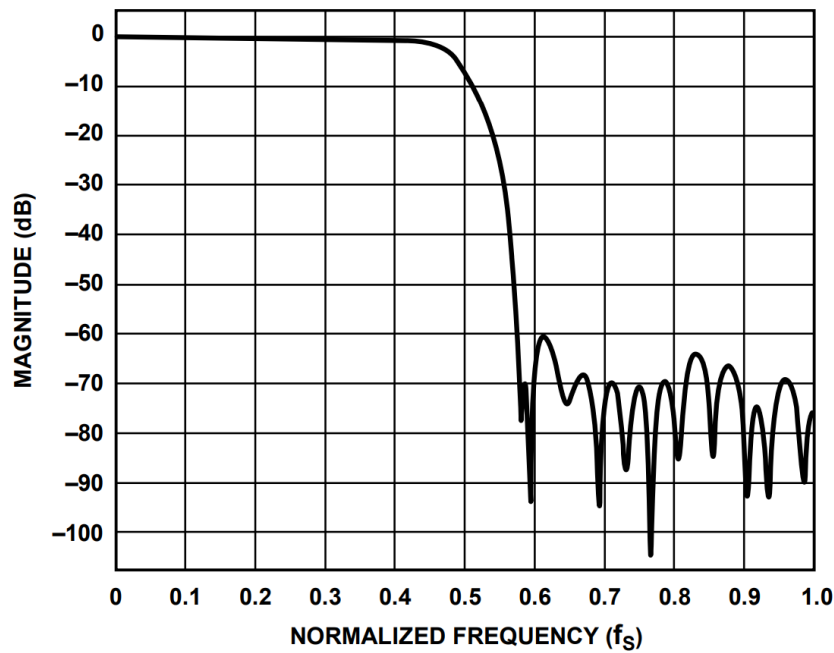
Právě díky těmto obtížím je třeba udělat kompromis a volit takový mikrofon, který bude dostačovat. Mnoho dostupných modulů s mikrofóny nabízí dolní frekvencí okolo 100 Hz, nejnižší ji mají na 50 Hz.

Další volbou je typ měniče, zmíněný piezoelektrický, kondenzátorový nebo klasický elektrodynamický. Obdobou kondenzátorového je MEMS mikrofon, tedy micro-elektromechanická struktura. Oproti kondenzátorovému mikrofónu je hlavní rozdíl v tom, že MEMS je přímo vytvořená na substrátu podobně jako tranzistor, vrstvu po vrstvě. Uvnitř kapsle o velikosti 5x7mm je celý elektronický obvod, který signál převede na digitální vzorky, odfiltruje a připraví na přenos přes I2S kanál.



Obrázek 11 Blokové schéma MEMS kapsle

Z modulu tedy rovnou vychází digitální data, která mají menší až žádný problém s rušením přenosu, což je nesporná výhoda oproti analogovému protějšku. Nevýhodou však mohou být permanentně dané parametry výstupních dat. Interní zesilovač často nemá možnost nastavení zisku, je tak pevně daný a případné ovlivnění hlasitosti je pomocí kódu. Stejně tak úroveň šumu, nebo frekvenční rozsah je pevně dán. Je však nutné zmínit, že jelikož uvnitř pouzdra je použitý sigma-delta AD převodník, jsou parametry závislé na vzorkovací frekvenci. Pomocí převzorkování, při AD převodu, se většina šumového výkonu posune daleko nad slyšitelnou mez. Také frekvenční filtr je závislý na frekvenci vzorkování, od které se odvíjí jeho propustné pásmo.



Obrázek 12 Frekvenční spektrum vnitřního filtru definované vzorkovací frekvencí

Lze tedy říci, že kvalita paramterů roste s rostoucí vzorkovací frekvencí. Z teorie vzorkování by vyplývalo, že volba f_s by měla být lehce nad dvounásobkem naší maxiální zazamenávané frekvence. Ale z důvodu použití výstupního I2S zesilovače je třeba volit 48 kHz jelikož tato frekvence zajistí nejčistší zvuk při jeho reprodukci v reálném čase.

Díky tomu, že mikrofon funguje jako celek a není třeba řešit externí převodníky a filtry, je vhodný pro integraci do přenosných zařízení, ve kterých je každý cm^2 cenný. Proto si našel cestu do dnešních chytrých telefonů, ve kterých přes svou velikost velmi kvalitně zachycuje zvuk.



Obrázek 13 modul MEMS mikrofonu INMP441

Právě to byl hlavní důvod výběru pro snímání srdečního tepu. Je to levný, hotový kompaktní celek, který není třeba dále ladit. Z důvodu neideálního frekvenčního pásma, byla v úvahu i jiná, více analogová varianta.

Elektretový kondenzátorový mikrofon, by mohl nabídnout podobné parametry, jeho použití by však vyžadovalo dodatečný obvod. Tento mikrofon v sériové kombinaci s rezistorem vytváří napěťový dělič. Jeho analogový výstup by bylo třeba nejprve zesílit, jelikož napěťový rozkmit výstupu je při napájení 3.3 V zhruba 30 mV. ESP32 má v použité variantě pouze 12bitový vstupní AD převodník, což je pro záznam zvuku dostatečná, avšak ne ideální hodnota. Proto by poté bylo třeba pomocí externího AD převodníku analogový signál kvantovat a následně zpracovat v MCU jako digitální data. Problémem by však mohl být šum způsobený ne příliš kvalitním 3,3 V napěťovým pinem MCU. Jakékoliv zvlnění by se hned projevilo na výstupu díky funkci napěťového děliče a rušilo by tak snímání zvuk. Bylo by tedy nutné mikrofon napájet přes low-drop stabilizátor. A náročnost celého obvodu by se tak značně zvětšila.

Použití MEMS mikrofonu tyto problémy eliminuje – má již digitální výstup a také je relativně odolný na horší stabilitu napájení, kterou 3,3V pin na ESP32 může trpět. Naproti tomu, při návrhu obvodu pro kondenzátorový mikrofon by bylo možné vyladit parametry zesilovače a převodníku tak, aby co nejlépe sloužil při snímání zvuku srdce. Otázkou je, zda by se tato práce vyplatila, jelikož testovaný mikrofon ld-mc-6035w má frekvenční odezvu od 50 Hz, což je podobné MEMS mikrofonu. Pokud by byl k dispozici mikrofon s výrazně lepšími parametry dolní frekvence, tak by stálo za snahu jej zaměnit za zmíněnou MEMS variantu.

4.1.2 Konfigurace periferie pro použití MEMS mikrofonu

```
void i2s_install() {
    const i2s_config_t i2s_config = {
        .mode = (i2s_mode_t)(I2S_MODE_MASTER | I2S_MODE_RX | I2S_MODE_TX),
        .sample_rate = I2S_SAMPLE_RATE,
        .bits_per_sample = I2S_BITS_PER_SAMPLE_16BIT,
        .channel_format = I2S_CHANNEL_FMT_ONLY_LEFT,
        .communication_format = I2S_COMM_FORMAT_STAND_I2S,
        .intr_alloc_flags = 0,
        .dma_buf_count = 3,
        .dma_buf_len = bufferLen,
    };
    i2s_driver_install(I2S_NUM1, &i2s_config, 0, NULL);
    i2s_driver_install(I2S_NUM2, &i2s_config, 0, NULL);
}
```

Kódová část 1 Inicializace I2S komunikace pro přenos dat mezi mikrofonem i zesilovačem

I2s periferii na ESP32 je třeba nakonfigurovat tak, aby pracoval jeden z periferních obvodů pro mikrofon a přijímal data, a druhý jako výstup pro DAC zesilovač. MCU je v obou případech jako master, který kontroluje externí zařízení. Dále je jeden kanál využit pro příjem, druhý pro vysílání, proto je v konfiguraci *mode* nastavena hodnota RX i TX. Sample rate je nastaven dle doporučených parametrů z katalogu. Při volbě bitové hloubky by obě zařízení měly zvládat 24bit vzorky, avšak se při testech nepodařilo tento režim zprovoznit. Proto je nastaveno 16 bitů na vzorek, čímž se lehce ulehčí výpočetní náročnosti zpracování signálu. Komunikace probíhá přes sériovou synchronní komunikaci, ale díky využití DMA přístupu je nutno nakonfigurovat parametry těchto bufferů, které se starají o ukládání vzorků z mikrofonu.

Nastavení parametrů DMA je jednak otázkou doby zpracování signálu, jednak rychlostí příchozích a odchozích dat. Obecně lze říct, že čím větší buffer, tím menší zátěž procesoru, jelikož bude k datům přistupovat a zpracovávat je po delších intervalech. Nevýhodou je však latence, která se s rostoucím bufferem zvětšuje. Je to tedy volba mezi malým zpožděním vstup-výstup a zátěží procesoru.

Pokud budeme uvažovat, že používáme f_s 48kHz a máme 16bit na vzorek pouze na jednom kanálu, lze spočítat datový tok jako:

$$\begin{aligned}
 Bps &= f_s \cdot \frac{\text{velikost vzorku}}{8bit} \cdot \text{počet kanálů} \\
 &= 48000 \text{ Hz} \cdot \frac{16bit}{8bit} * 1 \text{ ch} = 96 \text{ kBps}
 \end{aligned}
 \tag{4.1}$$

S touto rychlostí musíme počítat, pokud známe dobu zpracování dat v MCU. Pokud by zpracování jednoho bufferu trvalo 1 ms. Potom musíme zaručit, aby DMA buffer měl takovou velikost, aby se minimálně rovnal počtu zpracovávaných vzorků – aby překlenul dobu mezi čtením dat. Za 1 ms vznikne z předešlého výpočtu 96 bajtů dat neboli 48 16bit vzorků. Pokud chceme, aby měl procesor stálou zásobu bufferů plných dat, které zpracovává, poté je volba například 64 vzorků na jeden buffer. V případě digitálního processingu signálu je však doba zpracování závislá na velikosti bufferu. Pokud tedy čteme místo 64 vzorků 128, bude trvat zpracování přibližně dvakrát déle.

