



České
vysoké
učení technické
v Praze

Návrh a realizace zařízení pro monitorování fyzické aktivity

Dominik Patzák

Školitel: Ing., Ph.D. Alexandr Laposka
Květen 2024

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Patzák** Jméno: **Dominik** Osobní číslo: **498975**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra měření**
Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Návrh a realizace zařízení pro monitorování fyzické aktivity

Název bakalářské práce anglicky:

Design and implementation of device for the monitoring of physical activity

Pokyny pro vypracování:

1. Prostudujte základní metody a technická řešení při monitorování fyzické aktivity pomocí nositelné elektroniky (fitness náramky a chytré hodinky).
2. Navrhněte a realizujte technické řešení pro zařízení, které umožňuje měření parametrů jako jsou tepová frekvence, saturace krevního hemoglobinu kyslíkem, odhadovaná ušlá vzdálenost a počet kroků. Realizované zařízení by mělo splňovat tyto základní technické požadavky: mikrokontroler STM32, maximální rozměry 50 mm x 50 mm x 30 mm, LCD displej, bateriové napájení, Android aplikace pro zobrazení dat pomocí Bluetooth.
3. Realizované řešení ověřte a zhodnoťte dosažené výsledky.

Seznam doporučené literatury:

- [1] A. Ometov et al., "A Survey on Wearable Technology: History, State-of-the-Art and Current Challenges," Comput. Networks, vol. 193, p. 108074, Jul. 2021, doi: 10.1016/j.comnet.2021.108074.
- [2] T. T. Pham and Y. S. Suh, "Walking Step Length Estimation Using Waist-Mounted Inertial Sensors with Known Total Walking Distance," IEEE Access, vol. 9, pp. 85476–85487, 2021, doi: 10.1109/ACCESS.2021.3087721.
- [3] X. Kang, B. Huang, and G. Qi, "A novel walking detection and step counting algorithm using unconstrained smartphones," Sensors (Switzerland), vol. 18, no. 1, p. 297, Jan. 2018, doi: 10.3390/s18010297.
- [4] Open Watch Project, <https://github.com/SMotlaq/open-watch>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Alexandr Lapos, Ph.D. katedra mikroelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **26.01.2024**

Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce:

do konce letního semestru 2024/2025

Ing. Alexandr Lapos, Ph.D.
podpis vedoucí(ho) práce

_____ podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

_____ Datum převzetí zadání

_____ Podpis studenta

Poděkování

Tímto bych chtěl poděkovat vedoucímu Ing., Phd. Alexandrovi Laposovi za pomoc během práce. Zároveň bych chtěl poděkovat mé rodinně a přítelkyni za podporu.

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 24.5.2024

Abstrakt

Cílem práce je návrh a realizace chytrých hodinek. Dílčím cílem je vytvoření úvodu do problematiky chytrých hodinek jak z pohledu teoretického, tak praktického. Navržená řešení jsou prouzkoumána a popsána v několika kapitolách.

První se zaměřuje na software a obsahuje algoritmy pro senzory, zpracování dat, chod mikrokontroléru a další periferie. Druhá část popisuje na uživatelské rozhraní, kde je popsán design Android aplikace a použitý grafický LCD display. Ve třetí části je potom shrnuta implementace plošného spoje na čtyřvrstvé desce. Poslední část řeší návrh pouzdra pro samotné hodinky.

Klíčová slova: Chytré hodinky, Plošný spoj, Senzory, 3D návrh

Školitel: Ing., Ph.D. Alexandr Laposa

Abstract

The goal of this bachelor's thesis is to introduce the reader to smart wearable fitness trackers, exploring both theory and possible designs. My solutions are explored and described in three separate chapters.

The first is focused on software, which contains algorithms for sensor data processing, MCU, and other peripherals. Second chapter is aimed at user interface, describing the developed Android application and a graphical LCD display. The third chapter focuses on the implementation of the fitness tracker on a printed circuit board. Final chapter provides solutions for a case design.

Keywords: Smart watch, PCB, Sensors, 3D design

Title translation: Design and implementation of device for the monitoring of physical activity

Obsah

1 Úvod	1	3.1.2 Grafické zpracování	23
2 Software a periferie	3	3.2 Android aplikace	23
2.1 Detekce chůze pomocí akcelerometru	3	3.2.1 Cyklus aplikace	24
2.1.1 Teorie operace	3	3.2.2 Domovská stránka	24
2.1.2 Výběr akcelerometru	4	3.2.3 Stránka s grafy	24
2.1.3 Zpracování zrychlení	4	3.2.4 Informační stránka	25
2.1.4 Detekce chůze a počítání kroků	5	4 Desky plošných spojů	27
2.1.5 Filtrování	6	4.1 Deska hodiněk	27
2.1.6 Algoritmus detekce chůze	6	4.1.1 Obecné směrnice	27
2.1.7 Další aktivity	8	4.1.2 Prototypy	27
2.1.8 Měřené experimenty	8	4.1.3 Napájení	28
2.1.9 Diskuze nad možnými řešeními	9	4.1.4 Přepínání mezi zdroji	28
2.2 Senzor měření tepové frekvence a oxidace krve	9	4.1.5 Regulátor	28
2.2.1 Použité senzory	9	4.1.6 Procesor	28
2.2.2 Teorie operace	10	4.1.7 Krystal	29
2.2.3 Zpracování dat	10	4.1.8 Displej	29
2.2.4 Algoritmus pro rozeznání tepové frekvence	10	4.2 Deska oxymetru	29
2.3 Paměť EEPROM	13	4.3 Redukční deska	30
2.4 Rozvod napájení a baterie	14	4.4 Výsledný návrh, osazování	30
2.5 Struktura mikrokontroléru a hlavního programu	15	5 Pouzdro	31
2.5.1 Chod programu a jeho struktura	15	5.1 Horní část	31
2.6 Hodiny	15	5.2 Prostřední část	31
2.7 Debug a flashování programů	16	5.3 Spodní část	32
2.8 Komunikace s externími zařízeními	17	5.4 Víko	32
2.8.1 UART	17	5.5 Zhodnocení a výsledky pouzdra	33
2.8.2 SPI	18	6 Závěr	35
2.9 Bluetooth komunikace s Android	19	Literatura	37
2.9.1 Modul HC-05	19	A	39
2.9.2 Struktura komunikace	20		
2.9.3 Princip komunikace	20		
2.9.4 Zpráva pro časové nastavení	21		
2.9.5 Synchronizační zpráva	21		
2.9.6 Zpráva sloužící pro uložení do databáze	21		
2.9.7 Zpráva sloužící pro odeslání dat z EEPROM	21		
3 Uživatelské rozhraní	23		
3.1 Displej	23		
3.1.1 Způsob komunikace	23		

Obrázky

2.1 Diferenční uspořádání elektrod akcelerometru [15]	4
2.2 Mechanický model akcelerometru [10]	5
2.3 Zrychlení během chůze	6
2.4 Detekce kroků při chůzi	7
2.5 Třesení rukou	8
2.6 Princip pulzní oxymetrie [15] ...	11
2.7 Signál z oxymetru s odfiltrovanou DC složkou	12
2.8 Filtrovaný signál z oxymetru ...	13
2.9 Zapojení NTC termistoru [21] ..	15
2.10 Princip zapojení UART [14] ...	17
2.11 Datový paket UARTu [9]	18
2.12 Struktura I^2C rozhraní [7]	19
2.13 Datový paket I^2C [7]	20
2.14 Multi-Slave SPI struktura [8] ..	20
5.1 Navržené části pouzdra	32
A.31 Schéma plošného spoje	40
A.1 Použité piny procesoru (Převzato z prostředí STM32CUBEIDE)	45
A.2 Osazená deska - vrstva top	46
A.3 Osazená deska - vrstva bot	47
A.4 Schéma USB vstupu	48
A.5 Plošný stop - vrchní signálová vrstva	48
A.6 Plošný spoj - spodní signálová vrstva	49
A.7 První prototyp - Spodní vrstva ..	49
A.8 První prototyp - Vrchní vrstva ..	50
A.9 Plošný spoj - Napájecí vrstva ..	50
A.10 Plošný spoj - Zemnicí vrstva ..	51
A.11 Oxymetr - Vrchní vrstva	51
A.12 Oxymetr - Spodní vrstva	52
A.13 Druhý prototyp - Spodní vrstva ..	53
A.14 Druhý prototyp - Vrchní vrstva ..	54
A.15 Redukční deska	55
A.16 Pouzdro - Vrchní část	56
A.17 Pouzdro - prostřední část	56
A.18 Pouzdro - spodní část, pohled shora	57
A.19 Pouzdro - spodní část, pohled zdola	57
A.20 Pouzdro - víko na oxymetr ...	58
A.21 Výsledné pouzdro - Pohled 1..	59
A.22 Výsledné pouzdro - Pohled 2..	60
A.23 Výsledné pouzdro - Pohled 3..	61
A.24 Displej - domovská obrazovka ..	62
A.25 Displej - vedlejší obrazovka ...	63
A.26 Design krystalu	64
A.27 Aplikace - Domovská stránka ..	65
A.28 Aplikace - Grafy	66
A.29 Vizualizace ušlých kroků	67
A.30 Aplikace - Informace	68

Tabulky

2.1 Použité hodnoty pro algoritmus detekce chůze	5
2.2 Naměřený počet kroků na 100 kroků	9
2.3 Debug rozhraní	16
2.4 Zpráva pro uložení do databáze Android	22
A.1 Výčet pinů procesoru	69

Kapitola 1

Úvod

Návrh funkčních chytrých hodinek je složitý proces, který se skládá z mnoha částí. V této práci se zaměřuji na možné řešení, které rozdělují na tyto kapitoly: software, uživatelské rozhraní, návrh plošných spojů a design pouzdra. V každé z nich komentuji zvolené postupy, výsledky a problémy při vývoji.

V první části se soustředím na zvolené senzory snímající jak chůzi, tak tepovou frekvenci. Uvádím zde navržené algoritmy a prezentuji naměřené výsledky formou grafů. Také se zde nachází zpracování ostatních součástí, jako například EEPROM paměť pro lokální uchování dat a Bluetooth. Kapitola uzavírá hlavní výpočetní jednotka a srdce celého projektu, procesor ARM v mikrokontroléru od firmy ST Microelectronics. Veškerý software hodinek jsem programoval v jazyce C a v prostředí STM32CUBEIDE.

Druhá část se zaměřuje na uživatelské rozhraní, které jsem řešil návrhem jednoduché Android aplikace, která umožňuje zobrazit nasbíraná data jak v čase reálném, tak i v minulém, kde je možné si data zobrazit v grafech. O prezentaci dat se stará i kruhový grafický LCD display připevněný na desku hodinek.

V předposlední části uvádím svoji implementaci hodinek na plošném spoji, výběr a umístění jednotlivých součástí, prototypy hlavní desky hodinek, desky oxymetru a redukční desky a v poslední řadě postup při osazování a problémy, se kterými jsem se setkal.

Poslední kapitola uvádí čtenáře do problematiky návrhu tištěného pouzdra, jeho rozdělení do čtyř částí a způsob zaklapávání částí do sebe. Zároveň obsahuje návrh nožiček pro uchycení řemínku a prezentuje výsledný tvar hodinek.

Kapitola 2

Software a periferie

2.1 Detekce chůze pomocí akcelerometru

2.1.1 Teorie operace

Akcelerometry jsou mikromechanické systémy (MEMS) [11], které kombinují mechanické a elektrické komponenty dohromady na škále mikrometrů. Jejich hlavním úkolem je měření zrychlení, které můžeme obecně derivovat z prvního Newtonova zákona:

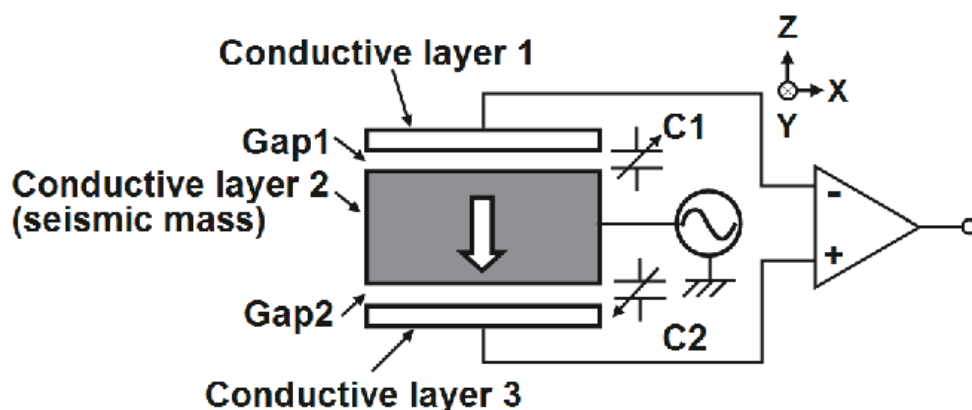
$$\vec{a} = \frac{\vec{F}}{m} \quad [m \cdot s^{-2}] \quad (2.1)$$

V případě akcelerometrů se ve většině případů počítá pomocí snímání přemístění elektrod od sebe, neboli vzdalování jedné flexibilní elektrody od dvou upevněných elektrod. Kapacita se v tomto případě dá získat zjednodušeně ze vzorce pro deskový kondenzátor:

$$C = \epsilon_0 \epsilon_r \frac{S}{d} \quad [F] \quad (2.2)$$

Kde C je výsledná kapacita, ϵ_0 je elektrická permitivita vakua, ϵ_r je relativní elektrická permitivita, S je efektivní plocha elektrod a d je jejich vzdálenost. Uspořádání elektrod, které může být buď jednostranné, nebo diferenční, určuje možnosti detekce zrychlení. V práci jsem použil právě diferenční akcelerometr, který je schopný zjistit obě polaritu zrychlení, jak je možné vidět na obrázku 2.1. To je dosaženo pomocí odečtení kapacit C_1 a C_2 od sebe.

Aby bylo dosaženo kapacity, kterou je možné snímat, páry elektrod jsou řazeny paralelně a výsledek se sčítá, viz obrázek 2.2. Změřené napětí, které získáme jakožto napěťový dělič kapacit C_1 a C_2 je zesíleno, filtrováno a demodulováno. Následně je převedeno pomocí Sigma-Delta AD převodníku do digitální podoby. Tyto kroky jsou přítomny pouze u digitálních akcelerometrů, analogové prezentují pouze změřené napětí na výstupních pinech formou napěťové úrovně.



Obrázek 2.1: Diferenční uspořádání elektrod akcelerometru [15]

2.1.2 Výběr akcelerometru

Pro tuto práci jsem zvolil akcelerometr ADXL345 od firmy Analog Devices [5]. Jedná se o 3-osý, digitální akcelerometr, který je vhodný na účely chytrých hodinek. Má velmi nízkou spotřebu, řádově $25 \mu A$, $3,3V$ napájení, podporuje sériovou komunikaci ve formě I^2C rozhraní a řadu funkcí je možné na něm nastavit. Ačkoliv akcelerometr umožňuje snímat zrychlení až $\pm 16g$, pro hodinky stačí pouze $\pm 2g$, jak je možné vidět v sekci u měření.

2.1.3 Zpracování zrychlení

Ke snímání zrychlení jsem použil vzorkovací frekvenci 50 Hz a rozlišení 10 bitů, je tedy možné snímat s přesností $4mg/LSB$. Akcelerometr jsem nastavil tak, aby data uchovával v FIFO poli o délce 32 vzorků a posílal přerušeni, jakmile je pole plné.

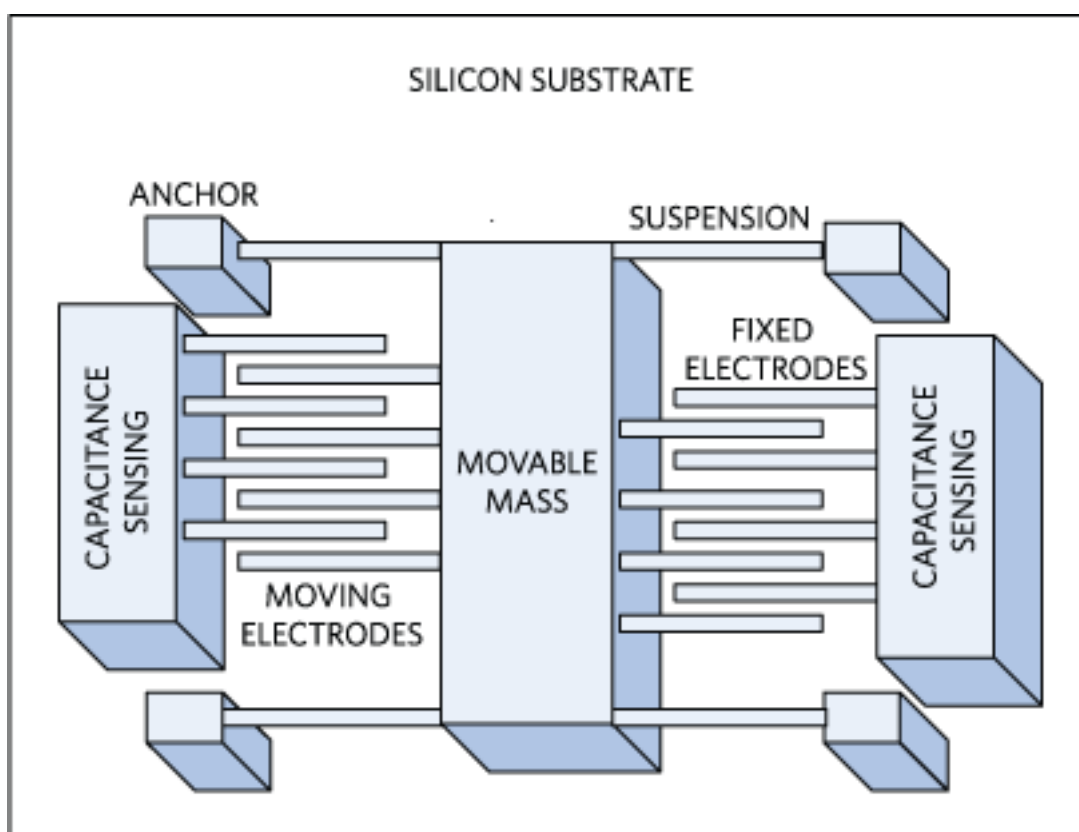
Každý vzorek se skládá ze tří 16 bitových čísel, každé odpovídající své ose zrychlení, tedy X, Y a Z a musí být čten ze 4 registrů. Po přečtení akcelerometr vysune nejstarší vzorek z FIFO do datového registru, kde se data opět mohou přečíst. Tuto operaci v případě plného pole je tedy nutné opakovat 32-krát. Protože orientace hodinek na zápěstí není fixní, není možné přesně určit orientaci akcelerometru v prostoru. Tudíž je z každého vzorku spočtena velikost zrychlení, spočtena pomocí:

$$|\vec{a}| = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (2.3)$$

a_x , a_y , a_z jsou jednotlivé složky zrychlení, které je možné získat jako

$$a = \frac{x \cdot g}{2^{10}} \quad [m \cdot s^{-2}] \quad (2.4)$$

Kde x je 10-bitové výstupní slovo akcelerometru pro danou osu a $g = 9.81$ je tíhové zrychlení. Výstupní bitové slovo se musí transformovat do 16-bitového a znaménkově rozšířit, aby bylo možné s ním dále pracovat. Výsledná velikost zrychlení je uchovávána v kruhovém poli, kde je následně zpracovávána dále.



Obrázek 2.2: Mechanický model akcelerometru [10]

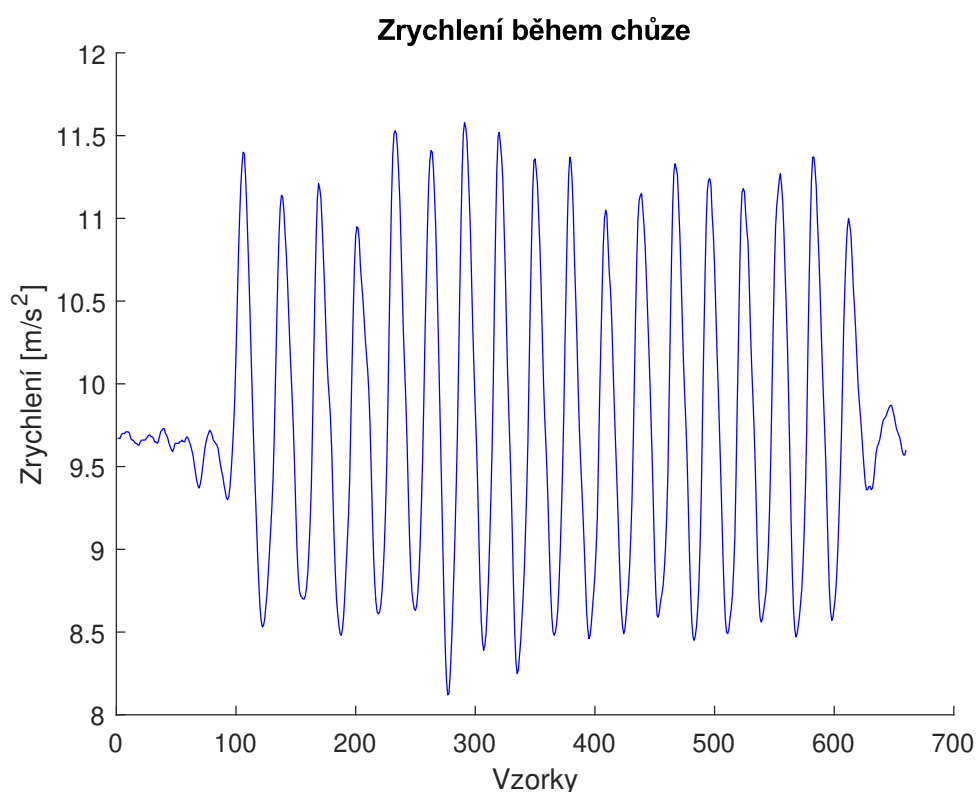
Parametr	Hodnota
t_{win}	0.6 s
t_{avg}	0.3 s
T	$10.5 \text{ m} \cdot \text{s}^{-2}$

Tabulka 2.1: Použité hodnoty pro algoritmus detekce chůze

V poslední řadě je nutné akcelerometr kalibrovat při prvotním nastavení, aby výstup v klidovém stavu odpovídal zrychlení $a = g$, aby algoritmy, které hledají aktivitu byly funkční. Toho se dá částečně docílit použitím registrů akcelerometru pro jednotlivé offsety os, kde je dostačující rozlišení 15mg/LSB.

2.1.4 Detekce chůze a počítání kroků

Rozhodl jsem se pro algoritmus v časové doméně, protože má nízkou výpočetní náročnost a protože přesnost ostatních algoritmů je podobná, ne-li shodná s ostatními, řádově 2-5% rozdíl chybě počítaných kroků. Změřené zrychlení je vhodné filtrovat pomocí aritmetického průměru $t_{avg} = 0,3 \text{ s}$, a následně použít okno o šířce $t_{win} = 0,6 \text{ s}$, kde se porovnávají hodnoty s hranicí $T = 10.5 \text{ m} \cdot \text{s}^{-2}$ [4].



Obrázek 2.3: Zrychlení během chůze

2.1.5 Filtrování

Prvním krokem je digitální filtrování naměřených hodnot. Bereme-li v potaz vzorkovací frekvenci 50 Hz a délku filtračního okna 0,3 s, dostaneme šířku 15 vzorků. Pro každé naměřené zrychlení je spočten aritmetický průměr, kde vzorek je ve středu okna, tj. počítáme se 7 vzorky na obě strany. Prvních 7 vzorků, co akcelerometr nasbírá po svém spuštění se nebere v potaz a filtruje se až od 8. dále. Pro výpočet průměru použijeme vztah:

$$x_k = \frac{\sum_{j=k-7}^{k-1} x_j + x_k + \sum_{l=k+1}^{k+7} x_l}{15} \quad (2.5)$$

Kde $k \in \{0; 31\}$ je pořadí vzorku, vyjma pravidla zmíněného výše. Odfiltrované zrychlení při chůzi lze pozorovat na snímku 2.3.

2.1.6 Algoritmus detekce chůze

Po odfiltrování je možné přistoupit na samotnou detekci chůze a počítání kroků. V definovaném okně, které má v tomto případě šířku $0,6 \cdot 50 = 30$ vzorků, následuje hledání lokálních maxim. Tedy ověřuje se vztah, že platí :

$$x_k > x_{k-1} \quad \& \quad x_k > x_{k+1} \quad (2.6)$$

Vzhledem ke kvalitnímu odfiltrování není nutné přidávat další parametry, které by nadále omezovali nalezená maxima, jako například minimální vzdálenost



Obrázek 2.4: Detekce kroků při chůzi

od lokálního minima mezi jednotlivými vzorky.

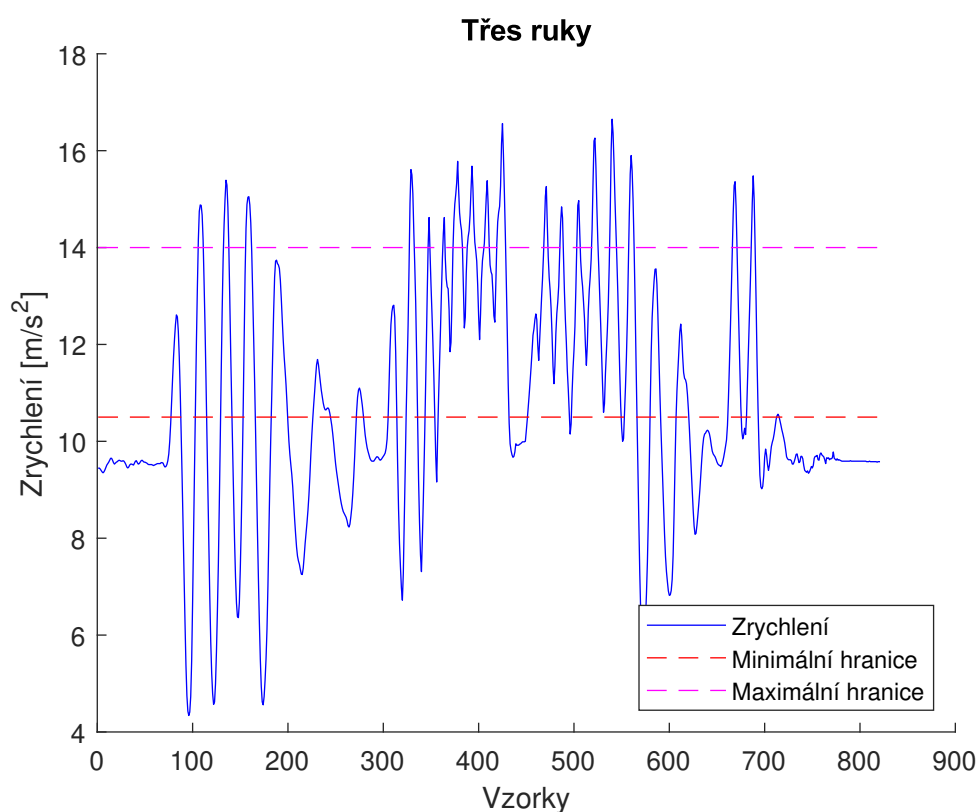
Maximum je nadále porovnáváno s rozhodovací úrovní T a omezeno zhora úrovní $T_2 = 14m \cdot s^{-2}$ pro lepší robustnost systému vůči dalším pohybům, jako je třes ruky. Dále musí mít maximum vzdálenost od posledního nalezeného maxima alespoň 20 vzorků a nanejvýš 60 vzorků. Pomalejší nebo rychlejší pohyb se neuvažuje.

Zároveň musí pro detekci chůze platit, že došlo k alespoň dvěma po sobě jdoucím krokům, aby nedocházelo k zachycení jednorázových aktivit, jejichž charakteristika je totožná s krokem. V tomto případě lze zvolit vyšší počet, kdy se na jednu stranu zvyšuje robustnost systému vůči jiným aktivitám, ale snižuje se schopnost počítat kroky. Tedy stanovíme-li nutnost na alespoň 7 po sobě jdoucím krokům a učiníme pouze 6, algoritmus je nezapočítá.

Vyhovuje-li tedy maximum všem zmíněným podmínkám, je započteno jako krok. Pokud se jedná o první krok v možné chůzi, je započten až zpětně. Algoritmus vyhodnotí, že chůze skončila, nedojde-li během 150 vzorků, tj. 2,5 sekundy k nalezení následujícího kroku. Nalezení kroků po začátku chůze lze pozorovat na grafu 2.4.

■ Odhad ušlé vzdálenosti

Z měřených experimentů jsem nebyl schopen zjistit délku kroku pouze na základě zrychlení. Byl jsem tedy nucen použít statický odhad vzdálenosti,



Obrázek 2.5: Třesení rukou

který s každým ušlým krokem přičte nemění se hodnotu. Protože chyba takto měřené vzdálenosti bývá nejméně 20%, rozhodl jsem se nevizualizovat tuto informaci vzhledem k tomu, že počet kroků udává přesnější informaci o aktivitě.

2.1.7 Další aktivity

Jsou i další pohyby, jejichž zrychlení má podobný charakter jako chůze, které jsou nežádoucí a které mohou chybně zvýšit počet kroků. Mezi tyto pohyby patří především třesení ruky, ale například i vstávání ze židle. Algoritmus nedokáže vždy vyřadit tyto nechtěné vlivy a kroky připočte. Navodí-li se například třesení ruky o vhodné frekvenci, je výsledná odezva srovnatelná s odezvou u chůze a rozdíl není možné rozeznat. Odezvu lze vidět na obrázku 2.5. Zde je vidět, že většina třesů má vyšší amplitudu a vyšší minima, ale některé pasáže jsou skoro identické s chůzí.

2.1.8 Měřené experimenty

Pro ověření správnosti algoritmu jsem provedl 5 chůzí, s akcelerometrem podél těla v klidu, o délce 100 kroků. Hodnoty spočtených kroků lze vidět v tabulce 2.2. Je zřejmé, že chybovost je velmi nízká a přičítání kroků je správné.

Experiment	Počet kroků
1	93
2	97
3	94
4	98
5	96

Tabulka 2.2: Naměřený počet kroků na 100 kroků

2.1.9 Diskuze nad možnými řešeními

Řešení problému vyhodnocování lze vést dvěma cestami, které jsem pozoroval při měření odezvy zrychlení. První cesta, která byla volena v tomto případě, je skromnějšího charakteru a na vyhodnocování kroku klade přísnější pravidla. Druhá cesta poté tyto pravidla značně odlehčuje, nebo je vynechává. Obě cesty mají své pro a proti.

Výhoda první cesty je, že počítané kroky spíše podhodnocuje a aktivity, jako je třesení ruky či náhle pohyby zpravidla vyhodnotí korektně a kroky nezapočítá. Mezi dohlížené parametry patří minimální výška sloupců, tj. vzdálenost maxima od minima, které se nachází mezi dvěma sloupci a maximální hodnota zrychlení.

Druhá cesta na zmíněné parametry nebere takový zřetel, tedy dochází zpravidla k nadhodnocování počtu kroků. Třesení ruky a další chaotické pohyby zachycuje a nekorektně je připočítává k ušlým krokům. Výhodou však je, že je schopna zachytit i chůzi, kdy dochází ke značnému máchání ruky (například pochod vojáků) a rozpozná i běh.

K dispozici jsem měl hodinky Garmin Venu 2S pro porovnání. Hodinky patří k zástupci druhé cesty, avšak chyby korektuje pomocí GPS lokátoru. Pokud ovšem přístup ke GPS nemá, velmi nadhodnocuje počet kroků. Učinil jsem několik chůzí s počtem kroků 20, kdy jsem použil značný pohyb rukou a kde mé hodinky měly chybu 40%. Hodinky Garmin v tomto ohledu počítali s chybou 90%.

2.2 Senzor měření tepové frekvence a oxidace krve

V dnešní době se pro snímání tepové frekvence používají senzory, které pracují na principu photoplethysmografie[19], jak je tomu i v této práci. Byl jsem nucen použít dva modely, první v modulovém řešení a druhý, pro který jsem desku navrhoval. Bylo to z toho důvodu, že podpora pro první model senzoru skončila, tudíž už nebyl na trhu dostupný. Experimenty jsem prováděl pouze na prvním senzoru, tj. MAX30100.

2.2.1 Použité senzory

Pro vývoj jsem použil senzory od Maxim Integrated, jež jsou ideální právě pro chytré přenosné zařízení. Jedná se o integrované řešení pulzní oxymetrie

a měření tepové frekvence. Infračervené a červené LED jsou napájeny 3,3 V a samotný čip poté 1.8 V. Oba senzory, MAX30100[16] a MAX30101[1] jsou digitální, komunikují pomocí I^2C rozhraní a příliš se od sebe neliší, co se týče stránky programovací i funkční.

2.2.2 Teorie operace

Pulzní oxymetrie pracuje na principu reflektivity světla. Srdce pumpuje krev v pulzech a tyto pulzy lze zaznamenat v krevním tlaku. Svítíme-li například na prst, zápěstí, či ušní lalůček LEDkou, reflektivita klesá a stoupá právě podle srdečních pulzů. LED jsou totiž schopné prosvítit vrstvy pokožky a pozorovat krevní proud. Pro měření tepové frekvence se obvykle používá infračervená led, pro kyslíkovou saturaci vyhovuje LED červená nebo zelená. Prosvícení vrstev pokožky a měření krevního tlaku můžeme vidět na obrázku 2.6. Je nutné zmínit, že měření tepové frekvence pomocí pulzní oxymetrie není příliš přesné a rozhodně neodpovídá kvalitnějším a používaným metodám v lékařském průmyslu, zpravidla je-li senzor umístěn na zápěstí. Chyba měřeného tepu se může pohybovat v hraničních případech až o 15 tepů za vteřinu.

2.2.3 Měření oxidace krve

oxidace u pulzních oxymetrů se měří pomocí dvou LED, jedné infračervené a druhé většinou červené. oxidaci lze spočítat pomocí takzvaného "poměru poměrů" [17]

$$R = \frac{\frac{AC_{red}}{DC_{red}}}{\frac{AC_{infrared}}{DC_{infrared}}} \quad (2.7)$$

Pomocí tohoto poměru lze poté pomocí aproximace zjistit oxidaci krve podle [17]

$$SpO_2 = 104 - 17R \quad (2.8)$$

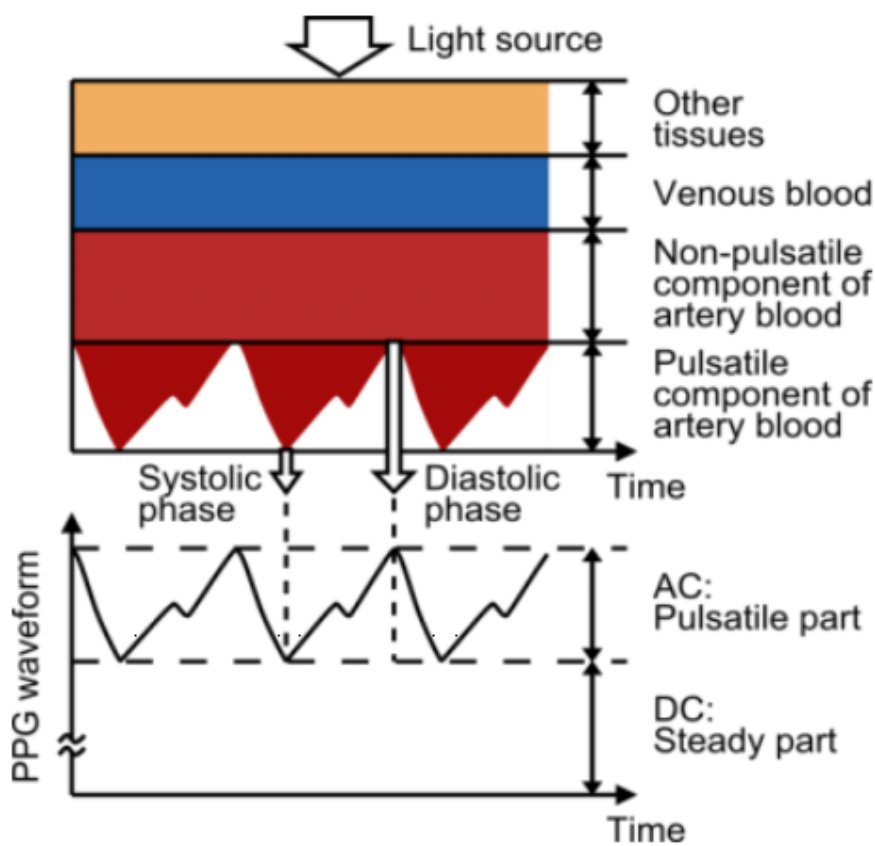
U zdravých jedinců se oxidaci pohybuje kolem 95% - 98%. oxidaci má tedy smysl počítat pouze u osob se zdravotními obtížemi.

2.2.3 Zpracování dat

Vzorkovací frekvenci jsem volil 50 Hz, proud LED 11,6 mA a šířku pulzu LED 1,6 ms. Zároveň jsem povolil ukládání dat do FIFO bufferu na senzoru, takže data jsou vyčítána, jakmile senzor odebere 15 vzorků. Poté je posláno přerušení procesoru, který data vyčte. Každý vzorek má délku 32 bitů, 16 pro infračervenou LED a 16 pro červenou LED. Rozlišení však 16-bitové není, použil jsem pouze rozlišení 10 bitů.

2.2.4 Algoritmus pro rozeznání tepové frekvence

Pro kalkulaci tepové frekvence je třeba několika kroků. Nejprve je z dat odstraněna DC složka, poté jsou data filtrována a až poté je možné přejít ke kalkulaci.



Obrázek 2.6: Princip pulzní oxymetrie [15]

■ Odstranění DC

K odstranění DC složky jsem použil jednoduchý algoritmus[12], který využívá odebraných vzorků, předešlé odfiltrované hodnoty a filtračního koeficientu $\alpha = 0.95$. Vzorec pro vypočtení aktuální hodnoty je

$$y[k] = x[k] - x[k - 1] + \alpha \cdot y[k - 1] \quad (2.9)$$

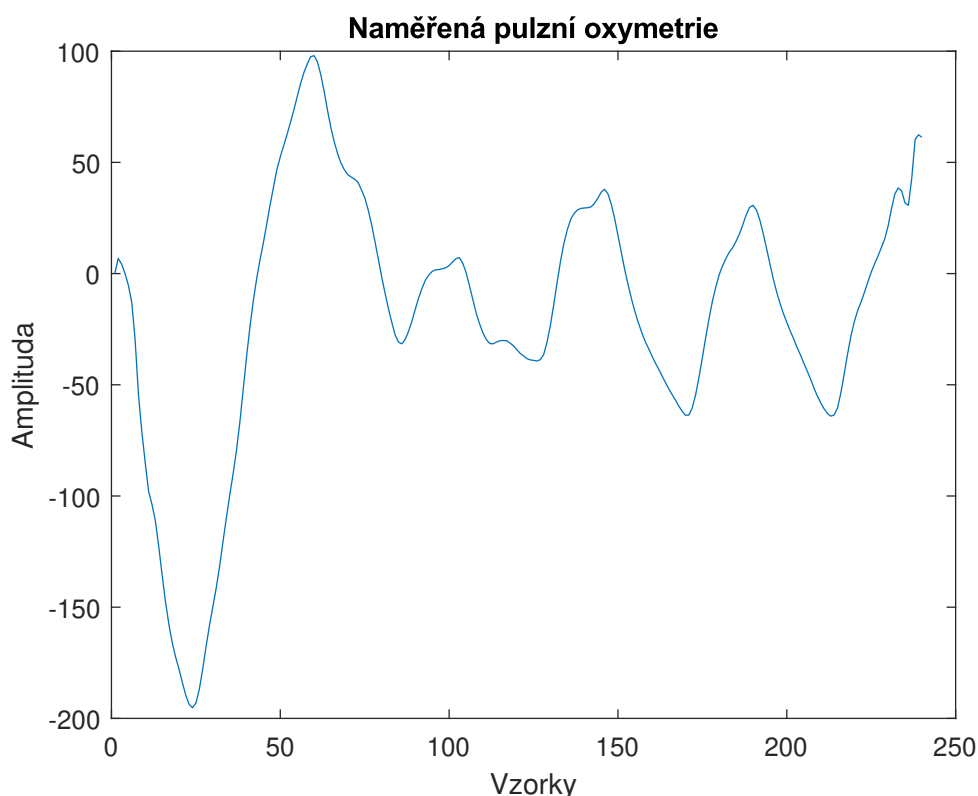
Signál, který má odstraněnou DC složku lze vidět na grafu 2.7.

■ Filtrování

Pro odfiltrování rušivých pulzních složek jsem použil podobně jako v případě akcelerometru středové průměrování v okně. Délku okna jsem volil 31, k čemu jsem došel jakožto optimální hodnotu z měřených experimentů. Pro průměrování tedy používáme vzorec

$$y[k] = \frac{\sum_{i=k-15}^{k+15} x_i}{30} \quad (2.10)$$

Přičemž okrajové vzorky počítáme tak, že odpovídající vzorky na stranách mimo pole se berou jako nulové.