Jelikož jeden DMA buffer může být přístupný výhradně pouze na čtení nebo pouze na zápis v jednu chvíli (v praxi to znamená, dokud se nezapíše nebo nepřečte požadované množství dat), je nutné zvolit větší počet těchto bufferů. Zatímco se jeden buffer načítá pro zpracování DSP, druhý buffer se mezitím plní dalšími daty z I²S sběrnice.

Hlavním úkolem je tedy zajistit, aby neustálý přísun dat z mikrofonu proudil do celkového DMA bufferu tak velkého, aby se při případném delším zpracování nezaplnil a nebylo možné dále ukládat nová data. Nebo naopak, aby při rychlém zpracování nedošlo k nespojitostech ve výstupních datech, z důvodu vyprázdnění všech bufferů a plnění jednoho, ze kterého v ten moment nelze číst. Jelikož však MCU pracuje se tokem živě generovaných zvukových vzorků, je tento problém usnadněn tím, že tok dat dovnitř se rovná toku dat ven. Interval čtení tedy závisí pouze na velikosti požadovaných bufferů pro zpracování.

Pokud bude zvolena velikost dat pro jeden cyklus filtrování 128 vzorků, potom načtení těchto dat do DMA bufferu bude trvat

$$T_r = \frac{\text{velikost bufferu}}{\text{vzorky za sekundu}} = \frac{128 \text{ [vz]}}{48000 \left[\frac{\text{vz}}{\text{s}}\right]} = 2,6 \text{ [ms]}
 \tag{4.2}$$

T_r zde označuje periodu plnění jednoho celého bufferu, k čemuž dojde jednou za 2,6 ms a následně se zpřístupnění ke čtení. Pokud bude doba potřebná k zpracování 128 vzorků dat např 2 ms, poté je teoreticky po dobu 0,6 ms MCU nečinný mezi cykly. Lze

říct, že čeká na data, jelikož je schopný je zpracovat rychleji, než přichází. Vidíme tedy jasnou souvislost mezi intervalem plnění DMA bufferů a rychlostí zpracování stejného množství vzorků. Vyplývá tak minimální podmínka nepřerušného toku dat.

$$T_r \geq T_{pr} \quad (4.3)$$

Kde T_{pr} je doba zpracování potřebná k provedení všech výpočetních operací jednoho plného bufferu.

Pokud by však doba zpracování přesahovala interval plnění bufferů, pak by docházelo buď k neustálému zvětšování latence in-out a zásoby dat před zpracováním, dokud by se všechny DMA buffery nezaplnily a poté by vzorky nebylo kam ukládat. Nebo v případě cyklického běhu exekuce kódu by byl výstup dat přerušovaný s intervalem T_{pr} po dobu $T_{pr} - T_r$. Pokud by tedy zpracování zabralo 3 ms, tak by právě každé 3 ms byla ve výstupu dat mezera 0,4ms.

Pro některé aplikace je možné takové přerušení toku a případné zobrazování poslední hodnoty, připustit, pokud se jedná například o zobrazování hodnoty čidla, která se výpočetně zpracovává a není nutné znát její aktuální hodnotu s velkou frekvencí. Pokud však pracujeme s tzv. time-critical kódem, při kterém jakékoliv přerušení toku dat nemůžeme dovolit, proto takové nastavení DMA kanálu nelze připustit. Jelikož pracujeme s konstantním tokem zvukových dat, musíme použít právě time-critical přístup. Jakékoliv nedoručení, byť jediného vzorku, může způsobit typické slyšitelné lupnutí. Stejně tak, pokud by došlo k nenávaznosti dvou bufferů, tak by při přechodu z posledního vzorku prvního bufferu a prvním vzorkem bufferu druhého opět došlo k lupnutí. Důvodem je vznik harmonických složek při následné rekonstrukci signálu do analogové formy. Náhlá změna navazujícího vzorku vytvoří široké spektrum harmonických složek s pomalu klesající intenzitou, které se zvukově projeví napříč celým spektrem.

Aby ke zmíněnému přerušení toku dat nedocházelo, volíme tedy dostatečný počet DMA bufferů, které dle svého počtu mohou sloužit jako vyrovnání nepředpovídaného zpoždění průchodu dat DSP. Pro použití v cyklickém chodu kódu však není třeba vytvářet nadbytečné množství, jelikož je přístup k datům pevně dán kódem, který z nich čte s danou periodou. Teoreticky tedy stačí dva DMA buffery, tak aby mohla být nová data zároveň zapisována a stará zároveň čtena. Pro dodatečnou rezervu je volba 3 bufferů ideální. Jeden by měl být volný, a tedy připraven na použití při nedodržení časování z důvodu vnitřního zpoždění.

Alokování DMA bufferů probíhá ve stejné interní SRAM paměti, ve které jsou uloženy proměnné kódu. Lze tedy použít pouze tak velký buffer, který ponechá dostatek datového místa pro běh kódu a jeho některých proměnných. V opačném případě dojde k chybovému

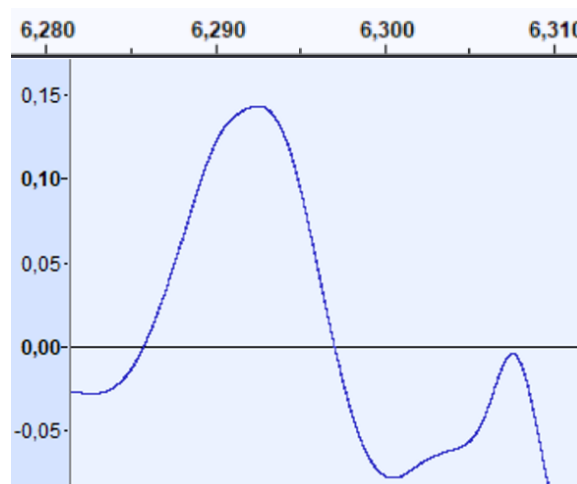
hlášení a opakujícímu se restartování programu. Celková velikost paměti rezervované pro DMA buffery s použitím jednoho zvukového kanálu bude tedy:

$$\begin{aligned} total_dma_memory &= dma_buf_count \cdot bufferLen \cdot \frac{bits_per_sample}{bits_per_byte} & (4.4) \\ &= 3 \cdot 256 \cdot \frac{16}{8} = 1536 \text{ [Bytes]} \end{aligned}$$

To v našem případě zabírá z 520kB místa v SRAM zhruba 1,5 kB, což je nepodstatné zabrání místa. Pokud by však kód byl velmi náročný, co se paměti týče, a měli bychom větší množství dat nutných pro uložení do více DMA bufferů, museli bychom to vzít v potaz. V každém případě je nutné zvolené parametry otestovat, i pokud se výpočetně zadají v pořádku.

4.2 Efektivní bitová hloubka

Při navrhování celého zařízení bylo z důvodu použití MEMS mikrofonu se zabudovaným zesilovačem a AD převodníkem zřejmé, že se nepovede využít celkový dynamický rozsah mikrofonu. Vybuzení takového mikrofonu (použitý inmp441 má typický rozsah 87 dBA) je problémové i s použitím trychtýřovitého nástavce na zlepšení izolace snímání srdce. Při záznamu by byl signál slabý a bez dodatečného zesílení by nebylo možné jej akusticky přehrávat.



Obrázek 14 Průběh signálu bez zesílení v místě největšího výkmitu, s časovou x osou [s] a amplitudou v ose y

Pro test byl vybrán z 10 sekund záznamu srdeční činnosti výkmit s největší maximální hodnotou a to 0,14289 z 1, což odpovídá 14,3 %. Maximální dekadická hodnota znaménkové

16bit hodnoty je 32767 a pokud uvažujeme pouze kladné hodnoty znamená to, že dosažený výkmit je:

$$\mathbf{Amplituda = 32767 \cdot 0,143 \doteq 4\,685 \text{ [-]}} \quad (4.6)$$

To znamená že potřebujeme zjistit takový minimální počet bitů, který definuje rozsah s touto hodnotou. Pokud začneme mocninami dvou, pak je zřejmá hodnota 2^{12} s hodnotou 4096. Binárně 1111 1111 1111 je v dekadické hodnotě také 4095. Naše hodnota je však o něco větší, proto je třeba alespoň 13 bitů, na vyjádření dekadické hodnoty 4685 a to konkrétně 1 0010 0100 1101. Pokud však chceme rozlišovat kladné a záporné hodnoty, musíme přidat ještě jeden bit. Celková využitá bitová hloubka je tedy 14 bitů z původních 16. Je však nutné zmínit, že ani 14 bitů není maximálně využito, jelikož maximální kladná hodnota 14bitového čísla je dekadicky 2krát 4096, tedy 8191. Pokud bychom chtěli takto zpřesnit využitou bitovou hloubku, dalo by se říct, že využíváme 13 a ½ bitu. Definování poměru úrovně signálů vůči kvantizačnímu šumu pro původních 16 bitů lze získat pomocí rovnice:

$$\mathbf{SQNR_{dB} = 20 \log_{10} \left(2^b \sqrt{1,5} \right) = 20 \log_{10} \left(2^{16} \sqrt{1,5} \right) = 98 \text{ dB}} \quad (4.7)$$

Pro efektivní počet bitů je pak snižená hodnota o 11 dB.

$$\mathbf{SQNR_{dB} = 20 \log_{10} \left(2^b \sqrt{1,5} \right) = 20 \log_{10} \left(2^{13,5} \sqrt{1,5} \right) = 83 \text{ dB}} \quad (4.8)$$

Maximální možný dynamický rozsah, který je s efektivní bitovou dosažitelný tedy klesá na 83 dB. Skutečný dynamický rozsah deklarovaný výrobcem s poměrem signál-šum je 61dBA.