Obrázek 2.7: Signál z oxymetru s odfiltrovanou DC složkou

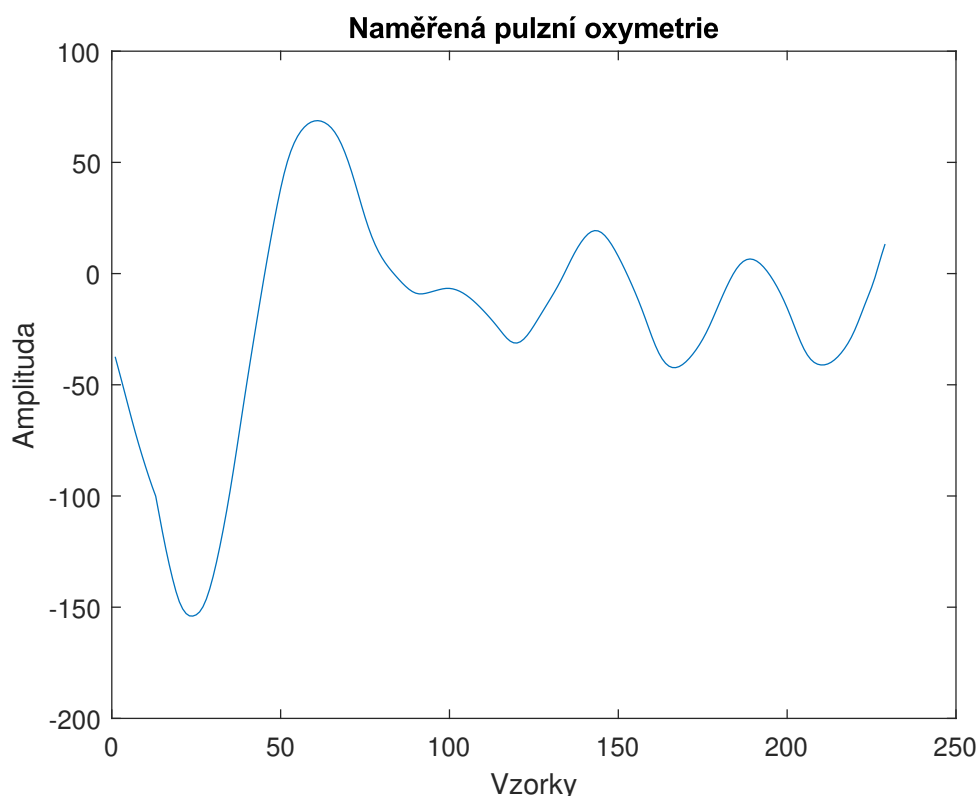
■ Nález pulzů a kalkulace tepové frekvence

Po odfiltrování získáme signál, který je v grafu 2.8. Z něj je vidět, že jednotlivé vrcholy jsou odfiltrovány od zubů a je možné pokračovat v další fázi, kterou je hledání maxim. Maxima jsou hledána vždy v 5ti vteřinových úsecích, aby bylo možné tepovou frekvenci průměrovat a obdržet tak přesnější hodnotu. K nalezeným maximům se zároveň hledá i globální minimum, které je určeno intervalem mezi dvěma nalezenými vrcholy. Aby bylo maximum uznáno jako platné pro kalkulaci srdečního tepu, je nejprve porovnána vzdálenost od předešlého platného vrcholu. Vzdálenost je ohraničena minimální a maximální měřitelnou frekvencí, tedy 30 a 220 tepů za vteřinu. Zároveň musí platit, že výška vrcholu, tedy vzdálenost vrcholu od nalezeného minima, které se nachází mezi nynějším vzorkem a vzorkem předešlým, je alespoň 40. Tímto získáme ve většině případů pouze platné srdeční pulzy.

Následuje už samotná kalkulace frekvence. Postupujeme tak, že vezmeme pár vrcholů, odečteme jejich vzdálenost a pomocí vzorce

$$bpm = \frac{60 \cdot 50}{x} \quad (2.11)$$

Kde x je právě vzdálenost dvou vrcholů, počítána ve vzorcích, určíme okamžitou tepovou frekvenci. Toto provedeme pro každý sousední pár vrcholů a hodnoty sečteme a vydělíme počtem párů. Tím obdržíme průměrnou srdeční frekvenci.



Obrázek 2.8: Filtrovaný signál z oxymetru

2.3 Paměť EEPROM

Pro lokální uchování dat v případě nemožnosti synchronizovat se s mobilním telefonem jsem zvolil paměť EEPROM od firmy Microchip[18], konkrétně model 24LC08BT-I/OT v pouzdře SOT23-5. Jedná se o 8-kbit sériovou paměť, rozdělenou do čtyř bloků o 256 bytech, komunikující na rozhraní I^2C . Má velmi nízkou spotřebu, v případě cyklu čtení/zápisu se jedná maximálně o 1 mA, v klidovém stavu jen 1 μA . Paměť je napájena ze 3,3 V a cyklus zápisu trvá maximálně 5 ms.

Pro výběr bloku je nutné poupravit I^2C adresu. Tato adresa se vždy skládá z kontrolní počáteční sekvence 1010, poté následuje jeden bit x - "don't care" a na závěr dva bity pro výběr bloku. Na paměti je možné uchovat až 168 datových vzorků, v mně implementovaném případě se na paměť ukládají data každou hodinu. Lze tedy uchovat data až jeden týden stará. Data se ukládají do paměti pouze, pokud není dostupné připojení s telefonem. Jakmile je spojení obnoveno, jsou data z paměti přečtena a zaslána do telefonu přes bluetooth Modul.

■ Zápis, čtení a struktura

Struktura EEPROM je následující : v prvních 4 bytech je uchována informace o čtecím a zapisovacím indexu a od 16 byte už následují data. Každý datový vzorek je složen z následujících hodnot: měsíc, den, hodina, počet kroků a tepová frekvence. Až na počet kroků, který má délku 2 byte, je zbytek o délce jednoho byte. Celkově má datový vzorek 6 byte.

Vzorky a jednotlivé byte jsou řazeny sekvenčně a pokud se vzorek již do daného bloku paměti nevejde, je jeho zbytek uložen do bloku následujícího. Index zápisu a čtení odpovídá pořadí byte v paměti, pro určení adresy čtení v bloku je tedy nutné modulovat 256 a pro určení bloku je nutné dělit 256 a zaokrouhlit dolů. V případě zapisování více vzorků nebo čtení během zápisu je nutné zkontrolovat, že paměť je připravena, neboť během zápisu nereaguje na požadavky se sériového rozhraní. Připravenost na data se kontroluje čekáním na acknowledge bit na rozhraní I^2C .

■ 2.4 Rozvod napájení a baterie

Pro přepínání mezi USB napájením a baterií jsem použil integrovaný obvod MP2667GG-0000-P od Monolithic Power Systems [21]. Tento integrovaný obvod dokáže spravovat rozvod napájení a zároveň dobíjet baterii ze vstupu. Komunikace je umožněna pomocí I^2C rozhraní a obvod je v pouzdře 10-QFN. Baterii jsem zvolil Li-Pol o nominální kapacitě 500 mAh a napětí 3.7V, jejíž rozměry jsou 30x35x6 mm. Nastavení integrovaného obvodu je triviální, postačí pouze zvolit maximální možný proud, kterým se smí baterie dobíjet a proud, který může součástka čerpat z USB konektoru. Zároveň je možné vyčítat stavový registr pro získání informace, zda-li dochází k dobíjení baterie atd. Obvod také generuje přerušování, dojde-li například k přehřívání, absenci USB napájení či přepětí na baterii.

K MP2667GG je připojen přes odporový dělič termistor NTC pro monitorování teploty baterie při dobíjení, aby nedošlo k jejímu přehřátí. Použil jsem termistor NTC o odporu $R = 10 \Omega$ při teplotě $t = 25^\circ C$. Jeho zapojení je na obrázku 2.9. Pro výpočet rezistorů R_{T1} a R_{T2} využijeme vzorce [21].

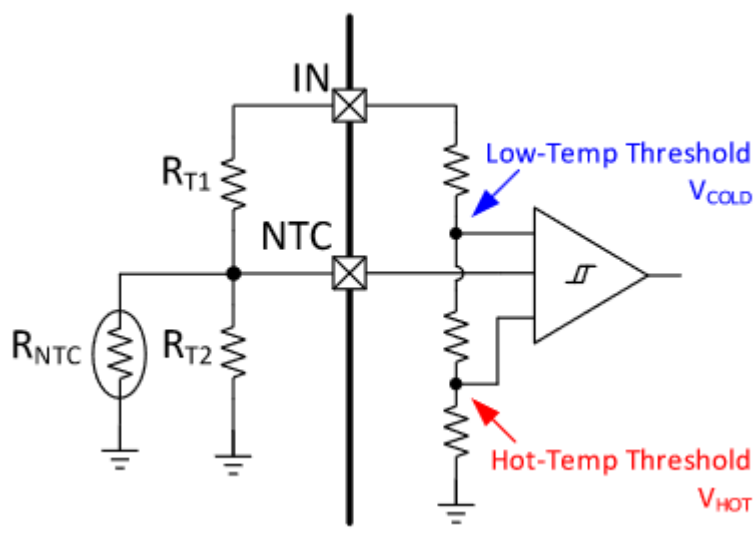
$$\frac{R_{T2} \parallel R_{NTC_COLD}}{R_{T1} + R_{T2} \parallel R_{NTC_COLD}} = 0.66 \quad (2.12)$$

$$\frac{R_{T2} \parallel R_{NTC_HOT}}{R_{T1} + R_{T2} \parallel R_{NTC_HOT}} = 0.33 \quad (2.13)$$

Kde R_{NTC_COLD} je 27280Ω a R_{NTC_HOT} je 4180Ω . Tím pádem obdržíme hodnoty pro jednotlivé rezistory

$$R_{T1} \sim 7500 \Omega \quad (2.14)$$

$$R_{T2} \sim 30000 \Omega \quad (2.15)$$



Obrázek 2.9: Zapojení NTC termistoru [21]

2.5 Struktura mikrokontroléru a hlavního programu

O plynulý chod chytrých hodinek se stará mikrokontrolér STM32F303CCT6 [2] s procesorem řady ARM. Řady F obsahují výkonné čipy, velkou řadu periférií a větší paměť RAM. Ačkoliv nebylo nutné použít tuto řadu co se týče výkonnosti, právě větší RAM byla nutností z důvodu uložení fontů k displeji a řady mnou naprogramovaných knihoven. Zároveň značný počet periférií byl potřebný pro funkčnost všech součástek.

V dalších kapitolách je sepsaný výpis všech periférií, co byly použity a informace o jejich funkčnosti. Výčet jednotlivých pinů lze pozorovat na obrázku A.1. Jednotlivé informace o pinech lze poté vyčíst z tabulky A.1.

2.5.1 Chod programu a jeho struktura

Zvolil jsem procedurální programování se stavovým automatem. Po inicializaci senzorů a součástek (tedy akcelerometru, displeje, externí paměti, oxymetru a dotykového senzoru) program setrvává v hlavní smyčce, kde se nachází rozhodovací strom, který reaguje na vstupní data. To jsou především přerušení, ať už vnitřní nebo vnější. Použil jsem knihovny HAL pro usnadnění přístupu k nastavení a pro sběr dat k řadě funkcí procesoru až na výjimky, kde jsem použil LL knihovny pro zrychlení.

2.6 Hodiny

Pro chod reálného času je použit externí krystal ABS06 od firmy ABRACON[3]. Jeho frekvence je 32,768 kHz, z čehož lze získat čas, kdy jeden tik představuje jednu sekundu.

Jméno	Funkce
VCC	Napájení procesoru
GND	Zem procesoru
SWCLK	Hodinový signál
SWDIO	Vstup/Výstup pro flashování
SWO	Debugovací výstup

Tabulka 2.3: Debug rozhraní

Blok RTC procesoru se stará o počítání kalendářního času. Umožňuje nastavení alarmu, který jsem použil v hodinkách pro refresh displeje a pro synchronizaci naměřených dat. Alarm je nastaven na každou vteřinu a kontroluje se, zda-li je nutné obnovit čas na displeji, který zobrazuje měsíce, dny, hodiny a minuty. Alarm zároveň slouží jako časovač pro Bluetooth komunikaci a odesílání zpráv.

Co se týče nastavení alarmu, nejdříve jsem pracoval s tří-sekundovým intervalem, kde všechny ostatní pole byly zamaskovány, tedy docházelo k alarmu každé tři vteřiny, nehladě na dny, hodiny a měsíce. Jakmile je alarm spuštěn, nejprve se nastavil další alarm o 3 sekundy později a poté se postupovalo dále.

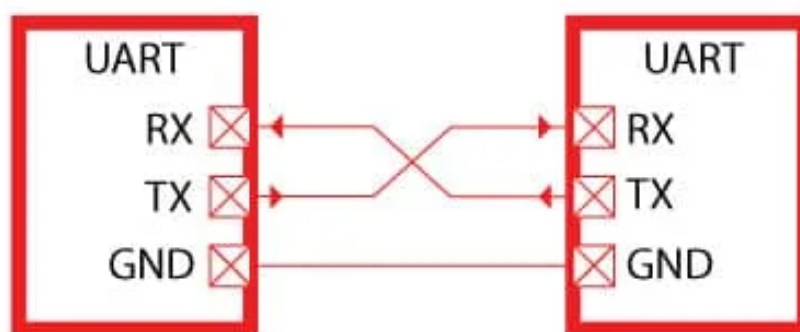
Tento přístup však nebyl korektní, neboť občas nedocházelo ke spuštění alarmu. Ačkoliv jsem nastavoval nejvyšší prioritu přerušení pro alarm, přerušení se nespustilo. Přešel jsem tedy k tomu, že jsem maskoval i vteřiny v alarmu. To znamená, že alarm se spustí každou vteřinu a není nutné alarmy do budoucna nastavovat. Pokud tedy jedno přerušení vyjimečně vynechá, program pokračuje ve své činnosti v dalším příchozím přerušení.

Zpracování instrukcí, které se mají vykonat během každého alarmu jsem nevykonával v metodě přerušení, nýbrž v metodě `HAL_RTC_ExtEventCallback()`, která je zavolána, jakmile se HAL o přerušení postará.

2.7 Debug a flashování programů

Pro nahrávání programů a jejich debugování jsem zvolil rozhraní SWD s asynchronním výstupem. Toto rozhraní má 5 vodičů a jejich funkce jsou popsány v tabulce 2.3. Umožňuje nejen nahrávat programy, ale zároveň je i debugovat. Debugoval jsem zpravidla v STM32CUBEIDE a jakožto debugger a programátor jsem použil zabudovaný ST-Link v NUCLEO kitu.

V Programu jsem použil řadu debugovacích módů. Těmi jsou konkrétně debug pro EEPROM, akcelerometr, napájení, oxymetr, RTC hodiny a Bluetooth. Každý debug vypisuje stavové zprávy o své činnosti, které se poté posílají přes SWO vodič do NUCLEA a přes něj do počítače, kde je možné tyto zprávy vyčítat v terminálu. Vypnutí jednotlivých debugů se nastavuje pomocí True-False v hlavičce hlavní funkce `main.h`.



Obrázek 2.10: Princip zapojení UART [14]

2.8 Komunikace s externími zařízeními

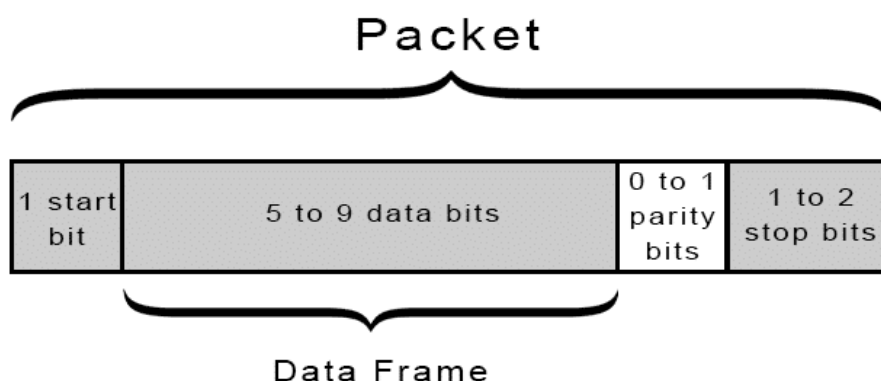
2.8.1 UART

UART (Univerzální Asynchronní Vysílač Přijímač)[9] je sériová linka pro datový přenos. Je vždy mezi dvěma zařízeními. Jsou dva vodiče, Rx a Tx (příjem a vysílání), které jsou zapojeny u zařízení křížem, viz obrázek 2.10.

Jelikož zde chybí hodiny pro synchronizaci signálů, musí obě zařízení znát přenosovou rychlost, která se uvádí v bitech za sekundu (baud). Linka je v klidovém stavu ve vysoké úrovni a komunikace začíná stáhnutím linky dolů, tzv. start bit. Poté následuje datový soubor o šířce 5 až 9 bitů. Za ním je možnost přidat paritní bit, jednoduchá kontrola pro ověření správnosti přenosu, která však není stoprocentní v zarušeném prostředí. Konec paketu je 1 až 2 stop bity, které linku stáhnou na vysokou úroveň. Celý paket lze vidět na obrázku 2.11.

U hodinek jsem použil dvě periferie UART. Jedna periferie o rychlosti 9600 baud komunikuje s Bluetooth modulem, kde dochází k výměně zpráv v rozhraní hodinky - aplikace. Druhá periferie o rychlosti 115200 baud je vhodná pro uživatele, kdy je možné se externě připojit k procesoru pomocí vyvedené kolíkové lišty a vyčítat stavové zprávy.

U UART jsem neposílal zprávy v blokovacím režimu, aby ostatní funkce, které je nutné řešit okamžitě, nečekali, až se zprávy odešlou, či přijmou. Pro odesílání jsem tedy použil mód přerušení, a pro příjem mód DMA, tedy přímý přístup do paměti. Procesor se v tomto případě nemusí starat o přicházející data a teprve až přijdou, dostane signál. Použil jsem vlastní vlajku, která je resetována, jakmile dojde k odeslání zprávy v módu přerušení a je nastavena v případě, že byla zpráva odeslána. Pouze pokud je vlajka nastavena, je možné



Obrázek 2.11: Datový paket UARTu [9]

posílat další data.

■ I^2C

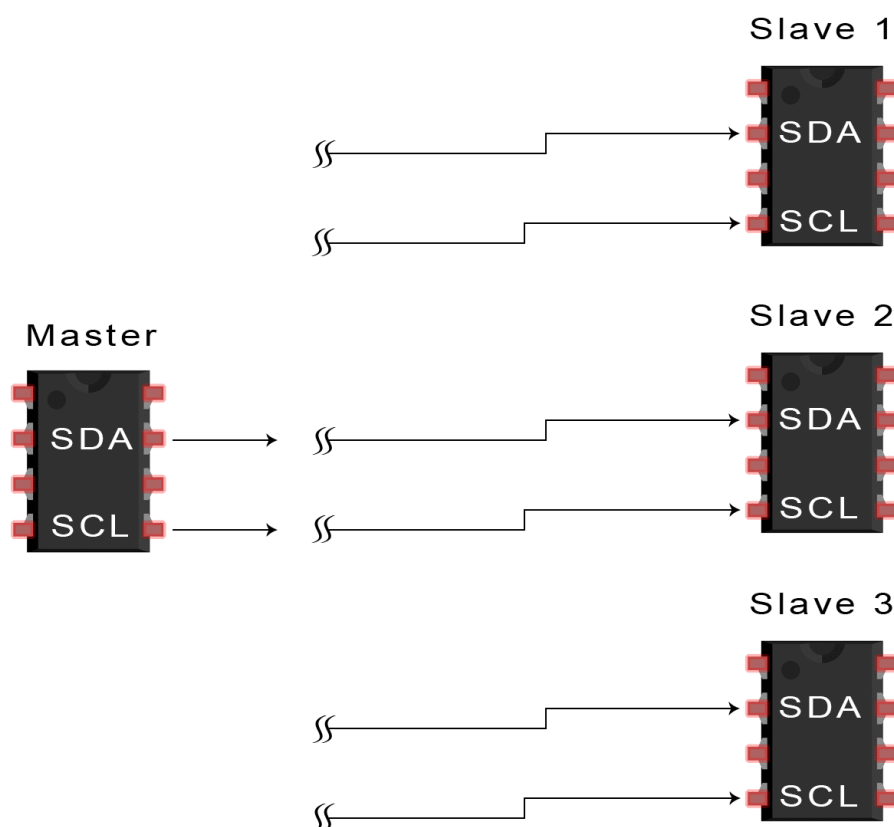
Narozdíl od UART, I^2C [7] umožňuje připojení daleko více zařízení. Má dva vodiče, jeden pro synchronizační hodiny a druhý pro datový přenos. Jedná se o Master - Slave komunikaci, kde Master vždy započíná komunikaci buď s náběžnou, nebo sestupnou hranou hodinového signálu a Slave odpovídá, je-li zvolen. Zvolení je umožněno pomocí adresy, která musí být jedinečná, jinak dochází ke kolizi dvou Slave a komunikace není možná. Master je většinou jen jeden, ale může jich být více, poté je však nutné hlídat datovou linku, tedy pokud je linka na nízké napětové úrovni, další Master nemůže zahájit komunikaci. Je-li Slave zvolen, stáhne datovou linku na nízkou úroveň, čímž pošle potvrzovací signál. Následují datové pakety a po každém z nich opět potvrzovací signál. Komunikace se Slave je ukončena pomocí stop bitu. Struktura obecně je na obrázku 2.12 a datový paket viz 2.13.

V případě hodinek jsem byl nucen použít dvě I^2C sběrnice, protože se překrývaly adresy oxymetru a paměti EEPROM. Paměť totiž reaguje a odpovídá na jakoukoliv adresu počínající sekvencí 1010, kterou zrovna oxymetr obsahuje, tudíž nemohou sdílet jednu sběrnici.

Na první sběrnici o rychlosti 400 kHz je napojen akcelerometr, dotykový integrovaný obvod displeje, oxymetr a dobíječ baterie, na druhé o rychlosti 100 kHz se nachází pouze EEPROM.

■ 2.8.2 SPI

SPI, neboli Sériové Paralelní Rozhraní, je v dnešní době standard pro synchronní sériovou komunikaci, ideální pro vestavné systémy [8]. Má čtyři vodiče - synchronizační hodiny, MOSI, MISO a CS. Jako u I^2C se jedná o Master - Slave strukturu, ale umožňuje komunikaci i pro Slave. MOSI vodič tedy odpovídá výstupu Master uzlu a vstupu Slave uzlu. MISO je naopak, tedy vstup Masteru a výstup Slave. CS, Chip Select, slouží pro výběr určitého



Obrázek 2.12: Struktura I^2C rozhraní [7]

Slave. Struktura je vidět na obrázku 2.14.

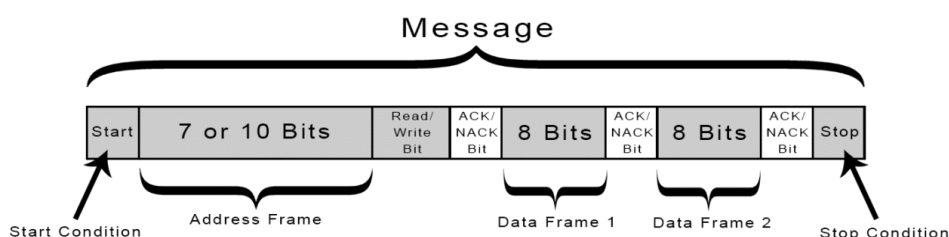
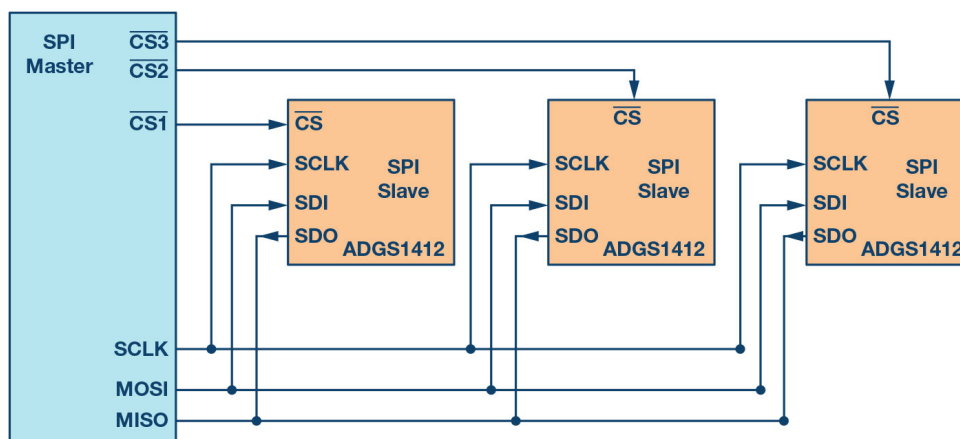
Ve své práci jsem použil SPI pro komunikaci o rychlosti 16 Mbit/s aby bylo možné docílit co nejrychlejší komunikace a tím pádem rychlejší obnově displeje. Nejprve jsem programoval rozhraní s knihovnamy HAL, ale protože HAL knihovny jsou obecně pomalejší a tomto případě byla rychlost nejdůležitější, použil jsem knihovny LL. Pro posílání dat jsem používal vlajky `SPI_TXE`, která je nastavena, pokud je možné odeslat další data. To v kombinaci s přerušením SPI zvýšilo dostatečně rychlost.

2.9 Bluetooth komunikace s Android

O Bluetooth komunikaci se stará modul HC-05, který umožňuje komunikaci s procesorem jak přes UART, tak přes SPI. Ve své práci jsem použil právě UART s rychlostí 9600 baud.

2.9.1 Modul HC-05

Zástavbový modul HC-05 je programovatelný obvod se zabudovanou anténou o velikosti 27x12,7 mm. Tento modul jsem zvolil proto, že byl v bastlířské

Obrázek 2.13: Datový paket I^2C [7]

Obrázek 2.14: Multi-Slave SPI struktura [8]

komunitě velmi používaný a práce s ním nevyžadovala složité programování co se týče nastavení a podpůrných obvodů. Tato volba ovšem nebyla tou nejchytřejší, neboť modul má v dokumentaci uvedené funkce několika pinů, které jsem ovšem při zapájení na plošný spoj nepozoroval. Byl jsem tedy nucen tyto funkce nahradit softwarově.

2.9.2 Struktura komunikace

HC-05[13] funguje v režimu Slave, tedy není schopné se připojit samo od sebe ke spárovanému zařízení. Spojení tedy iniciuje telefon, který v aplikaci spustí vyhledávání. Nalezne-li telefon zařízení s názvem "HC-05", pokusí se zahájit spojení pomocí unikátního řetězce, který pro softwarové desky často bývá stejný. Tento řetězec je "00001101-0000-1000-8000-00805F9B34FB" a představuje 128-bitový identifikátor.

2.9.3 Princip komunikace

Po úspěšném připojení pošle Aplikace paket s reálným časem desce pro synchronizaci. Tento paket má délku 8 bitů, kde první bit je identifikátor zprávy, v tomto případě "0" a následuje 7 bitů, které jsou jednotlivě přiděleny pro rok, měsíc, den, den v týdnu, hodinu, minutu a vteřinu.

Stav připojení měl signalizovat modul HC-05 vysokou napěťovou úrovní na jednom z výstupních pinů. Bohužel, úroveň na vybraném pinu se neměnila

nehledě na stav připojení. Hodinky tedy poznají, že jsou připojeny tak, že na některou ze zpráv obdrží acknowledge bit, který posílá aplikace.

V následujících kapitolách popíší typy odesílaných zpráv a jak jsou řešeny v programu.

■ 2.9.4 Zpráva pro časové nastavení

Zpráva, kterou obdrží hodinky po spojení a která je popsána výše slouží pro nastavení zprávného času. Je zastaven alarm v bloku RTC, jsou přepsány hodnoty v registrech, nastaven čas a poté je alarm obnoven. Chod programu pokračuje neovlivněn.

■ 2.9.5 Synchronizační zpráva

Synchronizační zpráva je základní zprávou s nejnižší prioritou, která je vysílána každé 3 vteřiny, pokud ovšem není nutné vyslat zprávu s vyšší prioritou, poté odeslána není. Obsahuje identifikační byte, tepovou frekvenci (1B) a počet ušlých kroků(2B), celkově o délce 4B.

Informace jsou uchovány ve struktuře a je nastavena vlajka pro synchronizační zprávu. V hlavní smyčce programu se poté struktura překopíruje do pole sloužícího pro přenos data přes UART a pomocí HAL v módu přerušování odeslána do Bluetooth modulu.

Pokud aplikace zprávu obdrží, pošle zpátky acknowledge byte. Je-li tento bit přijat hodinkami v synchronizačním okně, které má délku 3 vteřiny, hodinky nastaví vlajku, že je zařízení připojeno. Pokud tento byte obdržen není, vlajka je resetována

■ 2.9.6 Zpráva sloužící pro uložení do databáze

Každou novou hodinu je posílána zpráva, která má být uložena do databáze Android aplikace. Tato zpráva má délku 9 byte a její struktura je v tabulce 2.4. Zároveň má vyšší prioritu než zpráva synchronizační, tedy tato zpráva se odešle místo ní.

Pokud v synchronizačním okně nedojde k obdržení acknowledge, hodinky informaci uloží do EEPROM paměti. Toto se řeší nastavením 2 vlajek, jedna pro odeslání databázové zprávy a jedna pro přijetí acknowledge. Pokud jsou obě nastaveny v dalším synchronizačním okamžiku, data uložena nejsou a došlo k úspěšné transakci. Pokud je nastavena vlajka pro odeslání, ale není nastavena vlajka pro přijetí, jsou informace, které měly být odeslány, uloženy do paměti EEPROM. Vlajky jsou poté resetovány.

■ 2.9.7 Zpráva sloužící pro odeslání dat z EEPROM

Jestliže je nastavena vlajka připojení k zařízení, v hlavní smyčce programu dochází ke kontrole počtu vzorků v paměti EEPROM. Pokud je tento počet alespoň jedna, hodinky se snaží uložená data v EEPROM poslat aplikaci. Struktura zprávy je stejná jako v případě databázové zprávy, viz 2.4, ale má

Typ informace	délka [Byte]
Identifikátor	1
Počet vzorků	1
Rok	1
Měsíc	1
Den	1
Hodina	1
Počet kroků za hodinu	2
Průměrný tep	1

Tabulka 2.4: Zpráva pro uložení do databáze Android

jiný identifikátor.

Vzorky jsou čteny z paměti po jednom a odesílány aplikaci každou vteřinu. Pokud je obdrženo acknowledge bit, je posunut čtecí index v paměti, je zapsán index posledního úspěšného odeslaného vzorku a pokračuje se opět kontrolou, je-li ještě nějaký vzorek v paměti. Synchronizace dat z paměti s aplikací trvá zpravidla delší dobu, protože Android aplikace často neobdrží korektně zprávu.

Nedojde-li k obdržení acknowledge, obnoví se zasílání synchronizačních zpráv. Až dojde k přijetí acknowledge na synchronizační zprávu, je opět započat přenos vzorků z paměti EEPROM, kde se vysílá od vzorku, jehož index následuje po posledním úspěšně odeslaném indexu.

Kapitola 3

Uživatelské rozhraní

3.1 Displej

Pro zobrazení informací na hodinkách jsem použil kruhový grafický displej firmy Waveshare[24] s úhlopříčkou 1,28 palce a rozlišením 240x240 pixelů (pixely, jež se nenachází ve vymezeném kruhu se nepoužívají). Barevné spektrum je celé 10-bitové RGB. Display je zároveň dotykový díky integrovanému obvodu v displeji, což umožňuje rozpoznávání řady gest, od dotyku po přejetí po displeji.

3.1.1 Způsob komunikace

Displej pro obnovu dat používá SPI a pro dotykovou komunikaci I^2C . Celkově má 13 pinů a důležité piny jsou uvedeny v tabulce A.1. Spojení mezi displejem a plošným spojem zprostředkuje konektor 13 pinový konektor JST-GH s roztečí pinů 1.27mm. Vzhledem k tomu, že výrobce dodává konektor s výstupními DuPont konektory, na které není na plošném spoji místo, byl jsem nucen krimpovat vlastní vodiče ke konektorům.

3.1.2 Grafické zpracování

Cílil jsem pro jednoduchý design a přehlednou prezentaci informací. Displej tedy zobrazuje měsíce, dny, hodiny a minuty. Zároveň je zde uveden i počet kroků a tepová frekvence. Celkový design lze vidět na obrázku A.24. Každých pět vteřin je displej obnovován s novými daty. Druhá strana, na kterou se dostaneme přejetím dolů, zobrazuje status připojení k aplikaci, počet vzorků v EEPROM paměti a poté nastavení podsvícení displeje, které se volí dotykem na posuvnou lištu. Viz A.25. Na domovskou obrazovku se vrátíme přejetím nahoru.

3.2 Android aplikace

Pro design aplikace jsem použil jazyk Kotlin v Android Studiu. Vzhledem k rozlehlosti a komplexitě plošného spoje, pouzdra a programování hodinek jsem se rozhodl pro jednoduchý, ale efektivní design. Aplikace má dohromady

3 stránky.

■ 3.2.1 Cyklus aplikace

Aplikace je nastavena v režimu Master-Slave. Jediná hlavní aktivita přepíná obsah a různým elementům posílá informace. Grafy a další informační obsahy jsou uzavřeny do fragmentů, které se zobrazují na odlišných stránkách. Mezi stránkami se přepína pomocí dolní navigační lišty. V hlavní aktivitě dochází také k započítání a udržení připojení k aplikaci, spravování databáze atd. Pokud tedy uživatel aplikaci zavře, spojení je ukončeno.

Předávání dat je uskutečněno pomocí rozhraní ItemViewModel, které umožňuje přenos informací mezi jednotlivými částmi aplikace, které jsou jinak oddělené a není možné s nimi přímo komunikovat. Je-li některá z hodnot rozhraní změněna, jsou notifikovány i části, které tato hodnota ovlivňuje. To je například změnění data v kalendáři. Jestliže je datum změněno, změní se i počet kroků v dalších částích.

Pro uchování zpráv z hodinek jsem použil Room Database, která pracuje na principu SQLite. Vytvořil jsem celkem dvě databáze, jednu pro počet kroků a jednu pro tepovou frekvenci. Každý uložený vzorek obsahuje informaci o čase a o naměřené hodnotě.

■ 3.2.2 Domovská stránka

Domovská stránka má za úkol zobrazit měřené údaje, připojit se k hodinkám a nastavit si datum. Její zpracování můžeme vidět na obrázku A.27. Obsahuje textové pole, které reprezentuje momentální vybrané datum, tlačítko pro výběr datumu a tlačítko, které slouží k připojení aplikaci a hodinek přes Bluetooth.

Pod těmito elementy se nachází 2 fragmenty. První fragment obsahuje informaci o počtu ušlých kroků ve vybraném dni a také pokrok, který je vizualizován lištou, jež je postupně vyplňována. Denní cíl je nastaven na 10000 kroků. Druhý fragment obsahuje tepovou frekvenci a také lištu, která vizualizuje měřenou hodnotu a zároveň určuje stupeň zatížení pomocí barevného gradientu. Pod lištou se ještě nachází textové pole, které definuje aktivitu na základě tepové frekvence. Tato aktivita je buď : Klidový stav, průměrná zátěž, silná zátěž a extrémní zátěž.

■ 3.2.3 Stránka s grafy

Pro vytvoření grafů jsem použil MPAndroidCharts od Philippa Jahody. Celkem dva grafy pro vizualizaci, jeden pro počet kroků a jeden pro tepovou frekvenci, se nacházejí samostatně ve své stránce. Stránku lze vidět na obrázku A.28 a obsahuje výběr data pro inspekci předešlých vzorků a právě tyto dva grafy. Graf je obecně koncipován jako histogram obsahující 24 vzorků, jeden pro každou hodinu. Pokud v databázi chybí nějaký vzorek z daného času, je nahrazen nulou. Grafy je možné přibližovat a oddalovat pomocí dvou prstů.

Každý vzorek představuje měřenou hodnotu právě v dané hodnotě, tedy u kroků se vizualizuje pouze ušlý počet za danou hodnotu, nedochází ke sčítání. U tepové frekvence je poté uvedena průměrná hodnota v hodině. Vizualizace ušlých kroků je na obrázku A.29.

■ 3.2.4 Informační stránka

Tato stránka slouží pouze k uvedení informace o autorovi, použité knihovny pro grafy a pro odkaz na GitHub, kde je uložen celkový projekt včetně plošného spoje atd. Viz obrázek A.30

Kapitola 4

Desky plošných spojů

4.1 Deska hodinek

4.1.1 Obecné směrnice

Aby bylo možné opatřit hodinky všemi požadovanými funkcemi, bylo nutné použít čtyřvrstvou desku, tedy dvě signálové vrstvy, vrchní a spodní, a dvě vnitřní vrstvy pro napájení, jedna pro zem a druhá pro +3,3V.

Kladen byl i důraz na rozměry plošného spoje, kde horním stropem pro rozměry hodinek bylo 50x50x30 mm.

Rozložení bylo také limitováno displejem, který bylo nutné přišroubovat ke plošnému spoji pomocí 2 šroubů typu M2 umístěných křížem. Napájení jsem rozváděl pokud možno plochou nebo tlustými vodiči. Zbylé spoje byly zpravidla vedeny šířkou 200mm. Nepoužíval jsem žádné utopené díry a prokovy. Prokovy jsou o průměru 0,4mm a 0,6 mm ve většině případů. Výrobní limitace, jako minimální izolační vzdálenost, jsem převzal od výrobce JLCPCB.

U pasivních součástek, jako jsou kondenzátory a rezistory jsem volil co největší pouzdra, aby bylo osazování snadné. U kondenzátorů to bylo pouzdro 1206, u odporů potom pouzdro 0805. Ve vyjíměčných případech jsem ovšem byl nucen použít i menší pouzdra, konkrétně 0402, které se nachází u oscilátoru a u vstupu USB.

4.1.2 Prototypy

Navrhl jsem celkově dva prototypy, které se příliš nelišily, ale druhý řešil kritické nedostatky prvního. Výstup prototypů je shrnut v závěru kapitoly. První deska byla navrhována se všemi funkcemi. Vzhledem ke kruhovému displeji jsem se rozhodl pro kruhovou desku. Výběr funkcí pinů a umístění součástek jsem volil tak, aby pokud možno kritické vodiče jako I^2C byly v rovinně součástek, které je vyžadují. Zároveň jsem piny volil tak, aby nebylo nutné složitě vést cesty přes plošný spoj. Druhý prototyp řešil nesprávný design externího oscilátoru, protože u prvního prototypu jsem použil krystalový integrovaný oscilátor, který však měl špatné zapojení a nemohl fungovat, pokud nebylo přítomno bateriové napájení. Byl tedy nahrazen krystalem,

který už byl napájen i v případě, že přívod energie z baterie a USB není možný. Oba tyto designy jsou probrány v sekcích této kapitoly.

■ 4.1.3 Napájení

■ USB

Prvním zdrojem napájení je USB Mini-B konektor dimenzovaný na 5V, kde jsem uvažoval napájení z počítače. Hned za napájecími piny se nachází TVS dioda pro protekci vůči elektrostatickým výbojům. Volil jsem model ESD401DPYR[22] od Texas Instruments, protože nabízí ± 24 kV odolnost vůči kontaktu a ± 30 kV odolnost vůči výboji mezi vzduchovou mezerou. Má zároveň minimální odběr proudu a kapacitu 0.22 pF. Následuje poté PI můstek pro odfiltrování rušivých složek ze zdroje. Volil jsem kondenzátory o kapacitě 2.2 μF a tlumivku 22 nH. Schéma lze vidět na obrázku A.4.

■ Baterie

Druhým zdrojem je LiPol baterie o nominální kapacitě 500 mAh a napětí 3.7V. Je zakončena JST-PH konektorem, tudíž je na desce receptabilní konektor stejného typu, který je odfiltrován 10 μF tantalovým kondenzátorem. Dále se na plošné desce nachází knoflíková baterie CR1025 s výstupním napětím 3V, která slouží pro napájení záložních RTC registrů, RTC bloku a také pro napájení externího LSE krystalu.

■ 4.1.4 Přepínání mezi zdroji

Jak bylo uvedeno v první části, o přepínání mezi USB a baterií se stará MP2667G[21]. Na výstupu součástky, která je vstupem regulátoru, je baterie, pokud ovšem není přítomno napájení z USB konektoru, poté se přepne právě na něj. Výstup je ještě dodatečně filtrován 2,2 μF kondenzátorem.

■ 4.1.5 Regulátor

Všechny součástky až na jednu požadují napájení 3,3V, tedy volil jsem lineární regulátor TPS763[23] s nízkým spádovým napětím 150 mA v pouzdře SOT23-5. Vstup je dodatečně filtrován 2,2 μF kondenzátorem a výstup 4,7 μF tantalovým kondenzátorem. Tento výstup je následně rozveden do třetí vrstvy desky, kde je následně vyveden k napájecím pinům součástek.

■ 4.1.6 Procesor

Procesor jsem umístil na levou část desky, abych mohl maximálně využít prostor a jednotlivé piny. Každou dvojici napájecích párů jsem filtroval 100 nF kondenzátory a k tomu jsem navíc přidal 10 μF kondenzátor pro větší stabilitu. Díky tomu, že procesor nemá termální plošku pro odvod tepla, bylo snadnější rozvést jednotlivé cesty pomocí prokůvů pod pouzdem procesoru.