Jednotka dBA neboli A-weighted je odlišná v nelineárnosti průběhu. Její hodnota se vztahuje k vlastnostem lidského ucha, které je také nelineární napříč frekvenčním spektrem, co se vnímání intenzity týče. Proto zde vystupuje hodnota, která je lepší v porovnávání vnímané intenzity hlasitosti, místo absolutní strojově dané.

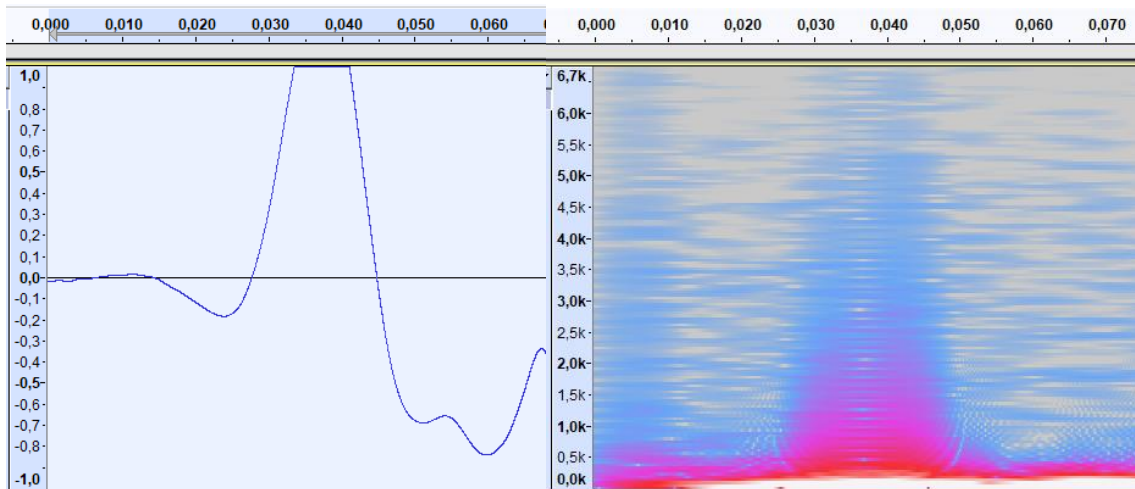
4.3 Problém hlasitosti signálu

Z důvodu malé hlasitosti nahraného záznamu je třeba jej zesílit. Digitální výstup mikrofonu nám nedává jinou možnost než zesílení zpracovat kódově. Každý vzorek je tedy ve for cyklu násoben konkrétní celočíselnou hodnotou. Pokud by však nebyl ošetřený maximální rozkmit mohlo by dojít k přetečení int16_t hodnoty a špička amplitudy nad maximální hodnotu 32767 by nabyla zápornou hodnotu.

```
void volume_adjust(int16_t* buffer, size_t bufferSampleCount,
                  int gain, const int rangeOfSample) {
    int mem;
    for (int i = 0; i < bufferSampleCount; i++) {
        mem = buffer[i] * gain;
        if ((mem < rangeOfSample) && (mem > -rangeOfSample)) {
            buffer[i] *= gain;
        } else if (mem > rangeOfSample) {
            buffer[i] = rangeOfSample;
        } else if (mem < -rangeOfSample) {
            buffer[i] = -rangeOfSample;
        }
    }
}
```

Kódová část 2 Funkce zesílení zaznamenaných vzorků s omezením na maximální hodnotě

V kódu dojde prvně k výpočtu nové hodnoty, ale pouze pokud je menší než maximální hodnota 32767, tak je uložena do původní. V opačném případě, je původní hodnota nahrazena zmíněnou maximální hodnotou. Tím se zajistí, že výkmit bude pouze zploštěn ale zachová si polaritu. Tento způsob zkreslení signálu označujeme jako clipping a ve většině případů je nechtěný. Nový tvar signálu, který obsahuje několik konstantních hodnot vytváří harmonické frekvence, které v superpozici s původním výkmitem dosáhnout právě tohoto zploštění špičky.



Obrázek 15 Signál s přebuzením a jeho efekt ve spektrogramu s rostoucí intenzitou od modré přes červenou k bílé, osa x nahoře označuje čas (s), osa y nejprve amplitudu a frekvenci (Hz)

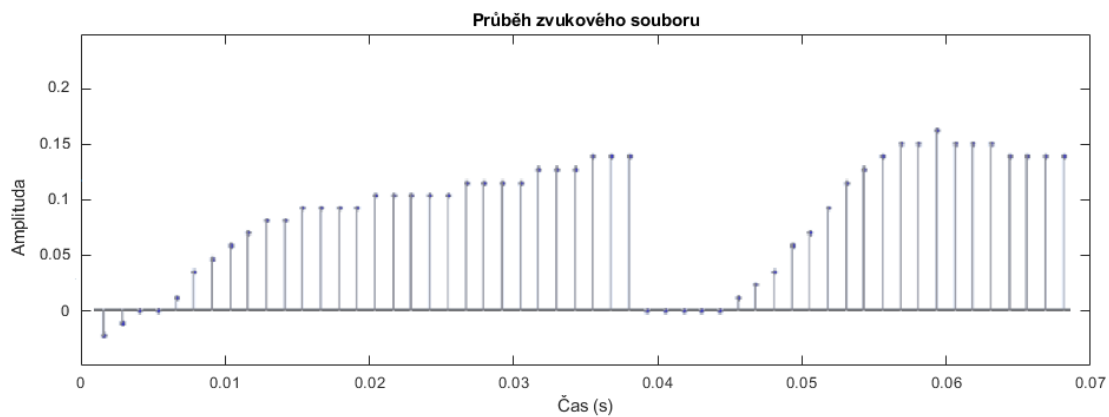
Ačkoliv je toto zkreslení nežádoucí, je stále přijatelnější než obrácení polarity přetečením, které by mělo silnější harmonické složky. Pokud bychom chtěli řešit problém s clippingem a i přesto mít velké zesílení byl by potřeba do kódu implementovat zvukový kompresor, který má nelineární zesílení v závislosti na aktuálním výkmitu. Je tedy možné ho nastavit tak, aby v případě možného přebuzení nedošlo ke zpolštění. Použití kompresoru tedy zamezuje přebuzení, avšak nelineární zesílení také vytváří harmonické složky, ale ne v takové míře a lze je stále korigovat vnitřním nastavením.

4.4 Digitální filtry

Získaný signál máme v digitální podobě zesílený a uložený v bufferu. Nyní potřebujeme použít adekvátní filtrování pro získání potřebného frekvenčního spektra.

4.4.1 Zpoždění způsobené výpočtem při filtrování a zesílení

První variantou bylo filtrovat signál pomocí FIR filtrů. Jejich implementace v různých DSP naznačuje, že by to mohla být správná cesta, jak dosáhnout slibného výsledku. Filtrování však uvažuje se vstupními vzorky jako float hodnoty, je tedy nutné `int16_t` přetypovat na float bez zkreslení hodnoty. Následně je třeba opět převést stejnými pravidly zpět. Náročnost výpočtu tedy značně roste jen z důvodu přetypování. Následná samotná filtrace probíhá tak, že se každý vzorek několikrát násobí všemi koeficienty filtru v jednom okně, které se posouvá pro každý vzorek.



Obrázek 16 Jednotlivé vzorky signálu s patrným nulovými hodnotami v místě, kde na sebe buffery navazují

Výsledkem filtrování byl signál, který však vykazoval problém kontinuitou vzorků. Přesně po jedné délce bufferu, který se zpracovával se ve výsledném zvukovém souboru objevily mezery s několika nulovými vzorky. Zvuk byl tak velmi trhaný a proto nepoužitelný.

```
static float filterCoefficients[21] = { 0.000642, 0.002213, 0.004967, 0.009090, 0.014552,
0.021060, 0.028048, 0.034756, 0.040344, 0.044052, 0.045351, 0.044052, 0.040344, 0.034756,
0.028048, 0.021060, 0.014552, 0.009090, 0.004967, 0.002213,
0.000642};
float fCommunicationData[numCommunicationData];
float filteredCommunicationData[numCommunicationData];
// zero padding for FIR window
for (int i = 0; i < numCommunicationData; i++) {
    fCommunicationData[i] = 0.0;
}
// Convert communicationData from int16_t to float
for (int i = 0; i < numCommunicationData; i++) {
    fCommunicationData[i] = static_cast<float>((int16_t)communicationData[i] / 32767.0f);
}
// Apply the FIR filter
for (int i = 0; i < numCommunicationData; i++) {
    filteredCommunicationData[i] = 0.0;
    for (int k = 0; k < (sizeof(filterCoefficients) / sizeof(filterCoefficients[0])); k++) {
        if (i >= k) {
            filteredCommunicationData[i] += fCommunicationData[i - k] * filterCoefficients[k];
        }
    }
}
// Convert the filtered sample to a signed 16-bit integer
for (int i = 0; i < numCommunicationData; i++) {
    communicationData[i] = static_cast<int16_t>(filteredCommunicationData[i] * 32767.0f);
}
```

Kódová část 3 Původní návrh FIR filtru, který však vykazuje nekontinuální výstup mezi buffery

Při měření zpoždění průchodu dat filtrem vyšly všechny velikosti od 16 do 512 vzorků v limitu maximálního časového zpoždění (viz tabulka), přesto se ve výsledném souboru objevují artefakty indikující nekonzistentní dodávku dat, případně chybu s nakládáním s nimi. Bylo by potřeba hlouběji prozkoumat fungování výpočtu a důkladně zkoumat tok jednotlivých zvukových vzorků. Dále je nutné zmínit, že pro konzistentnost měření s reálným použitím byl při výpočtu aktivní wifi server, který díky dvoujádrové architektuře běží na svém jádře a druhé je plně přístupné pro zbytek kódu.