■ 4.1.7 Krystal

Druhý prototyp nahrazuje špatně navržený krystalový oscilátor. V druhé verzi jsem použil samotný krystal. K oběma pinům krystalu jsou přivedeny kapacitory v pouzdře 0402 a kapacitě 15 pf, které jsem určil ze vztahu [20]

$$C_L = \frac{C_1 \cdot C_2}{C_1 + C_2} + C_S \quad (4.1)$$

Kde $C_L = 12,5 \text{ pF}$ je zátěžová kapacita krystalu, C_1, C_2 jsou kondenzátory a C_S je rozptylová kapacita, pro níž jsem volil $C_S = 5 \text{ pF}$. Pod oscilátorem jsem provedl výřez napájecí a zemnicí vrstvy, aby nedocházelo k rušení [6]. Zároveň jsem kolem krystalu vytvořil pomocí spoje a prokovů tzv. Guard Ring, který také slouží pro zlepšení efektivity a funkčnosti krystalu. Design je možné vidět na obrázku A.26

■ 4.1.8 Displej

Jak již bylo zmíněno, na uchycení displeje k desce jsem použil dva otvory pro šroub M2, které jsou umístěny křížem. Neuchytil jsem displej čtyřmi šrouby, protože na desce nebyl dostatek místa a dodatečná stabilita nebyla potřeba. Ze spodní strany jsem umístil matici a šroub přes ní navrtal do distančního sloupku o výšce těla 5mm a délky závitu 8mm.

Vysoká výška displeje nad deskou je způsobena tím, že displej má namontovaný JST-GH 13 pinový konektor, tudíž i na desce musí být. Volil jsem přímý konektor a umístil jsem ho na jih desky, abych většinu pinů mohl vyvést vedle sebe rovnou do procesoru. Piny konektoru jsou dostupné v tabulce A.1.

Problémem byla nutnost vytvořit si osobně nakrimpované dráty o požadované délce, neboť již zhotovené nebyly dostupné. Dráty jsem tedy nakonec krimpoval s použitím speciálních kleští, krimpovacích kontaktů dělaných pro tento typ konektoru a izolovaným jednožilovým vodičem šířky AWG30, tedy licny o průřezu $0,05 \text{ mm}^2$ na délku 3 cm.

■ 4.2 Deska oxymetru

Druhá deska se nachází externě na spodku hodinek. Je dvouvrstvá o rozměru 20x20 mm a obsahuje nejen oxymetr, ale i blokovací kondenzátory a regulátor pro napájení součástky. Jsou zde vyvrtné dvě díry pro 2 šrouby typu M2 pro uchycení k pouzdru. Dále se zde nachází pět děr pro přívod napájení o úrovni 3,3 V, dva vodiče pro rozhraní I^2C , přerušeni a zem.

Oxymetr je schválně umístěn veprostřed desky, aby mohl zbytek být přikryt víkem a tudíž aby nedocházelo k odírání desky o kůži. Oddělil jsem digitální a analogovou zem a spojil je v jednom bodě na spodní straně desky.

Drivery pro LED jsou blokovány $10 \mu F$ a 100 nF kondenzátory v pouzdře 1206. Vstup regulátoru je blokován $1 \mu F$ a výstup $4,7 \mu F$ a 100 nF . Výstup regulátoru je o jmenovitém napětí 1,8V a slouží k napájení oxymetru. LED drivery vyžadují napětí 3,3 V.

Výsledný návrh desky je na obrázcích A.11 a A.12.

4.3 Redukční deska

Redukční deska je adaptér pro DuPont konektory z NUCLEO kitu a pro kolíkovou debugovací lištu, protože vodiče nejde přímo připojit. Deska je jednostranná, obsahuje vidlici pro připojení DuPont konektorů a na spodní straně poté zásuvku pro piny s roztečí 2mm z desky hodinek. Jedná se o triviální desku a můžeme ji vidět na obrázku A.15.

4.4 Výsledný návrh, osazování

Výsledné signálové vrstvy lze vidět na obrázcích A.5, A.6, A.10 a A.9. Vyrobený první prototyp je poté na obrázcích A.8 - horní vrstva a A.7 - spodní vrstva. Rozměry plošného spoje se nakonec ustálili na průměru 44mm, menší už z mého pohledu bylo těžko proveditelné a ruční osazování až nemožné.

Většina součástek byla pájena pomocí páječky, až na součástky jako akcelerometr a rozvaděč, které měli plošky vyvedené pod pouzdrem, a tudíž vyžadovali osazení pomocí horkého vzduchu. Vzhledem k limitované předešlé zkušenosti bylo pájení náročné, zejména procesor a některé součástky v pouzdrech 0402 a 0805. Než bylo vše korektně zapájeno, bylo zapotřebí několik pokusů. Nakonec se vše podařilo a výsledný první zapájený prototyp lze vidět na obrázku A.2 a A.3. Chybou druhého prototypu byl špatný návrh krystalu a rozložení pinů resetovacího tlačítka. Druhý prototyp tyto nedostatky má řešit.

Kapitola 5

Pouzdro

Součástí hodinek je samozřejmě i pouzdro s řemínkem, aby se daly pohodlně nosit na zápěstí. K designu jsem použil Fusion 360 od Autodesk. Koncipoval jsem pouzdro na 4 části, které se spojují dohromady pomocí zaklapávacích svorek. Použil jsem materiál SLA - resin, výsledek je tedy matné, hladké a černé pouzdro. Co se týče rozměrů, hlavní tělo má průměr 50mm a výšku 35mm. Navržené pouzdro se všemi částmi je na obrázku 5.1

5.1 Horní část

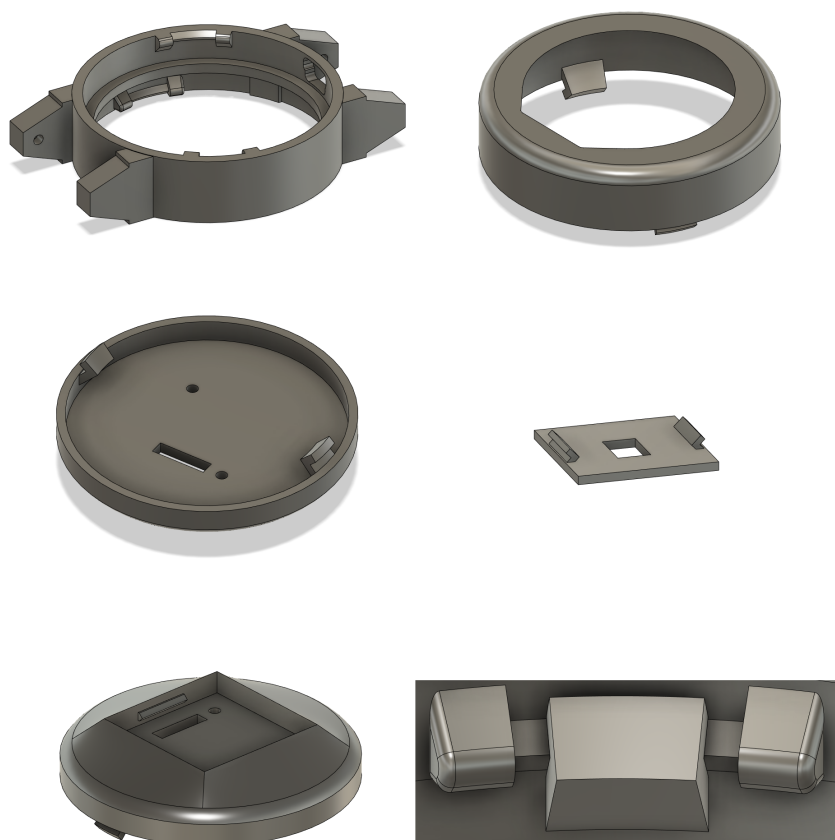
Horní část slouží pro ochranu horní části desky a můžeme ji vidět na obrázku A.16. Vytváří okraj displeje a má celkovou výšku 9 mm. Pro okraj jsem použil profil displeje a tento profil jsem poté oddálil o 0.5 mm. Zaklapávací svorky jsou dvě a mužského typu, tedy připínají se na hlavní středovou část. Vnitřní strana, která se sužuje k okraji displeje jsem vyhladil, aby nebyly ostré přechody.

5.2 Prostřední část

Hlavní část hodinek je právě část prostřední, jejíž design lze pozorovat na obrázku A.17. Vzhledem k tomu, že na pouzdro se již nevešly díry pro šrouby na uchycení, zachycení plošného spoje jsem volil tak, aby vnitřek měl plošinu, na které by ležel. Tato plošina vystupuje z těla 1mm. Plošný spoj je poté horní vrstvou přitlačen tak, aby se nepohyboval.

Nacházejí se zde 2 páry zaklapávacích svorek ženského pohlaví s tím, že okraje jsou tlusté, aby nemohlo dojít k nechtěnému otáčení jednotlivých částí vůči sobě.

Pro uchycení řemínku jsem udělal pár nožek, které vystupují z pouzdra a jsou vzdálené 20,2 mm od sebe. Do nich jsou poté vytvořené díry, aby se pomocí stěžejek mohl řemínek pevně uchytit k hodinkám.



Obrázek 5.1: Navržené části pouzdra

5.3 Spodní část

Spodek hodinek byl udělán užší, aby se do něj vešla baterie. Spodek má zevnitř dva otvory pro šrouby M2 díru ve tvaru obdélníku pro kabely oxymetru. Vnější strana spodní strany je vyhlazena a ústí v obdélníkový otvor, který má hloubku čtyř milimetrů, aby se do něj vešla deska s oxymetrem a zároveň zaklapávací pár. Tímto se nemusí deska odírat o pokožku. Část je možno vidět na obrázcích A.18 a A.19.

5.4 Víko

Poslední částí je víko, kterým se zaklapne spodní část pouzdra. Slouží k tomu, aby koukal pouze oxymetr a zbytek desky, na které je umístěn, je shován. Nachází se zde dva páry zaklapávacích svorek ženského pohlaví. Viz obrázek A.20.

■ 5.5 Zhodnocení a výsledky pouzdra

Ačkoliv jsem volil rozměry těl částí skromně, aby byly dodatečné mezery, pouzdro je naopak těsné. Zaklapnutí vrchní části s prostřední je relativně bezproblémové, ale zaklapnutí spodní části s prostřední vyžaduje odebrání šroubů pro uchycení oxymetru, neboť baterie je příliš tlustá a vytlačuje části od sebe.

Uchycení řemínku je snadné a drží pevně, zároveň na zápěstí hodinky drží a nošení není nepříjemné. Nevýhodou pouzdra je samozřejmě jeho výška, která bohužel snížit nešla.

Výsledné zaklapnuté pouzdro je na obrázku A.21,A.22 a A.23.

Kapitola 6

Závěr

Co se týče programovací části práce, akcelerometr i oxymetr fungují přesně a korektně zpracovávají hodnoty. U mikrokontroléru by bylo možná vhodnější použít vícevláknové programování, ale mé řešení pomocí rozhodovacího stromu na základě přichozích přerušení a nastavovaných vlajek je také efektivní.

Navržené plošné spoje, alespoň co se týče prvního prototypu, splňují funkce, pro které byly navrženy, je možné s nimi pracovat jako se samostatnou jednotkou díky bateriovému napájení. Největší problém u plošných spojů bylo paradoxně jejich doručení. Kvůli dlouhé době čekání jsem nebyl schopen otestovat druhý navržený prototyp a desku s oxymetrem. Do obhajoby by tyto prototypy měly být otestovány.

Pouzdro hodinek se povedlo bez větších výhrad. Pasuje na tělo plošného spoje a spolu s řemínkem je možné hodinky nosit na zápěstí. Pouzdro je však poněkud vyšší, než by bylo vhodné, kvůli potřebnému místu na konektor displeje, tloušťce použité baterie a vysokému pouzdru pro knoflíkovou baterii. Tyto nedostatky by bylo možné řešit vhodnějším výběrem součástek a displeje, jehož konektor by byl řešen například pomocí flexi kabelu.



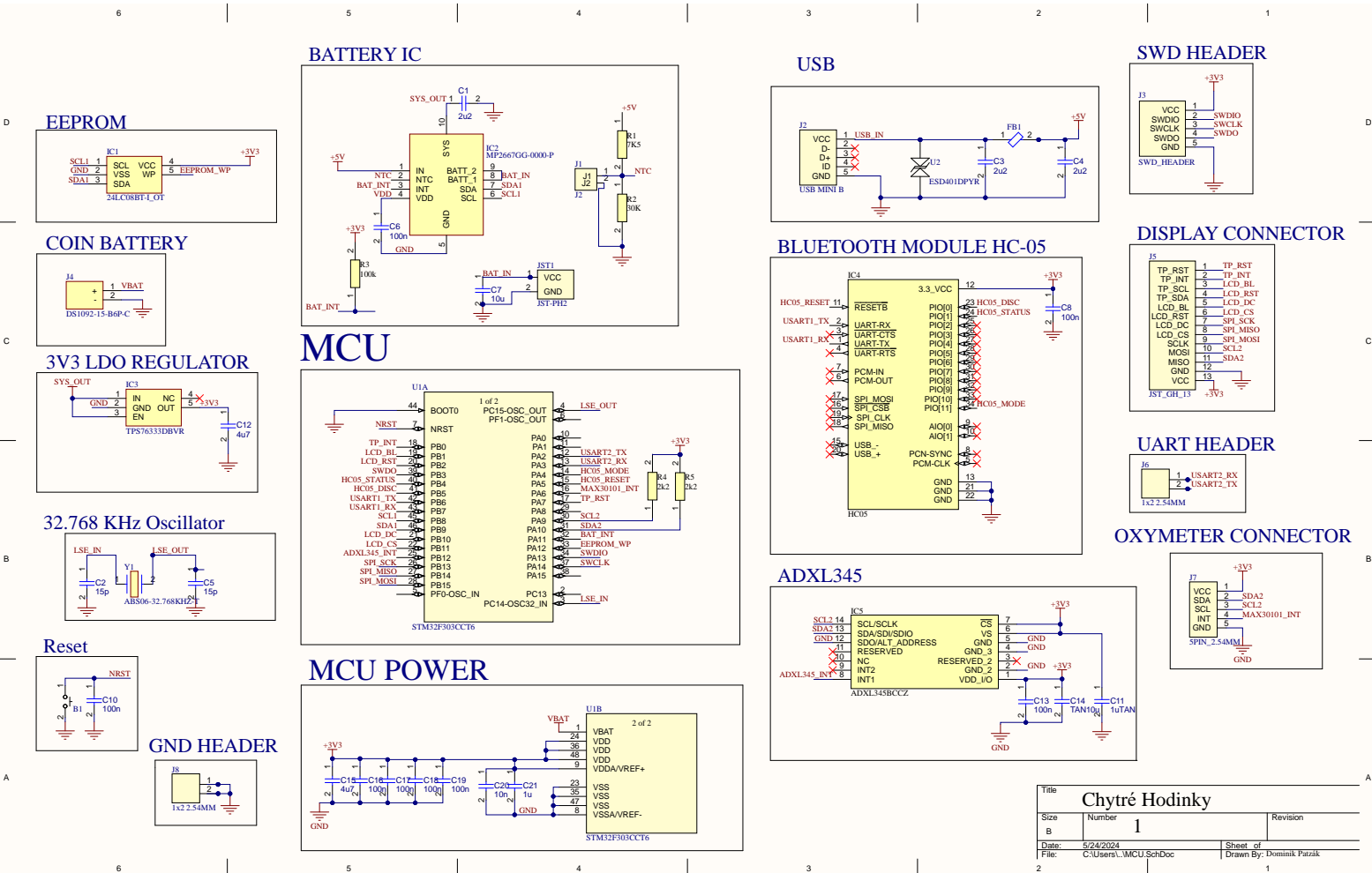
Literatura

- [1] Max30101. <https://www.analog.com/en/products/MAX30101.html>. Online, navštíveno 22/5/2024.
- [2] Stm32f303cct6. <https://www.st.com/en/microcontrollers-microprocessors/stm32f303cc.html>. Online, navštíveno 22/5/2024.
- [3] Abracon. 32.768khz smd crystal. <https://abracon.com/Resonators/ABS06.pdf>, 2020. Online, navštíveno 22/5/2024.
- [4] R. H. Agatha Bradjic. Walk detection and step counting on unconstrained smartphones. *ACM Digital Library*, 2013.
- [5] Analog Devices. *ADXL345*. <https://www.analog.com/en/products/adxl345.html>.
- [6] Atmel. *Best Practices for the PCB Layout of Oscillators*, avr186 edition, September 2016.
- [7] S. Campbell. Basics of i2c communication. <https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol>. Online, navštíveno 22/5/2024.
- [8] S. Campbell. Basics of spi communication. <https://www.circuitbasics.com/basics-of-the-spi-communication-protocol>. Online, navštíveno 22/5/2024.
- [9] S. Campbell. Basics of uart communication. <https://www.circuitbasics.com/basics-uart-communication/>. Online, navštíveno 22/5/2024.
- [10] M. Dadafshar. Accelerometer and gyroscopes sensors: Operation, sensing, and applications. *Maxim Integrated*, 2015.
- [11] A. Devices. Accelerometer and gyroscopes sensors. <https://www.analog.com/en/resources/technical-articles/accelerometer-and-gyroscopes-sensors-operation-sensing-and-applications.html>. Online, navštíveno 22/5/2024.

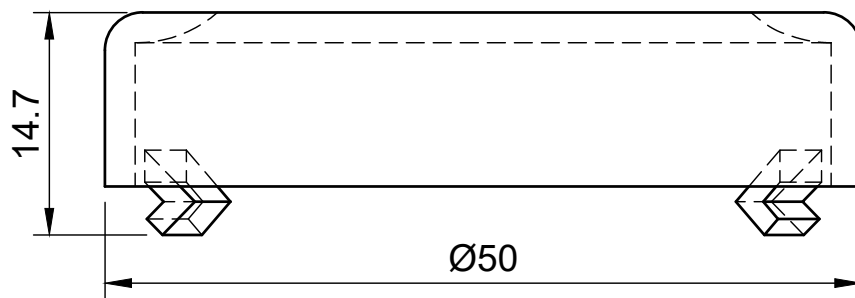
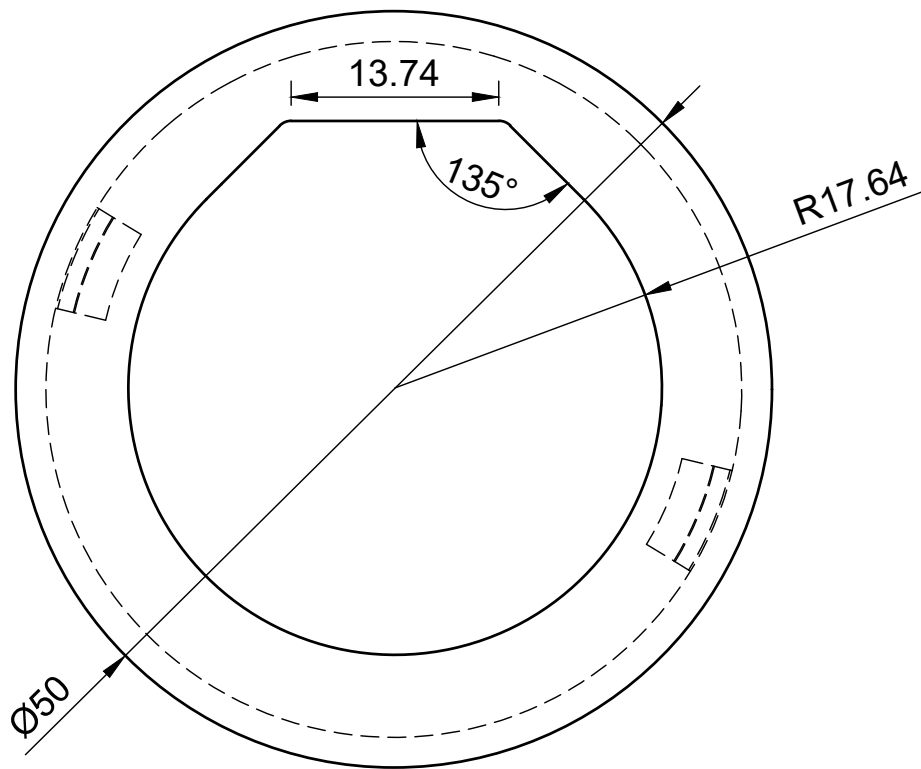
- [12] K. Embedded. Dc signal removal. <https://kleinembedded.com/quick-and-simple-digital-filters-for-smoothing-and-dc-removal/>. Online, navštíveno 22/5/2024.
- [13] ITead Studio. *HC-05 Datasheet*, 2010.
- [14] M. Lab. Esp32 uart communication explained. <https://microcontrollerslab.com/esp32-uart-communication-pins-example/>. Online, navštíveno 22/5/2024.
- [15] Y. N. M. Fujiyoshi. An soi mems 3-axis accelerometer with a zigzag-shaped z-electrode for fully differential detection. *Semantic Scholar*, 2014.
- [16] Maxim Integrated. *MAX30100*. <https://www.analog.com/en/products/max30100.html>.
- [17] Maxim Integrated. *Recommended Configurations and Operating Profiles for MAX30101/MAX30102 EV Kits*, ug6409 edition, March 2018.
- [18] Microchip. *24LC08B*. <https://www.microchip.com/en-us/product/24LC08B>.
- [19] S. Quinn. Photoplethysmography. <https://www.instructables.com/Photoplethysmography-IR-Heart-Rate-Monitor/>. Online, navštíveno 22/5/2024.
- [20] ST Microelectronics. *Guidelines for oscillator design*, an2867 edition, May 2024.
- [21] M. P. Systems. Mp2667. <https://www.monolithicpower.com/en/mp2667.html>. Online, navštíveno 22/5/2024.
- [22] Texas Instruments. *ESD401 1-Channel ESD Protection Diode*, slvse49a edition, July 2017.
- [23] Texas Instruments. *TPS763 Low-Power, 150-mA, Low-Dropout Linear Regulator*, slvs181j edition, September 2019.
- [24] Waveshare. 1.28inch touch lcd. https://www.waveshare.com/wiki/1.28inch_Touch_LCD. Online, navštíveno 22/5/2024.