Tab 1 Změřené zpoždění výpočtu zpracováním signálu

	Perioda čtení @48kHz	AVRG[10]	FIR [21]
Vzorky	T_r [μ s]	T_{pr} [μ s]	T_{pr} [μ s]
1024	21333	653	-
512	10667	347	1437
256	5333	198	732
128	2667	115	382
64	1333	83	208
32	667	59	114
16	333	48	72

Stejnému měření byl podroben i filtr průměrování vzorků, a to pomocí jednoduchého cyklu, který do funkce s filtry pošle plný buffer o délce 16,32,64,128,256,514 a 1024 vzorků. Každý výpočet celého bufferu provede 5krát a postoupí k většímu bufferu. Pro výpočet délky výpočtu u FIR filtru při zvolení 1024 vzorků na buffer došlo k pádu MCU a program nebyl schopen výpočet dokončit. Funkce ReadProcessWrite je základem celého zpracování signálu. V tomto měřicím programu má však pouze jeden filtr, a to výpočetně nejnáročnější průměrování 10 vzorků a dvě separátní zesílení zachyceného signálu pomocí jednoduchého násobení. Jedno zajišťuje dostatečnou hlasitost přehrávání, druhé pak další zesílení pro uložení v záznamu. Druhá varianta používá FIR filtr a opět dvě zesílení hlasitosti. V původní funkci je navíc možnost zapisování bufferů do souboru na SD kartě, což se děje pouze při nahrávání, ale zde není zahrnuta.

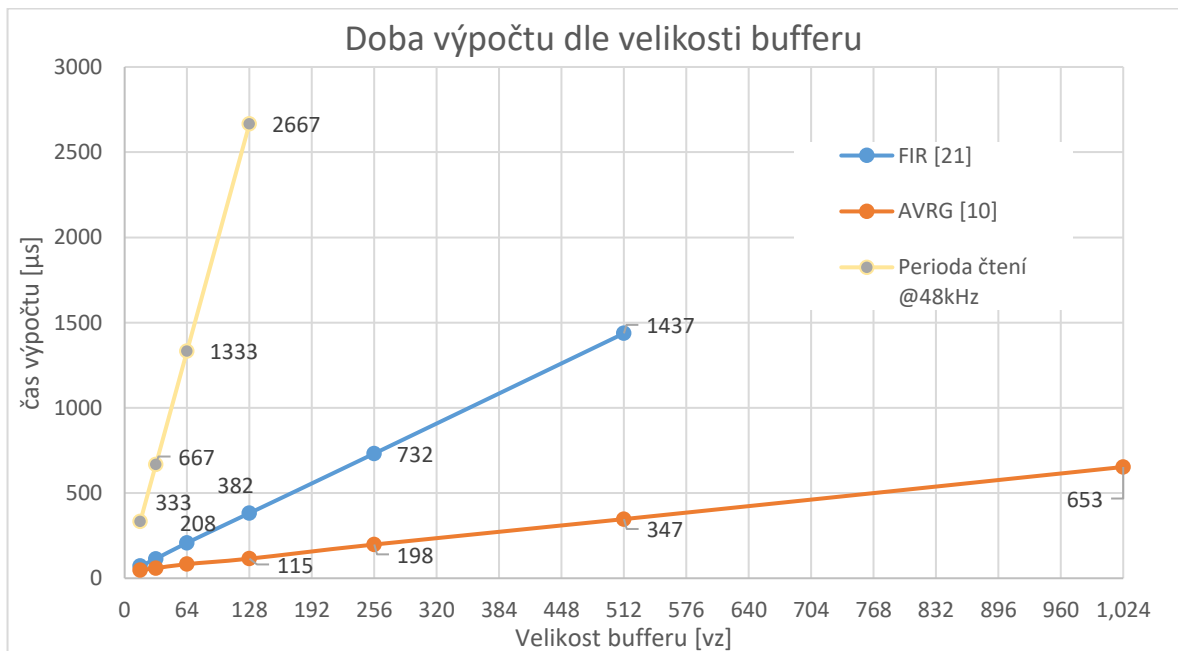
```

static int bufferLen = 0;
void loop() {
  if (Serial.read() == 'c') {
    bufferLen = 16;
    for (int j = 0; j < 6; j++) {
      int16_t data[bufferLen];
      for (int i = 0; i < bufferLen; i++) {
        data[i] = (i - 50) * i;
      }
      for (int i = 0; i < 5; i++) {
        ReadProcessWrite((int16_t*)data);
      }
      bufferLen *= 2;
    }
  }
}

```

Kódová část 4 cyklus měření doby zpracování jednoho zvukového bufferu pro 16,32,64,128,256,512 a 1024 vzorků

Z měření jasně vyplývá, že délka výpočtu není limitací ani při vzorkování 48kHz. Při zobrazení dat do grafu je vidět lineární závislost již od 16 vzorků na buffer a značně větší náročnost u FIR výpočtu. Žlutá čára naznačuje maximální teoretické zpoždění, které je z důvodu již uvedené podmínky $T_r \geq T_{pr}$ s dostatečnou rezervou, takže zpracování signálu by mohlo být výpočtově náročnější a stále by zpracování signálu trvalo menší dobu, než je doba plnění bufferu.



Obrázek 17 Časové zpoždění výpočtu pro různé velikosti bufferu FIR filtru s 21 koeficienty a průměrový 10 vzorků

4.4.2 Zpracování signálu ve funkci

V následujících řádkách můžeme vidět celou funkci, která běží ve smyčce. V každém průběhu dojde k novému načtení dat z i2s periferie do lokálního int16_t pole data[] s velikostí odpovídající velikosti DMA bufferů. Tento buffer je následně zpracován DSP funkcemi a v případě aktivního výstupu nahrán zpět do i2s periferie, tentokrát ale do výstupního bufferu, který vzorek po vzorku pošle do D/A převodníku se zesilovačem. V případě, že je aktivní nahrávání dojde k uložení bufferu do souboru na SD kartě. Celý proces práce s daty je podmíněný úspěšnou kontrolou načtení dat z periferie. V opačném případě funkce končí a čtení se znovu opakuje novým během funkce.

Zvukový výstup je třeba pouze v případě poslechu pacienta, proto je k jeho aktivování využitý kapacitní snímač dotyku. Mikrokontroler EPS32 má několik pinů disponujících tímto snímáním. Výstupem funkce touchRead() je analogová hodnota v rozmezí od 0 do 100. Ta indikuje, jak moc je pin zatížen a při dobrém odvodu náboje hodnota klesá k 0. Cyklickým opakováním čtení této funkce každých 100 ms. Kontaktní plocha pinu je umístěna na hlavě stetoskopu a pokud je poslech připraven, tak pouhým dotykem aktivujeme zvukový výstup.


```

void ReadProcessWrite() { // i2s read-process-write
    size_t bytesIn;
    size_t bytesOut;
    const int range16 = 32767;
    const int nData = bufferLen;
    int16_t data[nData];
    esp_err_t result = i2s_read(I2S_NUM1, &data,
                               nData * sizeof(data[0]), &bytesIn, portMAX_DELAY);

    if (result != ESP_OK) {
        Serial.printf("Error during reading %d \n", result);
    }
    else {
        if (rec == true) {
            recCycles++;
        }
        switch(filter){
            case 1:
                average_ten((int16_t*)data, nData);
                break;
            case 2:
                break;
        }
        volume_adjust((int16_t*)data, nData, gain_A, range16); // amplify for output
        if (soundOut < 50) { // (soundOut < Threshold)
            i2s_write(I2S_NUM2, &data, bytesIn, &bytesOut, portMAX_DELAY);
        }
        if (gain_A > 0) // amplify for wav file
            volume_adjust((int16_t*)data, nData, (gain_B / gain_A), range16);
        }
        if ((rec == true)) { // write buffer to file
            file.write((uint8_t*)data, nData * sizeof(data[0]));
        }
        if (recCycles == ((waveDataSize / (sizeof(data[0])) / nData))) {
            endRecord();
        }
    }
}
}

```

Kódová část 5 Funkce řešící načtení, zpracování, záznam a výstup zvukových bufferů

FIR tedy musel být nahrazen jiným filtrem, který nevykazoval nespojitosti výstupu. Další možností bylo jednoduché průměrování dvou a více vzorků.

4.4.3 Průměrový filtr

Kvůli výstupu přes I2S zesilovač musela být nastavena relativně vysoká vzorkovací frekvence 48 kHz. Proto je snímané frekvenční spektrum od 0 - 24kHz. To vyplývá z teorie vzorkování, která uvádí, že nejvyšší frekvenční složka snímaného signálu musí být maximálně $\frac{1}{2}$ frekvence vzorkování celého signálu.

$$f_{vz} \geq 2f_{max} \quad (4.9)$$

Případně ekvivalentně v periodě:

$$T_{vz} \leq \frac{T_{min}}{2} \quad (4.10)$$

Kde f_{vz}/T_{vz} označuje frekvenci/periodu vzorkování, f_{max} je nejvyšší frekvenční složka signálu a T_{min} je nejkratší perioda frekvenčně nejvyšší složky signálu.

Pro filtrování zvuku můžeme použít jako filtrovací metodu průměrování vzorků. Tím, že spojíme několik vzorků do jedné hodnoty a následně vydělíme jejich počtem odfiltrujeme nechtěné vysoké frekvence.

Další výhoda a primární účel průměru vzorků je odstranění šumu. Ten má totiž plně náhodný charakter a pokud průměrujeme dostatečné množství vzorků, šum se ve výsledku odečte zhruba na nulu. To však neplatí v nízkofrekvenčním šumu, kde je perioda šumové složky větší než perioda průměru vzorků.

Pro filtrování frekvencí můžeme získat odhad výsledku frekvenční závislosti z počtu vzorků v průměru. Pokud průměrujeme přes celou periodu frekvence, tak její složka kompletně zmizí. To platí i pro její násobky. Průměrování periody, která se opakuje přes 4 vzorky dojde k jejímu odstranění s průměrem 4,8,12... vzorků. Průměr té samé periody 3 vzorky sníží její výkmit zhruba na třetinu původní. Frekvenční spektrum tedy klesá až k frekvenci, která má periodu se stejným počtem vzorků, jaký je průměrován. Vzorkování 48 kHz a průměrování 10 vzorky odstraní frekvenční složku 4800 Hz a všechny její celočíselné násobky. Předchozí frekvence postupně utlumuje. Filtrování 5 vzorky by odstranilo složku 9600 Hz a její násobky a tak dále (viz rovnice 4.21).

Vůči polovině vzorkovací frekvence, se spektrum zrcadlí a útlum filtru zde naopak s rostoucí frekvencí klesá. V této oblasti se však filtr neprovozuje záměrně, případně se používá anti-aliasing filtr, který frekvenční složky nad polovinu vzorkovací frekvence odstraní.

Jako ilustrace kódové implementace filtru je ukázán filtr s velikostí okna 5, čistě pro lepší čitelnost, princip filtru je jinak zachován.

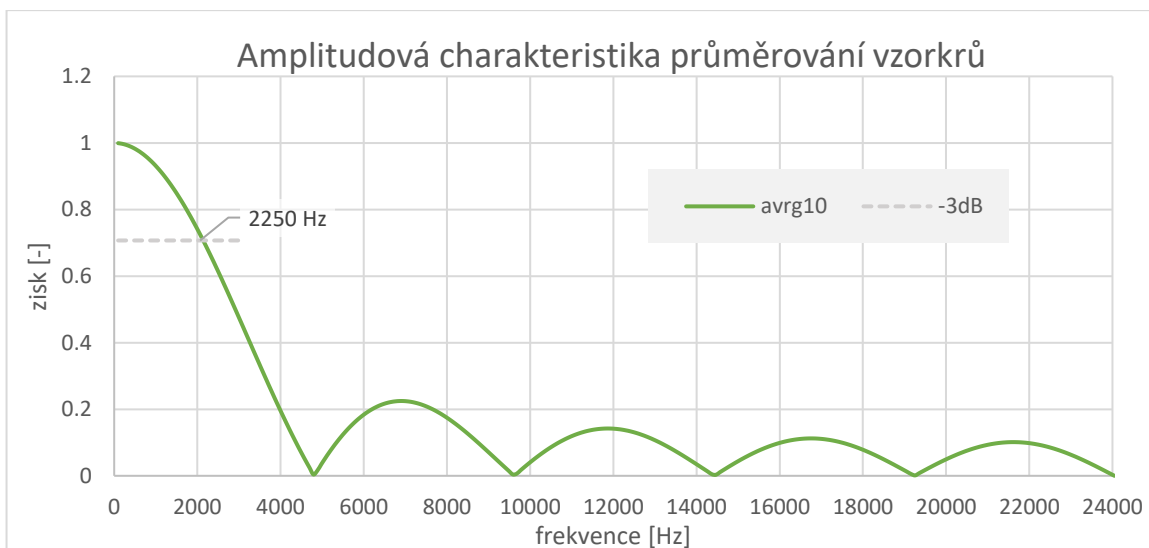
```
void average_five (int16_t buffer[], size_t size){
    int16_t bufferM[size];
    static int16_t mem[4] = {0, 0, 0, 0};
    bufferM[0] = (mem[0] + mem[1] + mem[2] + mem[3] + buffer[0])/5;
    bufferM[1] = (mem[1] + mem[2] + mem[3] + buffer[0] + buffer[1])/5;
    bufferM[2] = (mem[2] + mem[3] + buffer[0] + buffer[1] + buffer[2])/5;
    bufferM[3] = (mem[3] + buffer[0] + buffer[1] + buffer[2] + buffer[3])/5;

    // Update memory buffer
    for (int i = 0; i < 4; i++) {
        mem[i] = buffer[size - 4 + i];
    }
    // Apply the filter to the rest of the data
    for (int i = 4; i < size; i++) {
        bufferM[i] = (buffer[i-4] + buffer[i-3] +
                    buffer[i-2] + buffer[i-1] + buffer[i]) / 5;
    }
    // Copy the result back to the original buffer
    for (int i = 0; i < size; i++) {
        buffer[i] = bufferM[i];
    }
}
```

Kódová část 6 Zvukový filtr řešený průměrováním 5 vzorků s postupným posunem průměrového okna

Filtr 10 vzorků byl při testu aplikován na bílý šum vygenerovaný v programu Audacity. Hodnoty po převedení do excelu a provedení průměrování prošly transpozicí do frekvenčního spektra pomocí Fourierovi transformace. V grafu je vidět nelineárně rostoucí útlum se zvyšující se frekvencí.

Pro zobrazení průběhu útlumu je třeba porovnat rozdíly absolutních amplitud u jednotlivých frekvencí před a po filtrování. Následující graf (17) odhaluje typickou obálkovou strukturu sinc funkce s tendencí klesajících maxim jednotlivých obálek do nekonečna.

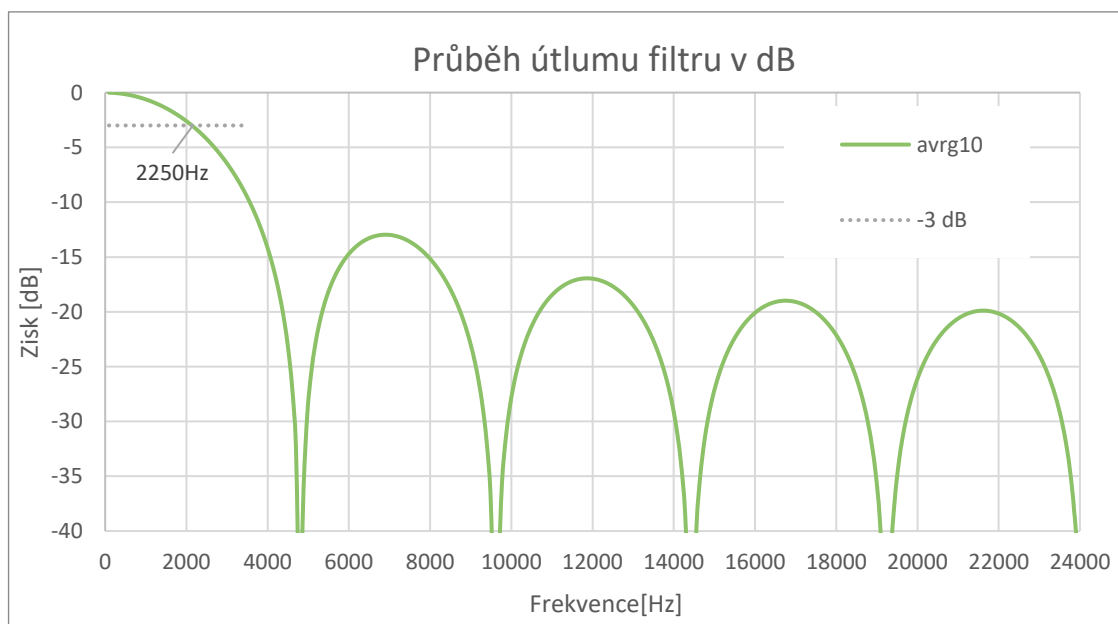


Obrázek 18 Útlumy průměrových filtrů

Z průběhu pak jasně vyplývají i mezní frekvence dolní propusti. Pro filtr s délkou okna 10 vzorků je 2250 Hz. První nulový bod je na 4800 Hz a další na jeho celočíselných násobcích. Pokud bychom chtěli například zajistit nulový bod na 500 Hz, musel by filtr mít délku okna:

$$\text{délka okna filtru} = \frac{f_{vz}}{f_0} = \frac{48000}{500} = 96 \text{ [vz]} \quad (4.31)$$

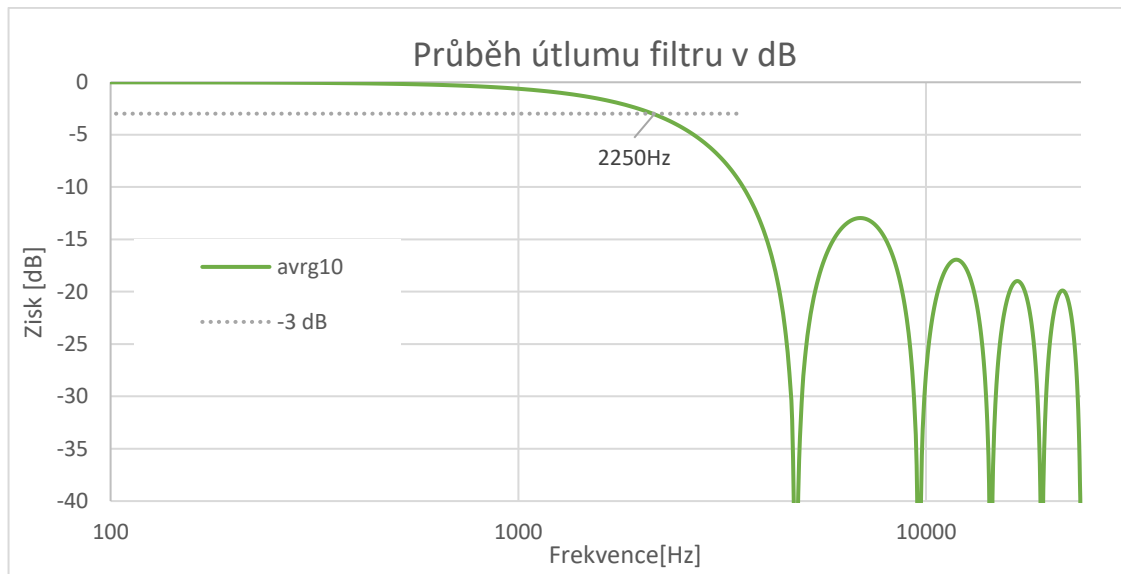
Kde f_{vz} je vzorkovací frekvence a f_0 je první nulový bod filtru, s nekonečným útlumem.