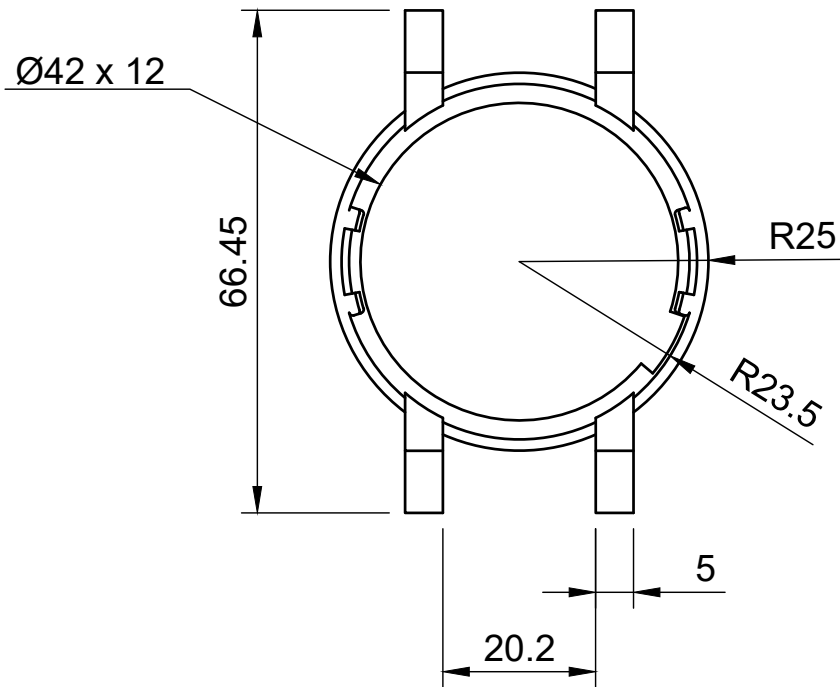
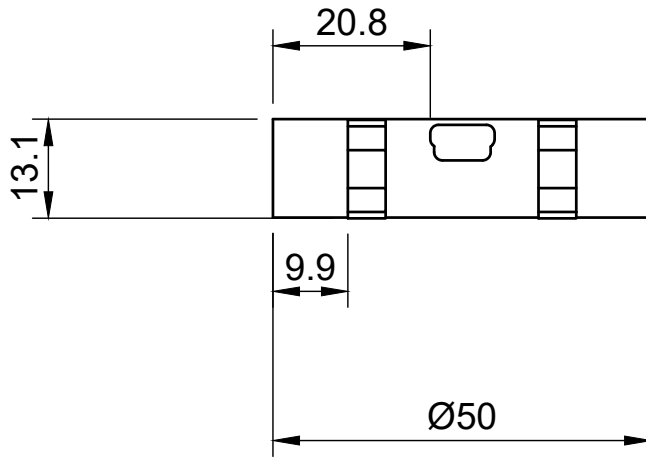
Příloha A



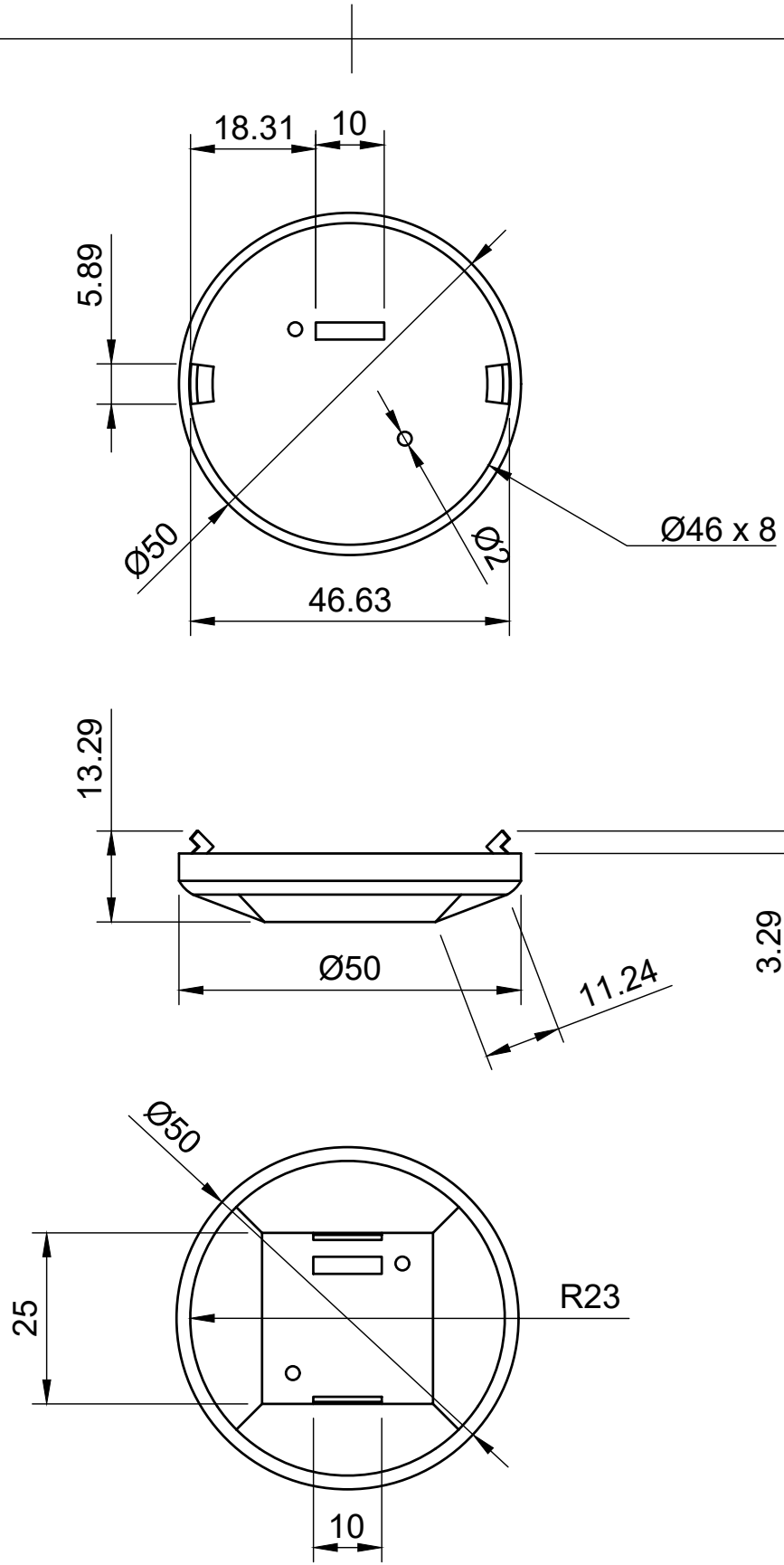
Obrázek A.31: Schéma plošného spoje



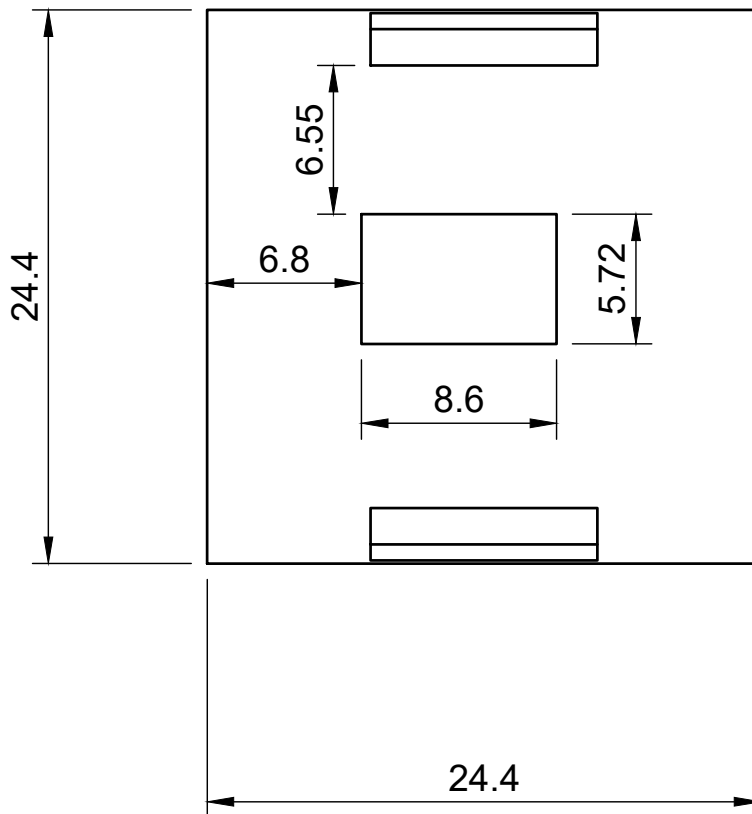
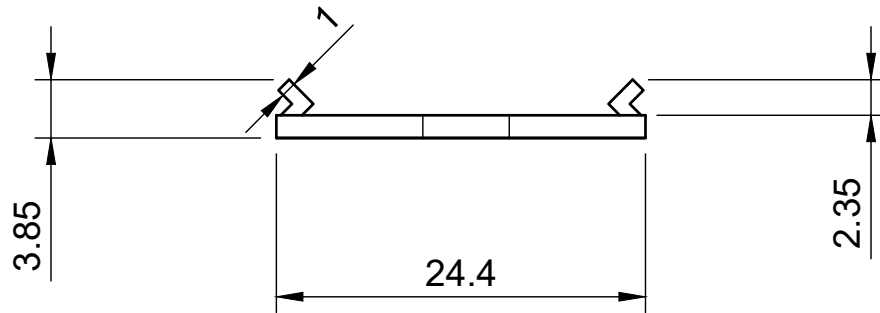
Dept.	Technical reference	Created by Dominik Patzák 5/24/2024	Approved by	
		Document type	Document status	
		Title Vrchní část	DWG No.	
		Rev.	Date of issue 5/24/2024	Sheet 1/1



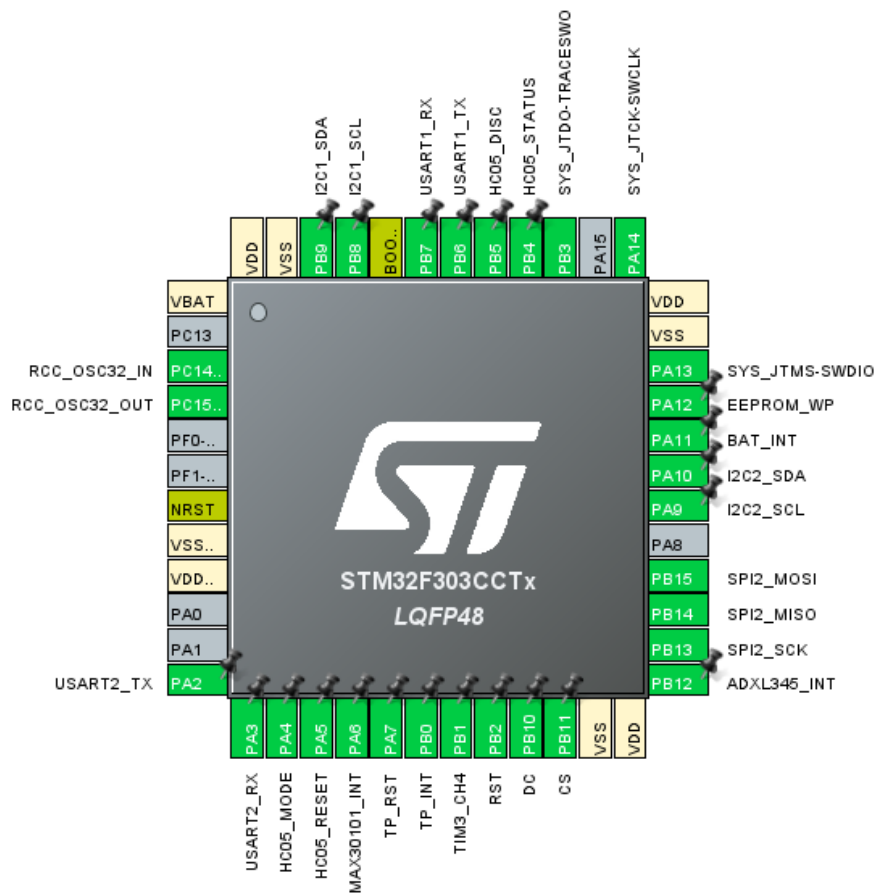
Dept.	Technical reference	Created by Dominik Patzák 5/24/2024	Approved by	
		Document type	Document status	
		Title Prostřední vrstva	DWG No. 2	
		Rev.	Date of issue 5/24/2024	Sheet 1/1



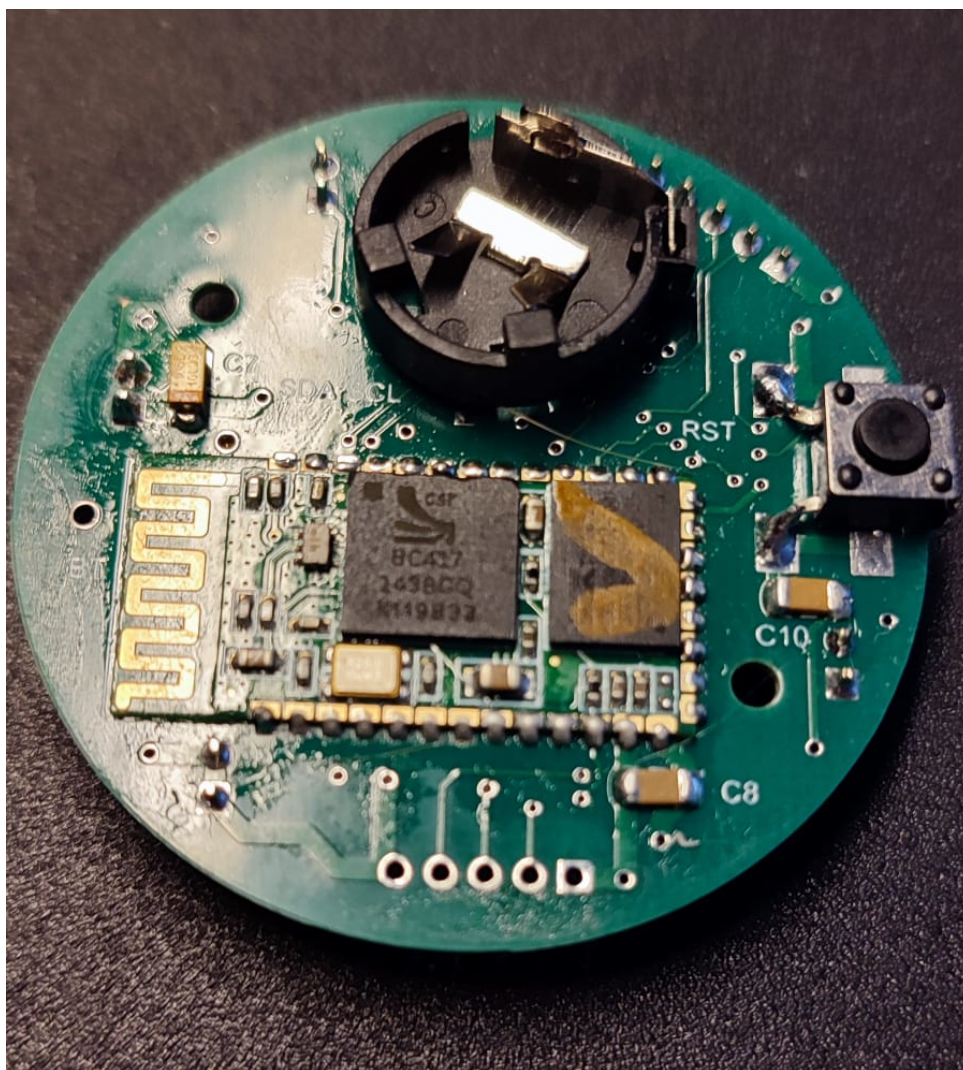
Dept.	Technical reference	Created by Dominik Patzák 5/24/2024	Approved by	
		Document type	Document status	
		Title Spodní část	DWG No. 3	
		Rev.	Date of issue 5/24/2024	Sheet 1/1



Dept.	Technical reference	Created by Dominik Patzák 5/24/2024	Approved by		
		Document type	Document status		
		Title Víko	DWG No. 4		
		Rev.	Date of issue 5/24/2024	Sheet 1/1	

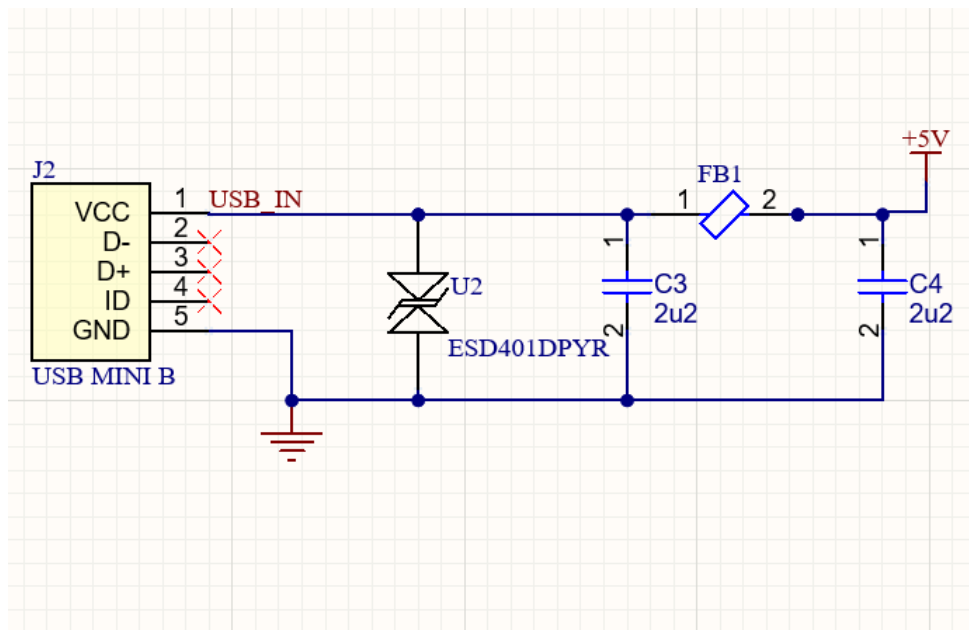


Obrázek A.1: Použité piny procesoru (Převzato z prostředí STM32CUBEIDE)