Obrázek 19 Zobrazení útlumu průměrových filtrů v decibelech

Při porovnání útlumu v decibelech se však filtr nejeví jako dokonalý nejen relativně malým útlumem, ale také svou charakteristikou podobnou hřebenovému filtru. V případě snímání srdečního tepu je jeho spektrum omezeno pod 1 kHz, a proto se charakter hřebenu nijak neprojeví. Z důvodu příliš vysoké mezní frekvence bude filtr použit pouze na odstranění ruchu ve frekvencích vysoko nad typickým spektrem zachycovaného signálu. Ty by bez použití filtru byly díky vysoké vzorkovací frekvenci zachyceny na záznam také.

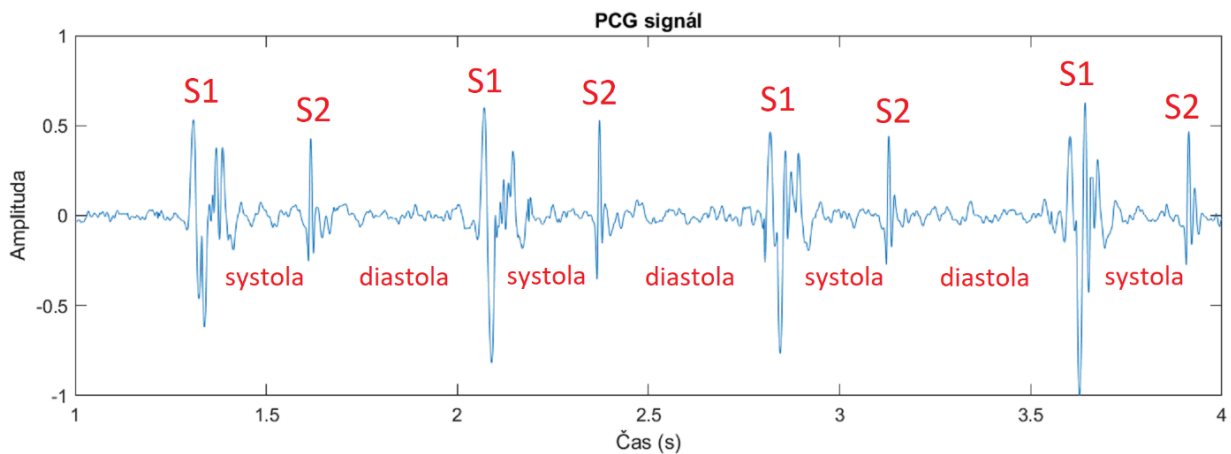
Zde je uvedena útlumová charakteristika filtru v korektní podobě s logaritmickou frekvenční osou.



Obrázek 20 Zobrazení útlumu průměrových filtrů s logaritmickou osou v decibelech

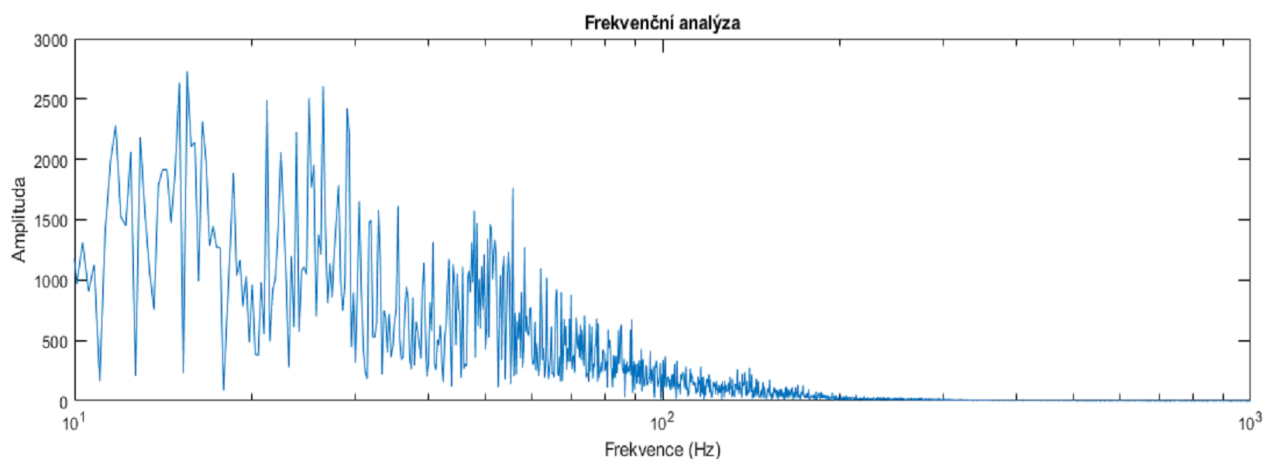
4.5 Průběhy zvukových záznamů

Pro ukázkou průběhu záznamu bylo provedeno měření srdce, bez dalších specifických parametrů.



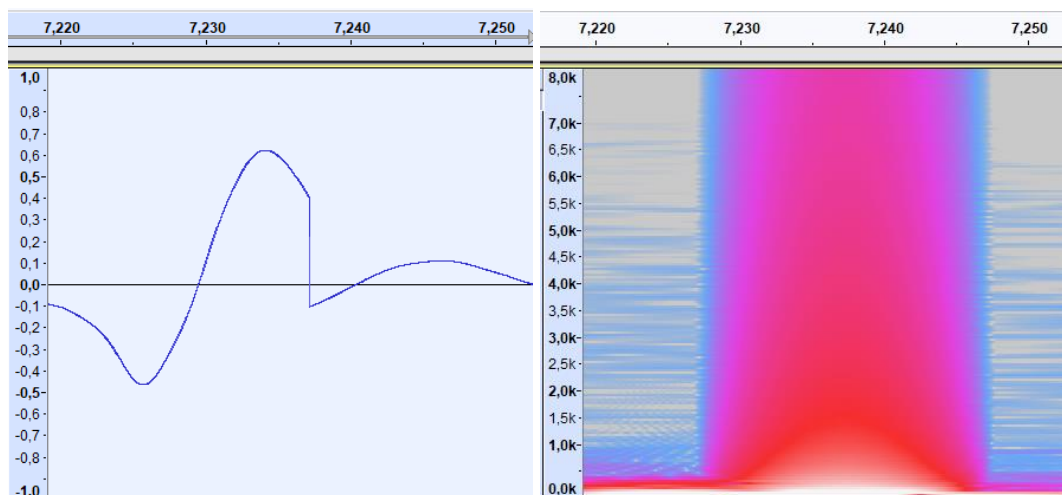
Obrázek 21 Ukázkový zaznamenaný průběh signálu

Při zobrazení průběhu signálu v nahraném souboru lze jasně vidět první a druhou srdeční ozvu. Třetí ozva, která je nepatrná na záznamu není k rozpoznání. První ozvou začíná systola, druhou ozvou pak diastola. Nejintenzivnější signál vzniká při zavření chlopní a akceleraci krve stažením svalu, což je vidět v první srdeční ozvě. Druhá ozva je pak znatelně kratší, jelikož se při ní srdeční sval uvolní a nedochází při ní k tak výrazné změně toku krve jako při silovém stažení.



Obrázek 22 Frekvenční analýza záznamu obr(21) s avrg10 filtrem

Při nahrávání testovacích záznamů výjimečně docházelo k přerušení toku vzorků, čímž se v záznamu ozývalo silné lupání. Tento defekt je pravděpodobně díky nedokonalému kontaktu v signálové nebo napěťové cestě. Na pevně připájený obvod by těmito chybami trpět neměl. Při měření intervalů zpracování také výjimečně docházelo k blíže neidentifikovatelnému „zamrznutí“ při kterém se výpočet opozdil až o 5ms. I to může mít v běžném provozu následky lupnutí v záznamu, proto je třeba pořizovat dostatečně dlouhé nahrávky, aby tyto případné anomálie mohly být vypuštěny a stále zůstal dostatečně dlouhý nerušený záznam pro lékařskou evaluaci.



Obrázek 23 Přerušení toku dat v záznamu + spektrogram s frekvenční osou y [Hz] v namísto amplitudové ve stejné části signálu s časovou osou x [s] nahoře

4.6 Záznam akustického výstupu

Při poslouchání elektronickým fonendoskopem je hlavní výhodou možnost digitálního záznamu. I přesto že poslech provádí výhradně doktor, může být i pro něj výhoda opětovného poslechu srdce, aniž by musel být pacient přítomen. Druhá možnost, inovativní telemedicíny by mohla spočívat v samostatnosti pacienta. Buď v případě, že by měl pocit obtíží, nebo pravidelně jako prevence by mohl sám vytvářet tento záznam a poté ho online odeslat svému lékaři.

4.6.1 Zvukový soubor WAVE

Existuje několik teoretických možností, jak záznam vytvořit. Zaznamenané vzorky lze přímo ukládat do souboru na SD kartě. Takový soubor lze označit jako RAW audio data. Ty však nelze nijak přehrát, jelikož se jedná pouze o vzorky a není určena vzorkovací frekvence, bitová hloubka, ani počet kanálu. I přesto mohou být tyto data užitečná v případě dopředu nastavených parametrů, které jsou stejné jako parametry v programu určeném pro jejich přehrávání. Kódově náročnější, avšak uživatelsky mnohem přívětivější je vytvoření souboru s datovou hlavičkou, ve které jsou uloženy zmíněné informace o způsobu záznamu. Konkrétní volbou je datový typ WAVE, který ukládá vzorky bez komprese, je to tedy obecně bezztrátový kodek i přes možnost komprese. Na jeho počátku je zmíněná datová hlavička, která obsahuje 44 bajtů a pevně určuje parametry zvukového záznamu.

První část hlavičky má 12 bajtů a označuje typ uspořádání sektorů – RIFF. Využívají ho i jiné typy multimediálních souborů nejen WAVE. Tímto označením rozdělíme zbytek dat na

dva pod-sektory, a to fmt a data. Sektor fmt určuje konkrétní parametry souboru jako je právě bitová hloubka, počet kanálu, datový tok, vzorkovací frekvence. Další sektor data, jak plyne z názvu, obsahuje samotné vzorky a celkovou velikost souboru. Pokud tedy chceme vytvořit záznam jako WAVE soubor, musíme nejprve do souboru zapsat tuto datovou hlavičku, než do něj budeme vkládat jednotlivé buffery dat.