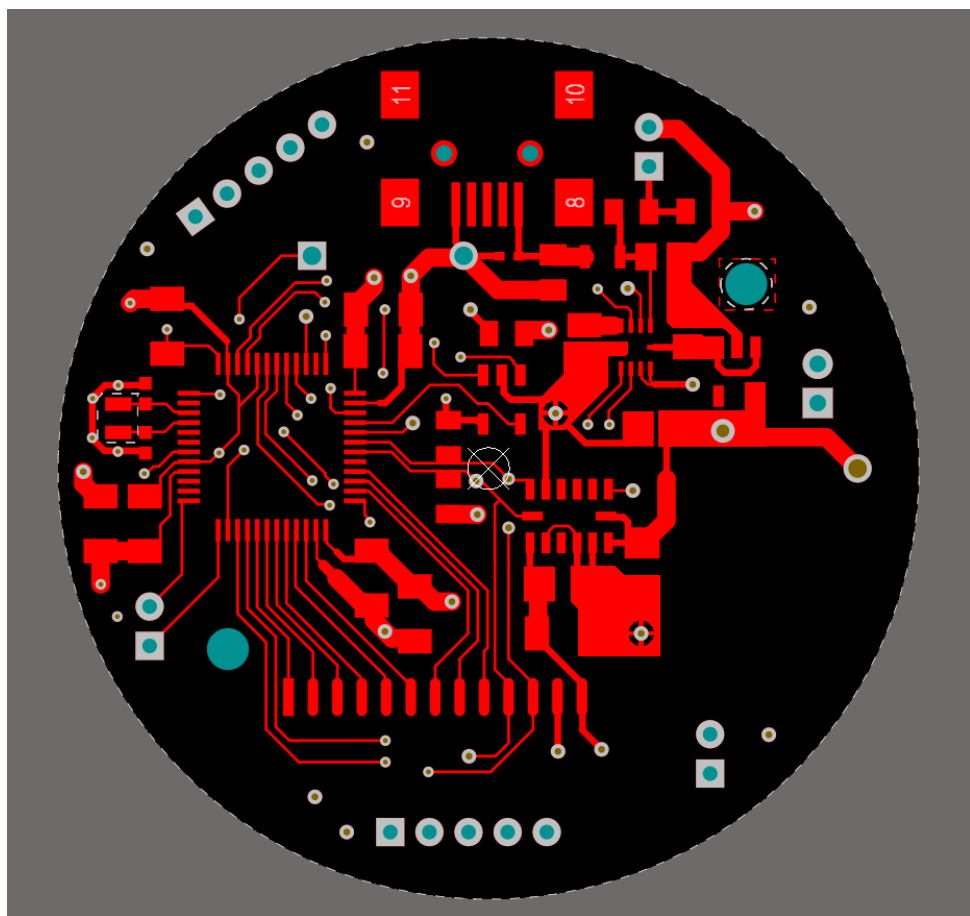


Obrázek A.3: Osazená deska - vstava bot

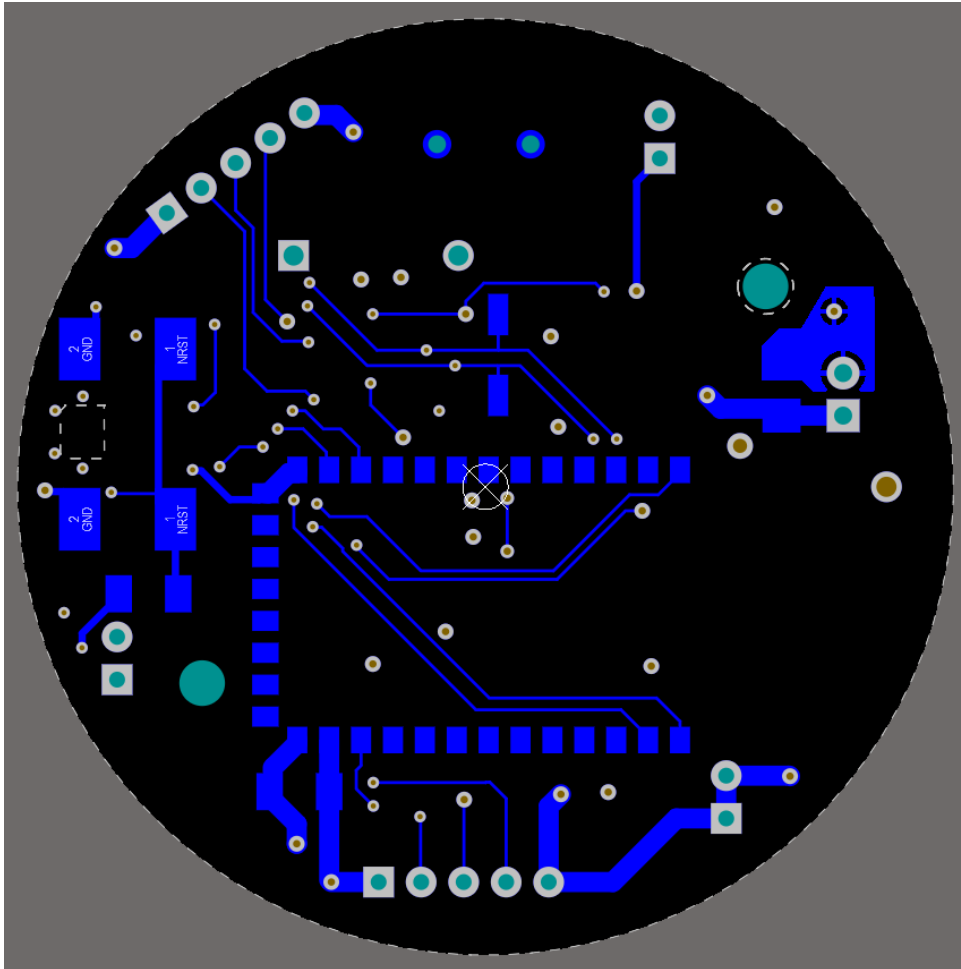
A.



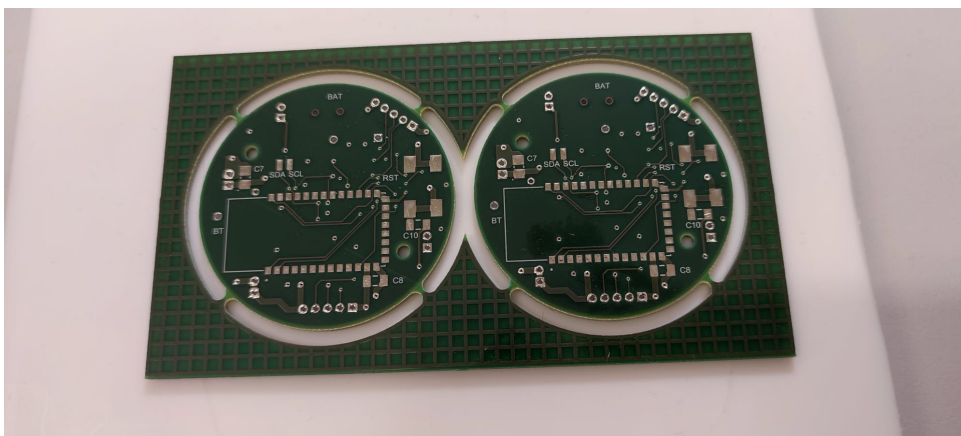
Obrázek A.4: Schéma USB vstupu



Obrázek A.5: Plošný stop - vrchní signálová vrstva

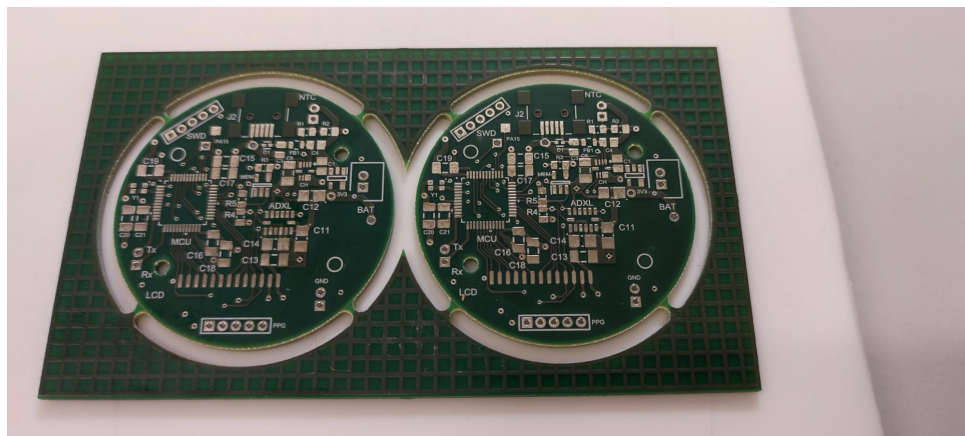


Obrázek A.6: Plošný spoj - spodní signálová vrstva

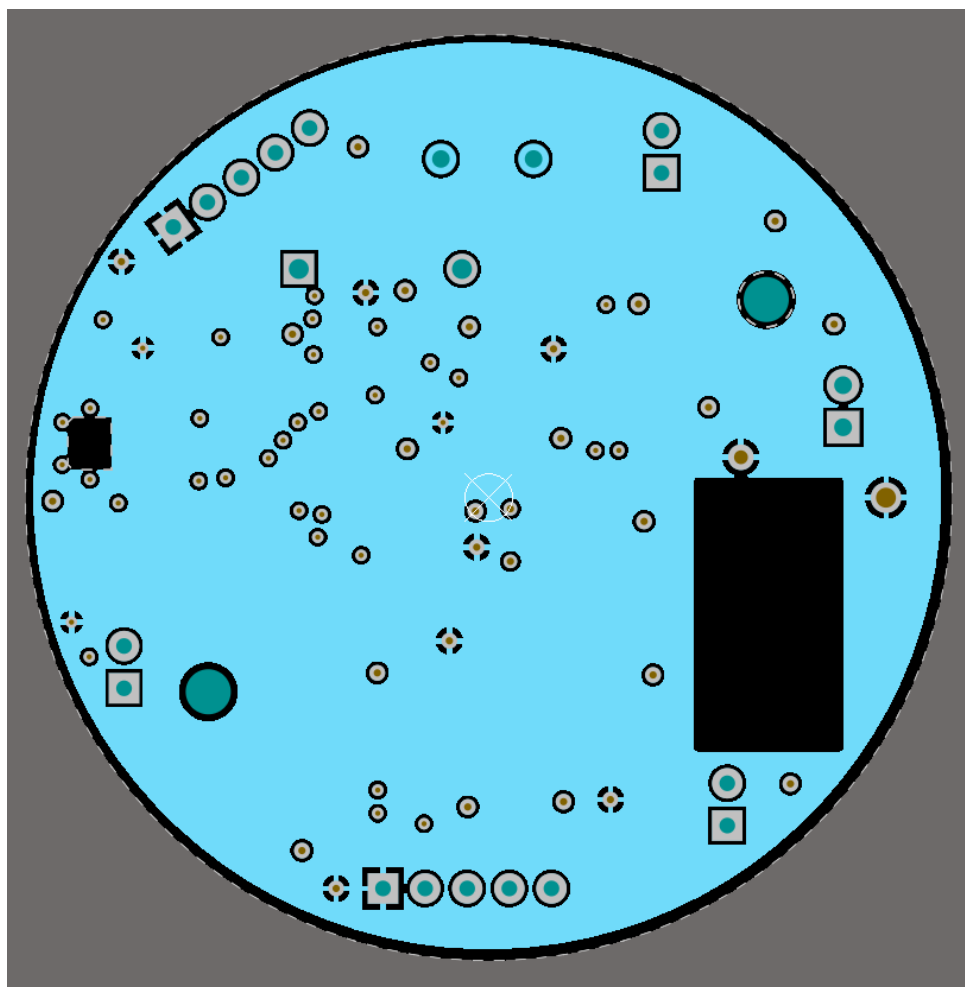


Obrázek A.7: První prototyp - Spodní vrstva

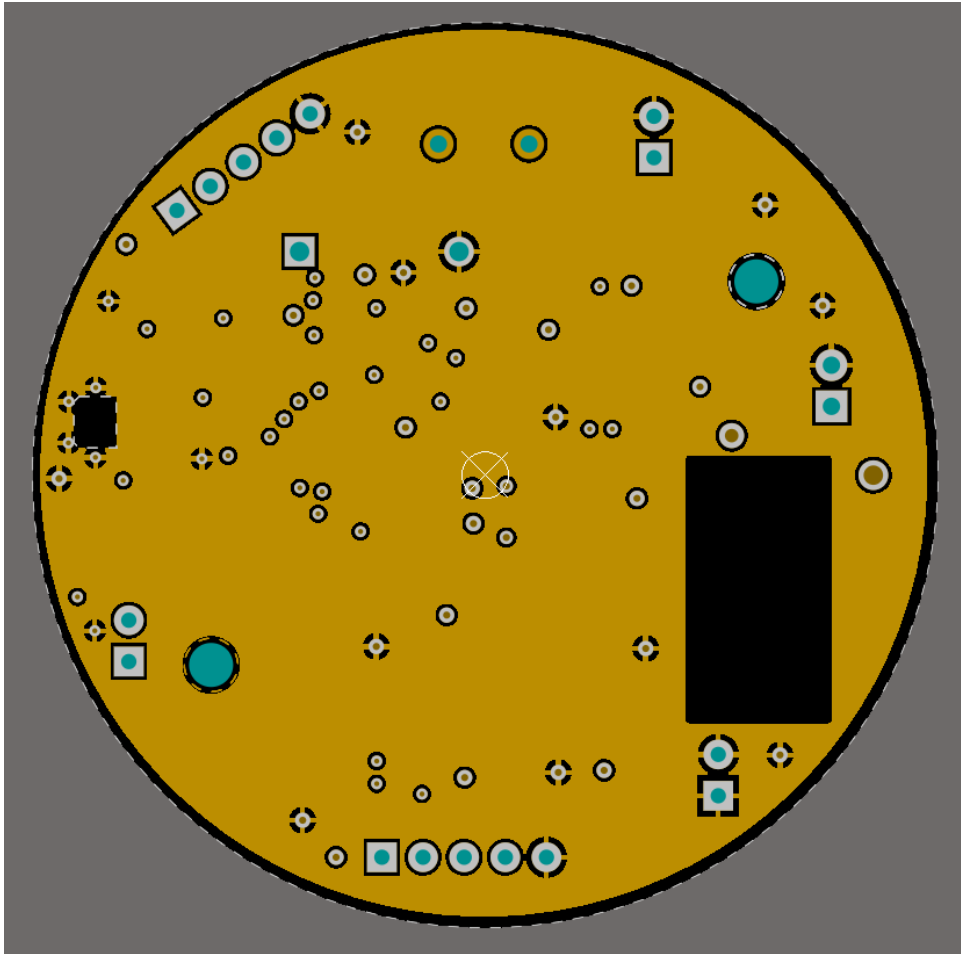
A.



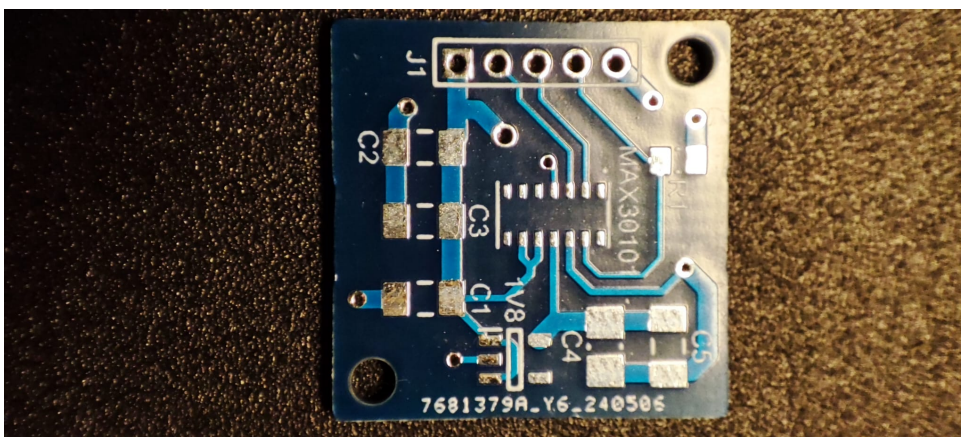
Obrázek A.8: První prototyp - Vrchní vrstva



Obrázek A.9: Plošný spoj - Napájecí vrstva

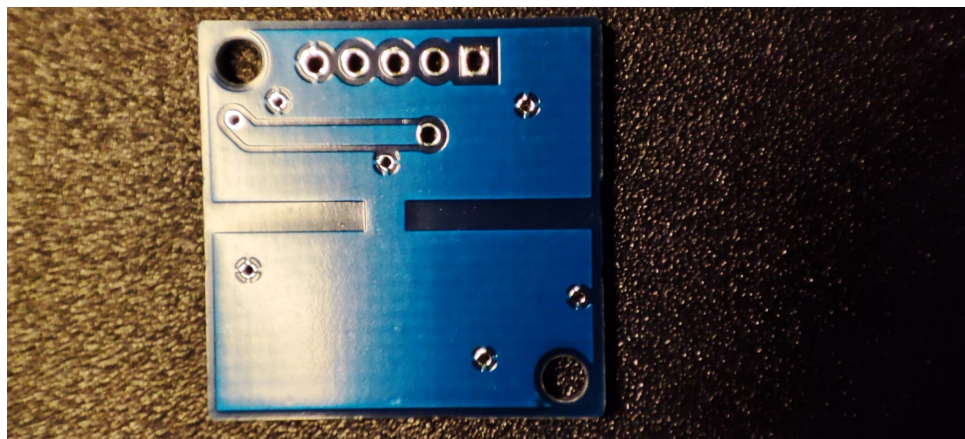


Obrázek A.10: Plošný spoj - Zemnicí vrstva

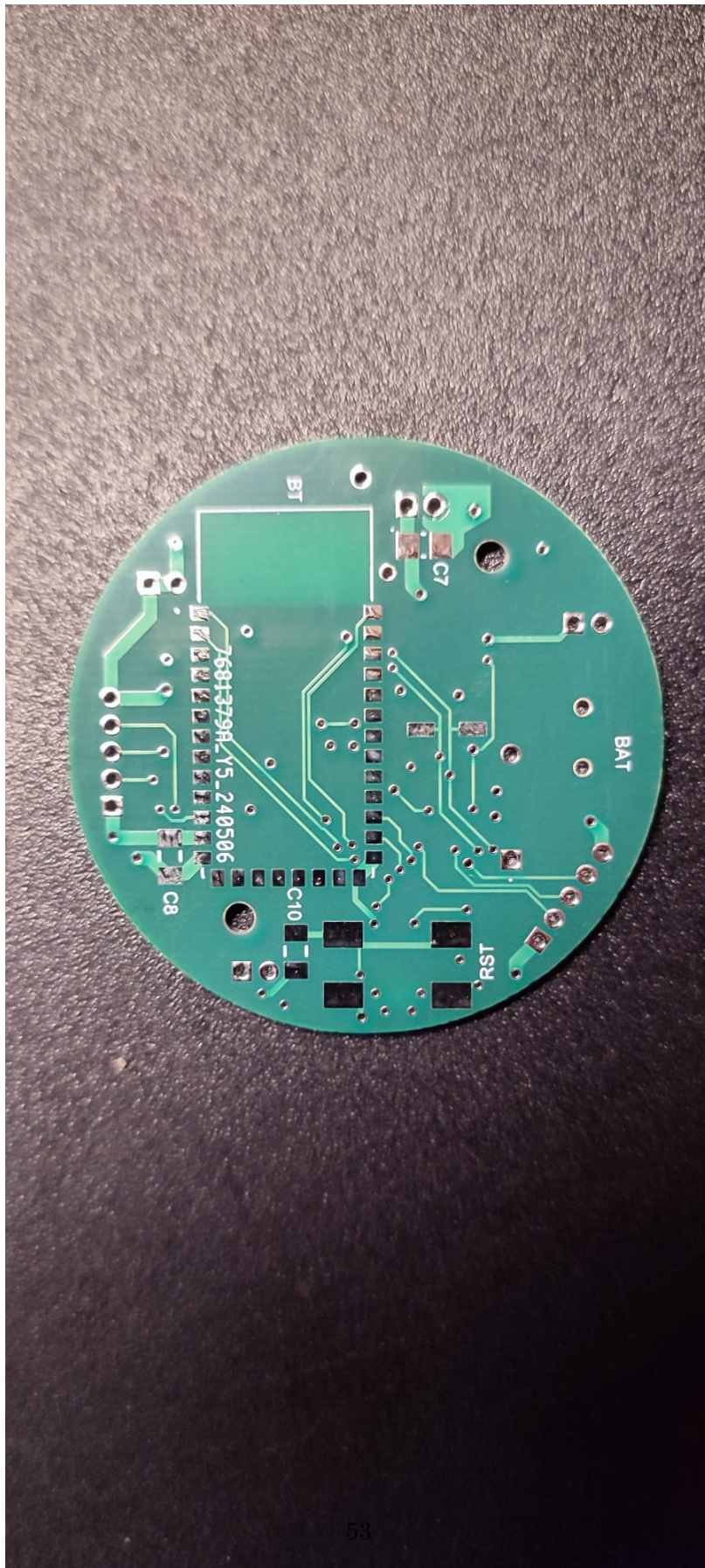


Obrázek A.11: Oxymetr - Vrchní vrstva

A.