4.6.2 Kódové řešení záznamu

Záznam je započat stiskem tlačítka, které zavolá funkci startRecord() a ta zprostředkuje počáteční kroky. Prvně se ujistí, že je v pořádku SPI komunikace se zvoleným chip select pinem. Poté vytvoří datovou hlavičku celého wave souboru s předem danými parametry. Následně vytvoří soubor na kartě s daným názvem, otevře jej pro zápis a jako první запиše zmíněnou hlavičku. Při celém záznamu zůstává soubor načtený, dokud není záznam ukončen.

```
void startRecord() {
    rec = true;
    if (!SD.begin(SD_CS))
        Serial.println("SD begin failed");
    while (!SD.begin(SD_CS)) {
        Serial.print(".");
    }
    CreateWavHeader(header, waveDataSize);
    sprintf(filename, "HeartRecording%N.wav", fileCounter);
    SD.remove(filename);
    file = SD.open(filename, FILE_WRITE);
    if (!file) return;
    file.write(header, headerSize);
}
```

Kódová část 7 Funkce počátku záznamu

Ve funkci, která řeší veškerou práci s daty ReadProcessWrite(), je pomocí jednoduché if podmínky zvolené kopírování filtrovaných dat v bufferu do souboru na SD kartě.


```

void ReadProcessWrite() {
...
if (rec == true) {
    recCycles++;
}

if ((rec == true)) {
    file.write((uint8_t*)data, nData * sizeof(data[0]));
}

if (recCycles == (((waveDataSize / (sizeof(data[0]))) / nData)))) {
    endRecord();
}
...
}

```

Kódová část 8 Řešení záznamu pomocí příznaku nahrávání a přičítající se hodnotě počtu uplynulých cyklů

Záznam je pevně nastaven na 10 sekund a od toho se i odvíjí celkový počet proběhlých cyklů, než dojde k shoení příznaku rec. Celková velikost souboru je spočtena z vzorkovací frekvence a bitové hloubky jednoho kanálu.

$$\mathit{waveDataSize} = \mathit{recod.time} \cdot \mathit{I2S.SAMPLE.RATE} \cdot \mathit{sizeof(int16.t)} \quad (4.12)$$

Pokud chceme zaznamenat 10 sekund je celkový počet cyklů roven:

$$\mathit{recCycles} = \frac{\mathit{waveDataSize}}{\mathit{sizeof(data[0])}} \div \mathit{nData} \quad (4.13)$$

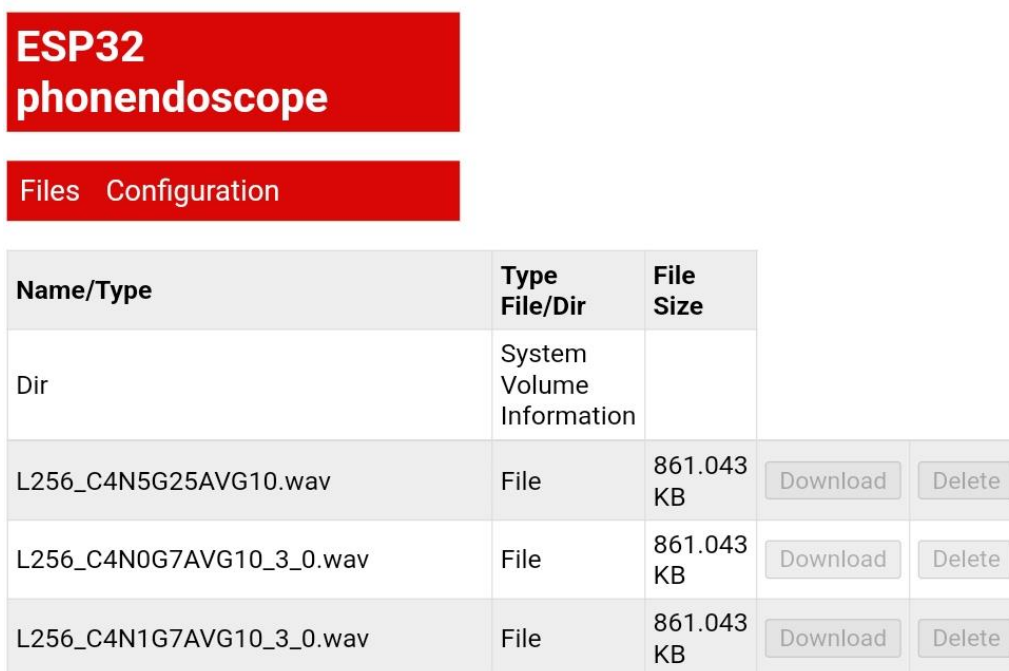
Každý průběh cyklu tedy přičte hodnotu o 1 a pokud je rovna celkovému počtu cyklů, tak se zavolá funkce end_record(), která soubor uzavře a shodí příznak rec na 0. Oznámení o ukončení záznamu je pomocí zvukového signálu. Takto uložený soubor je připraven ke stažení přes webový interface.

4.7 Wifi server

Jak již bylo zmíněno, esp32 je schopné bezdrátové wifi komunikace. Interní požadavky na funkčnost wifi připojení si vyžadují, aby veškerá wifi činnost probíhala bez přerušení. To

je díky dvoujádrovému MCU bezproblémové, a tak není třeba při používání wifi pozastavovat zbytek kódu, který zároveň běží na druhém jádře.

Použití knihovny ESP32WebServer umožňuje jednoduchou implementaci wifi serveru spolu se všemi funkcemi ohledně práce se soubory. Použitím CSS hlavičky webové stránky vytvoříme webové rozhraní, které zobrazuje adresář souborů s jejich názvem a velikostí. U každého z nich pak přidává možnost ho stáhnout nebo případně smazat. Po inicializaci serveru je pak v loop funkci zastoupen pouze `server.handleClient()`, který zprostředkovává požadavky připojených uživatelů. Rychlost wifi připojení je výrobcem deklarovaná až na 150Mbit/s. Při stahování souboru z SD karty je však reálná rychlost stahování okolo 30kBps. Například 10s soubor má okolo 861 kB a stahuje se asi 27 sekund.



Name/Type	Type File/Dir	File Size		
Dir	System Volume Information			
L256_C4N5G25AVG10.wav	File	861.043 KB	Download	Delete
L256_C4N0G7AVG10_3_0.wav	File	861.043 KB	Download	Delete
L256_C4N1G7AVG10_3_0.wav	File	861.043 KB	Download	Delete

Obrázek 24 Webová stránka s adresářem SD karty umožňující stažení záznamů

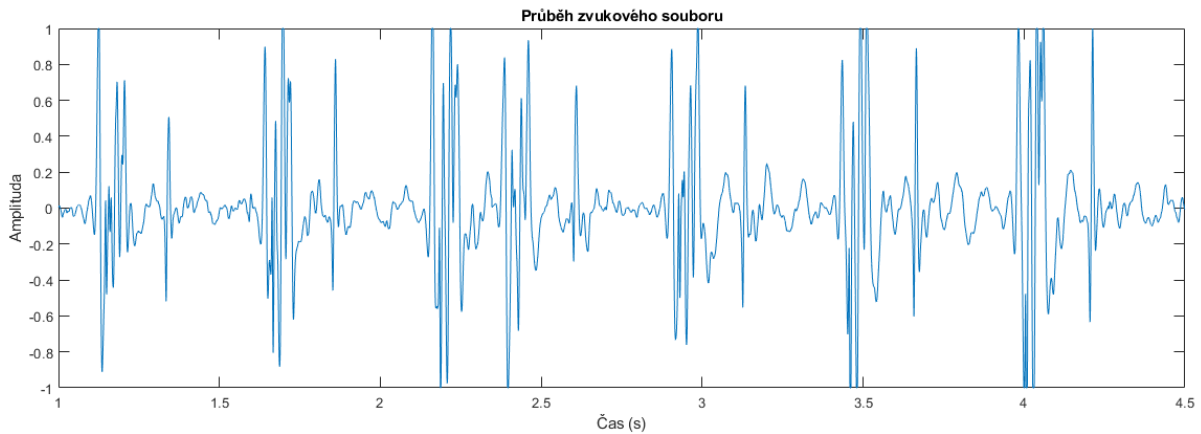
4.8 Srovnání získaných průběhů

Měření srdeční činnosti bylo provedeno pro tři scénáře fyzického zatížení. První měření proběhlo po naprostém 10minutovém klidu v sedě. Očekávaný průběh je srdeční tep s periodou jedné sekundy a více což odpovídá tepu 60 a méně úderů za minutu. Druhý scénář byl po lehké zátěži, která by měla zvednout jak tep, tak i intenzitu úderů, tedy i amplitudu výkmitu. Třetí test byl po intenzivní zátěži s přidanou váhou. Při této zátěži bije srdce ještě rychleji, a i intenzita úderu se blíží maximu. Všechny průběhy jsou filtrovány v reálném čase

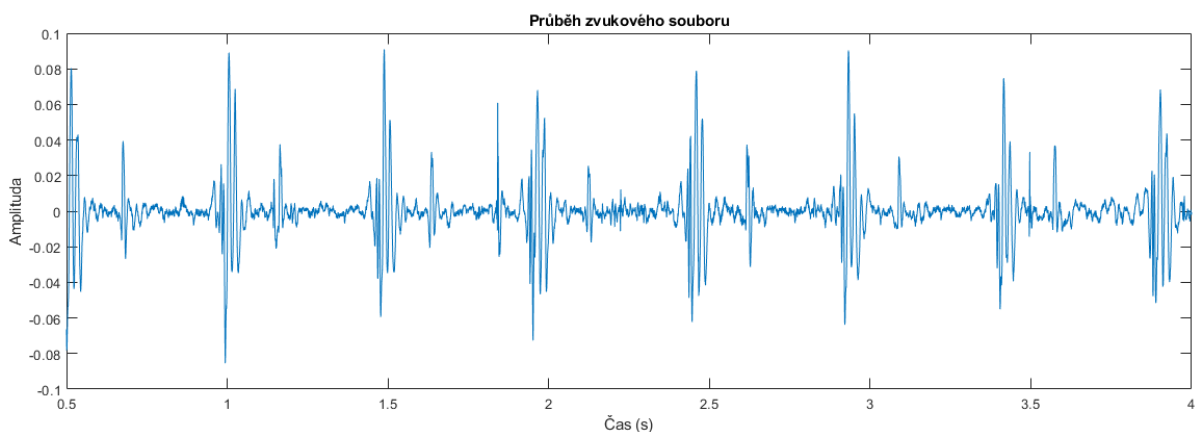
filtrem zmíněným v kapitole 4.4.3. Zobrazeny jsou nejrepresentativnější 3,5 sekundy záznamu.



Obrázek 25 PCG signál s nulovým fyzickým zatížením



Obrázek 26 PCG signál se středím zatížením



Obrázek 27 PCG signál se snahou největšího zatížení

První průběh indikuje jak malou amplitudu signálu okolo jedné třetiny maximálního rozsahu, tak i delší periodu srdečního cyklu. Druhý záznam po mírné zátěži je z hlediska signálu přebuzen, ale zároveň dosahuje plného výkmitu a tedy i rozlišení. V záznamu je vidět

kratší perioda tepu a méně klidná systola i diastola. Ve třetím záznamu je fyzická zátěž největší, ale i tak je amplituda pouze na desetinu maximálního rozsahu. Tep je však znatelně nejrychlejší.

Při těchto testech se tak prokázala silná závislost přiložení hlavy fonendoskopu na tělo pacienta. Intenzita přitlaku má velký vliv na získaný signál. Přesto že se měření v těchto třech případech provádělo stejným způsobem, tak se nepodařilo zajistit stejné podmínky přitlaku. Nutno poznamenat, že i místo poslechu ovlivňuje intenzitu signálu, ale tento faktor je snadněji ovlivnitelný, takže není hlavním problémem.

5 Zhodnocení a závěr

Základem bylo úspěšné nastavení komunikace periferií přes I2S interface. S tímto funkčním datovým tokem šlo použitím MEMS mikrofonu převádět akustické vlny produkované srdcem do elektronické digitální podoby. Zvážením prošel i elektretový mikrofon, ale pro jeho komplexnější zapojení do systému a malý přínos zvukové kvality byl vyloučen. Problém nastal při návrhu akustických filtrů, které mají za úkol přiblížit posluchači pouze ty frekvence, které jsou pro srdce charakteristické. Kódová implementace univerzálního FIR filtru s volbou délky koeficientů se nepodařila. Filtr vykazoval nulové hodnoty na konci každého bufferu, který by měl správně navazovat na předchozí. Proto byl změněn za jednodušší filtr. Vzorky zpracovává podobně, akorát místo násobení jednotlivých vzorků různými koeficienty je všechny násobí jedním, a to konkrétně obrácenou hodnotou jejich celkového počtu.

S délkou 10 vzorků měl tento filtr dolní propusti mezní frekvenci f_3 2250 Hz. Výhodou tohoto filtru je odstranění náhodného šumu vyšších frekvencí. Kontrola doby výpočtu s velkou rezervou prošla maximálním přípustným zpožděním, proto byl tento filtr zvolen jako primární možnost. Druhá volba je bez filtrování pro možnost externího upravení záznamu. Velkou výhodou je záznam signálu do wave souboru na připojenou micro SD kartu.

Mikrokontroler umožňuje přes online wifi server po připojení nahrané soubory stáhnou přes přehledný výpis adresáře. Akustický výstup při poslechu pacienta je v tomto návrhu vyřešen externím I2S zesilovačem, který zvuk směřuje do připojeného pasivního reproduktoru. Výsledný záznam lze v akustickém programu jako je například Audacity zobrazit a zpracovat dále.

Dosažený výkmit při srdečních ozvách je vůči zbytku záznamu dostatečně výrazný, a tak je možná i optická kontrola auskultace čistě ze zobrazeného průběhu signálu. Vstupní signál je lehce degradovaný díky použití mikrofonu, který není přizpůsobený pro velmi nízké až ultrazvukové frekvence. Také jeho bitová hloubka není maximálních 24bitů, kterou se nepovedlo zprovoznit, ale 16bitů. Efektivní bitová hloubka je pak 13,5bit. Kvalita záznamu také neprošla vhodným lékařským ověřením, takže schopnost fonendoskopu posloužit jako aktivní přístroj při poslechu si nedovolují deklarovat. Stejně tak způsob mechanického řešení hlavy fonendoskopu je pouze ilustrační a jeho úkolem je pouze otestovat vlastností samotného zařízení, nikoliv funkčnost při reálném použití.

V případě rozšiřování tohoto projektu bych jako autor doporučil vyřešit kvalitu digitálního filtru, v případě potřeby izolace konkrétního frekvenčního spektra. Důležitou částí by také mělo být zlepšení integrace mikrofonu do poslechové hlavy, čímž by se dramaticky zlepšily parametry výsledného signálu. Dle aplikace by bylo možné zajistit výstup do sluchátek, místo akustického výstupu do reproduktoru, pomocí sluchátkového zesilovače.

I přesto se však povedlo získat pomocí levného mikrokontroleru signál s průběhem tvarově odpovídajícím již hotovým fonendoskopům, a možnost uložení záznamu dále posouvá dostupnost telemedicíny se snižováním nákladů za použité součástky.

Literatura

- [1] Castaneda D, Esparza A, Ghamari M, Soltanpur C, Nazeran H. A review on wearable photoplethysmography sensors and their potential future applications in health care. *Int J Biosens Bioelectron.* 2018;4(4):195-202. [cit. 25.4.2023] doi:10.15406/ijbsbe.2018.04.00125. Epub 2018 Aug 6. PMID: 30906922; PMCID: PMC6426305. Dostupné z NationalLibraryOfMedicine: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6426305/>
- [2] Abay TY, Kyriacou PA. Photoplethysmography for blood volumes and oxygenation changes during intermittent vascular occlusions. *J Clin Monit Comput.* 2018 Jun;32(3):447-455. [cit. 20.5.2023] doi: 10.1007/s10877-017-0030-2. Epub 2017 May 25. PMID: 28547651; PMCID: PMC5943414. Dostupné z NationalLibraryOfMedicine: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5943414/>
- [3] Chow HW, Yang CC. Accuracy of Optical Heart Rate Sensing Technology in Wearable Fitness Trackers for Young and Older Adults: Validation and Comparison Study. *JMIR Mhealth Uhealth.* 2020 Apr 28;8(4):e14707. [cit. 3.2.2023] doi: 10.2196/14707. PMID: 32343255; PMCID: PMC7218601. Dostupné z NationalLibraryOfMedicine: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7218601/>
- [4] Aziz S, Khan MU, Alhaisoni M, Akram T, Altaf M. Phonocardiogram Signal Processing for Automatic Diagnosis of Congenital Heart Disorders through Fusion of Temporal and Cepstral Features. *Sensors (Basel).* 2020 Jul 6;20(13):3790. [cit:22.5.2023] doi: 10.3390/s20133790. PMID: 32640710; PMCID: PMC7374414. Dostupné z NationalLibraryOfMedicine: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7374414/>
- [5] Espressif Systems (Shanghai) Co., Ltd. ESP-IDF Programming Guide [online]. Springer, 2023 [cit. v5.0.2]. Dostupné z Doc.Espressif: <https://docs.espressif.com/projects/esp-idf/en/v5.0.2/esp32/index.html>
- [6] DanielChangMD. Porovnání čtyř výstupů srdečních cyklů [obrázek] WikimeidaCommons, 2016 [cit. 26.5.2023] Dostupné z: https://commons.wikimedia.org/wiki/File:Wiggers_Diagram-ca.svg#file

- [7] Littmann. Digitální stetoskop firmy Littman [foto] Littmann stethoscopes [cit. 26.5.2023] Dostupné z: <https://multimedia.3m.com/mws/media/1865486P/3m-littmann-core-stethoscope-8480-01-hero-hr.jpg>
- [8] Thinklabs. Digitální stetoskop Thinklabs [foto] Thinklabs One [cit. 26.5.2023] Dostupné z: <https://www.thinklabs.com/plug-and-practice>
- [9] Espressif Systems (Shanghai) Co., Ltd. Vývojový modul esp32 wroom [foto]. Springer, 2023 [cit. v5.2]. Dostupné z: Doc.Espressif: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html>
- [10] Femur. Lékařský fonendoskop. [foto]. Femur.cz sro. Dostupné z: <https://femur.cz/jak-vybrat-fonendoskop/>
- [11] PhDr. Mgr. Lenka Adámková, Ph.D. Průtok krve srdcem. [obrázek]. MUNI. 2016 [cit. 26.5.2023] Dostupné z: https://is.muni.cz/el/1441/podzim2016/ZS1BP_ZBZ1/um/Kardiovaskularni_sousta_va.pdf
- [12] Madhero88 Průběhy signálu srdečních chorob ve fonokardiogramu [obrázek] WikimeidaCommons, 2020 [cit. 26.5.2023] Dostupné z: https://commons.wikimedia.org/wiki/File:Phonocardiograms_from_normal_and_abnormal_heart_sounds.png
- [13] Had, Anas & Sabri, K. & Aoutoul, Mohssin. Typický průběh akustického záznamu se srdečními ozvy S1 a S2, mezi nimi systola a diastola [obrázek]. ResearchGate. 2020 [cit. 26.5.2023] Dostupné z: https://www.researchgate.net/figure/Normal-phonocardiogram-signal-for-a-single-cardiac-cycle_fig1_338803547