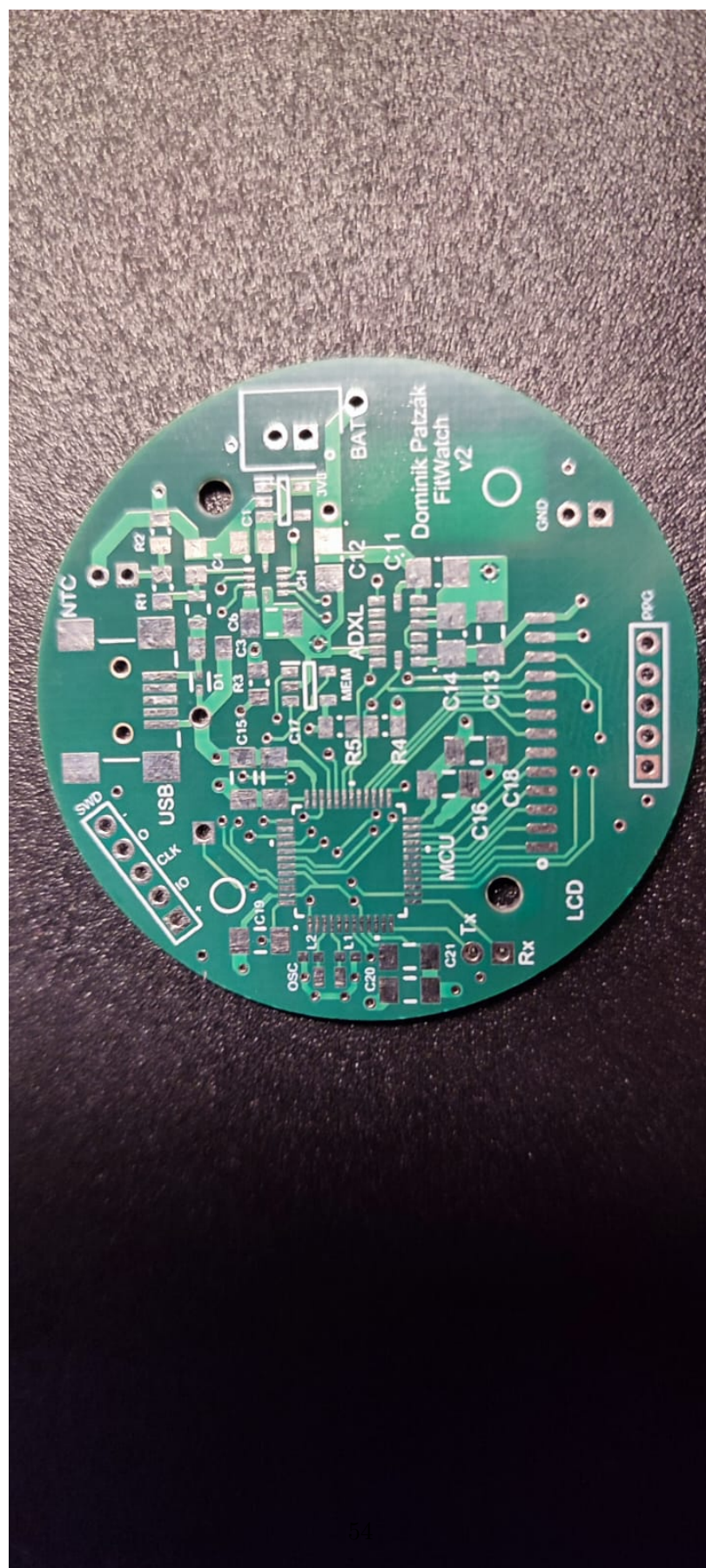


Obrázek A.12: Oxymetr - Spodní vrstva

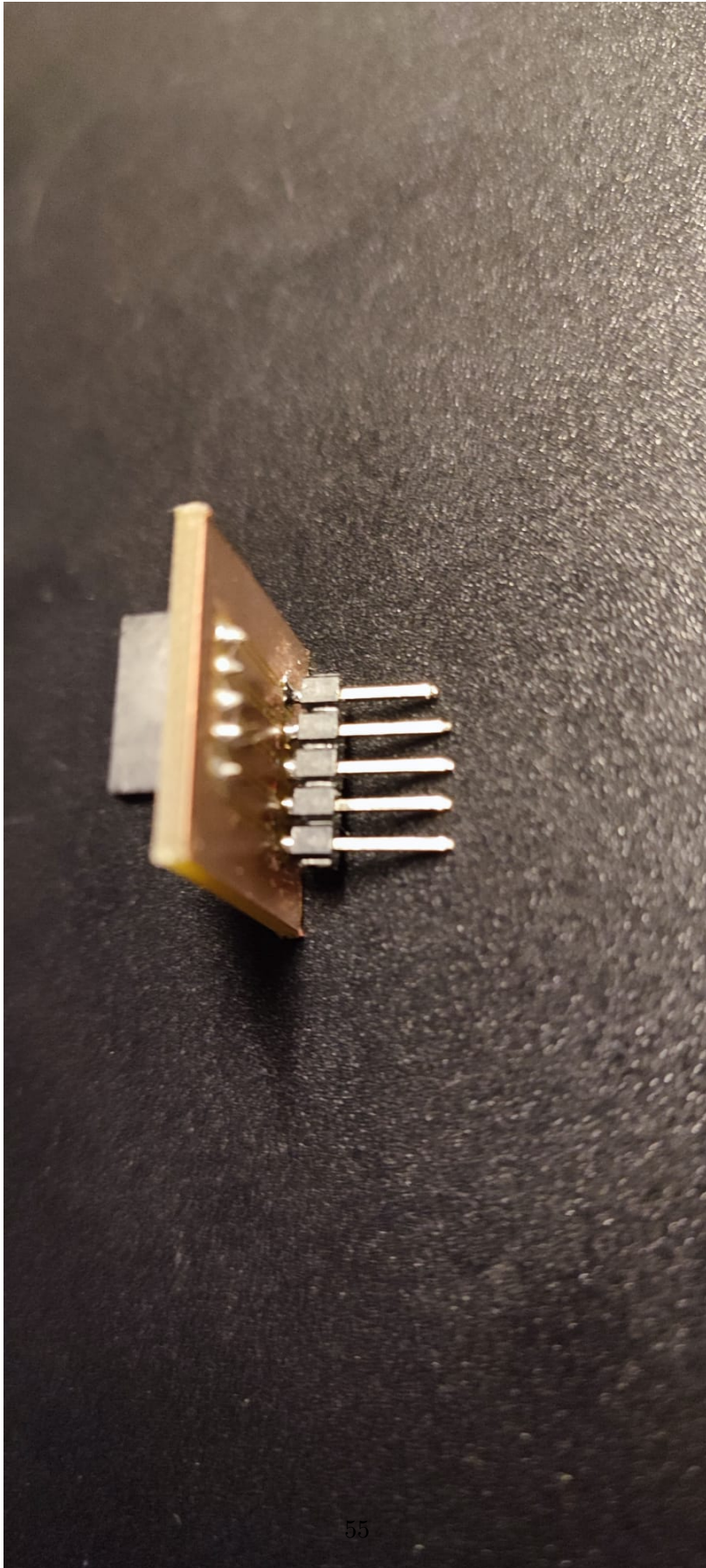


Obrázek A.13: Druhý prototyp - Spodní vrstva

A.

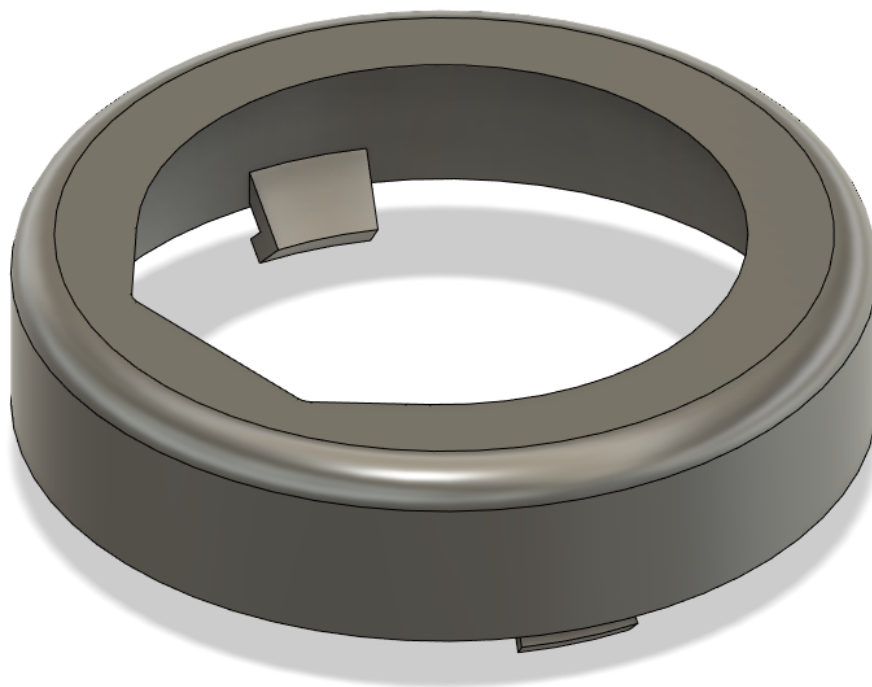


Obrázek A.14: Druhý prototyp - Vrchní vrstva

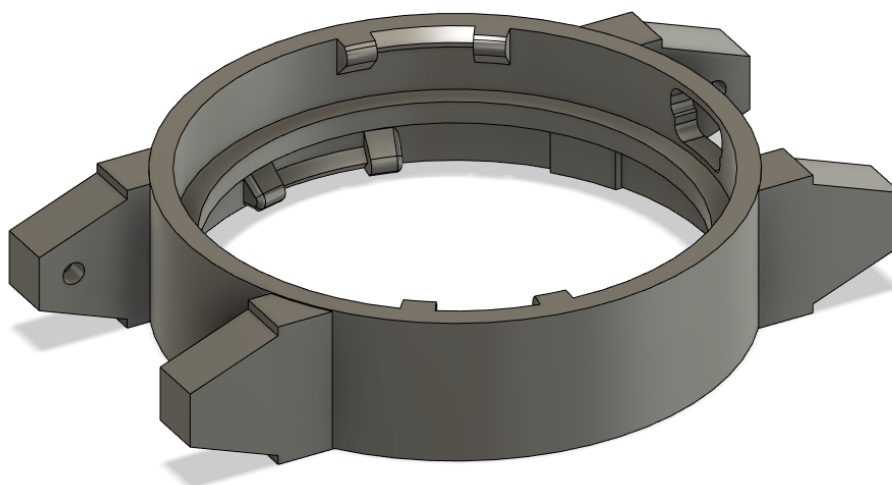


Obrázek A.15: Redukční deska

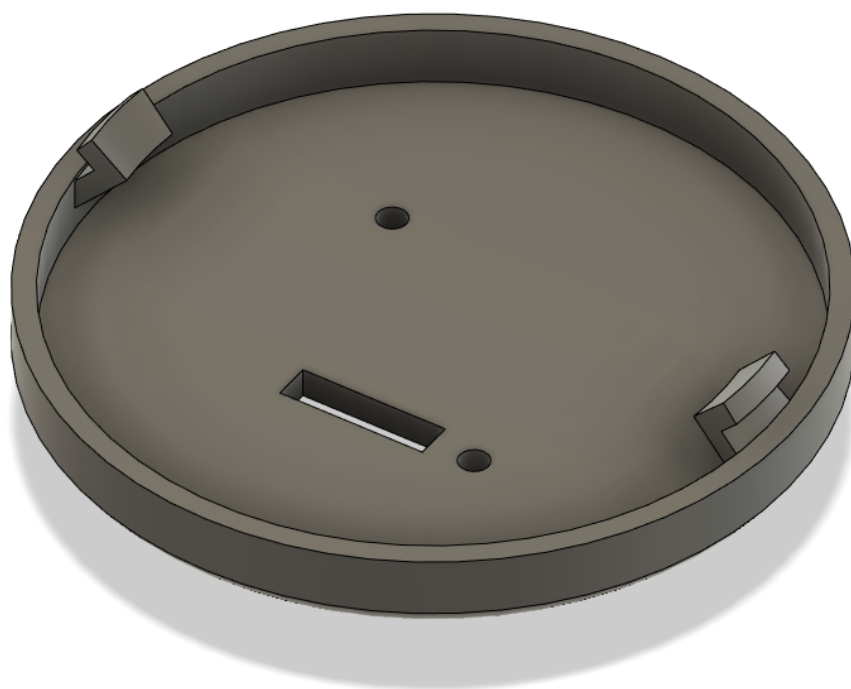
A.



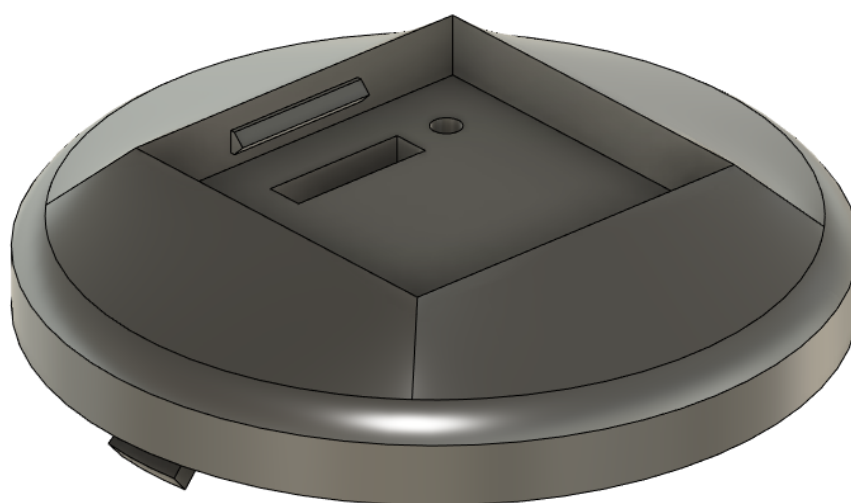
Obrázek A.16: Pouzdro - Vrchní část



Obrázek A.17: Pouzdro - prostřední část

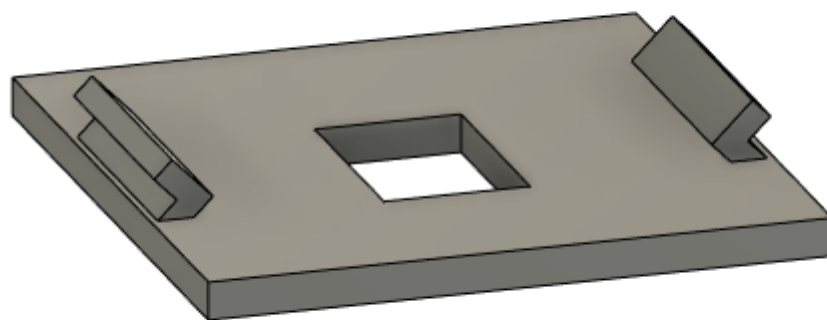


Obrázek A.18: Pouzdro - spodní část, pohled shora



Obrázek A.19: Pouzdro - spodní část, pohled zdola

A.



Obrázek A.20: Pouzdro - víko na oxymetr



Obrázek A.21: Výsledné pouzdro - Pohled 1

A.



60

Obrázek A.22: Výsledné pouzdro - Pohled 2



Obrázek A.23: Výsledné pouzdro - Pohled 3

A.

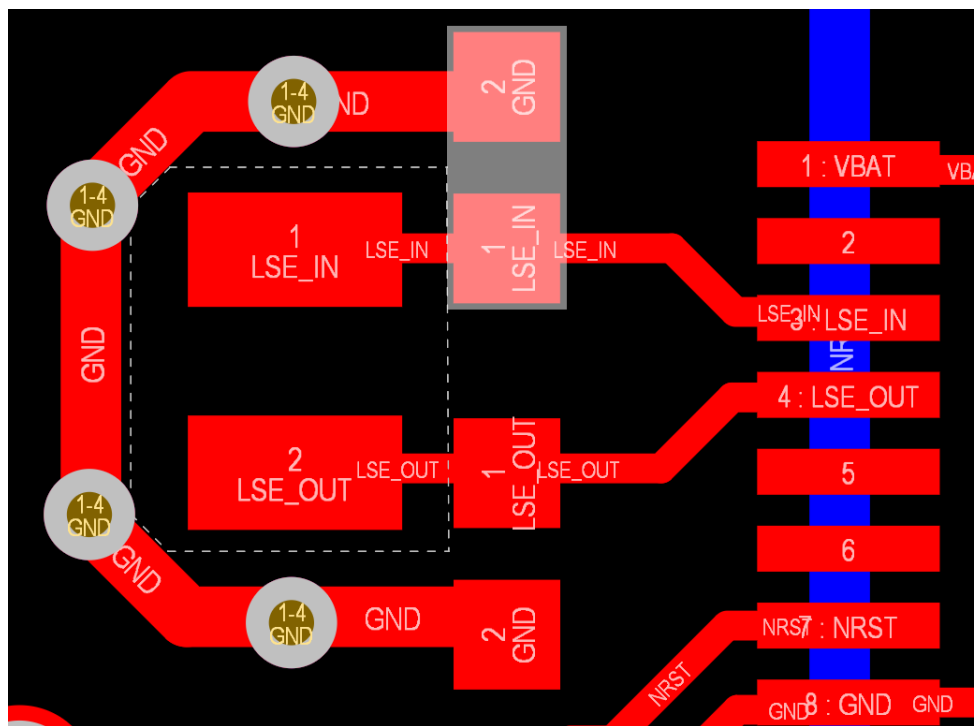


Obrázek A.24: Displej - domovská obrazovka

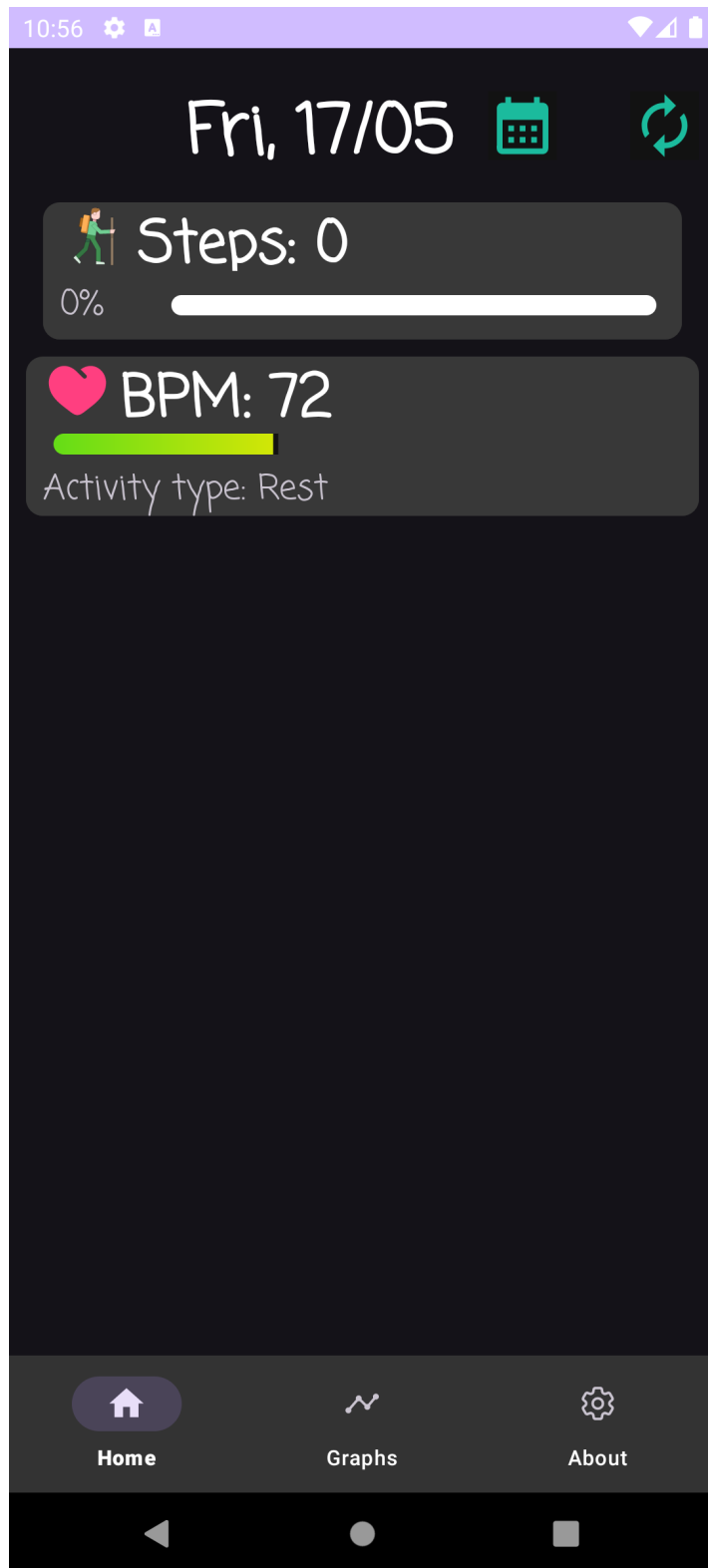


Obrázek A.25: Displej - vedlejší obrazovka

A.



Obrázek A.26: Design krystalu

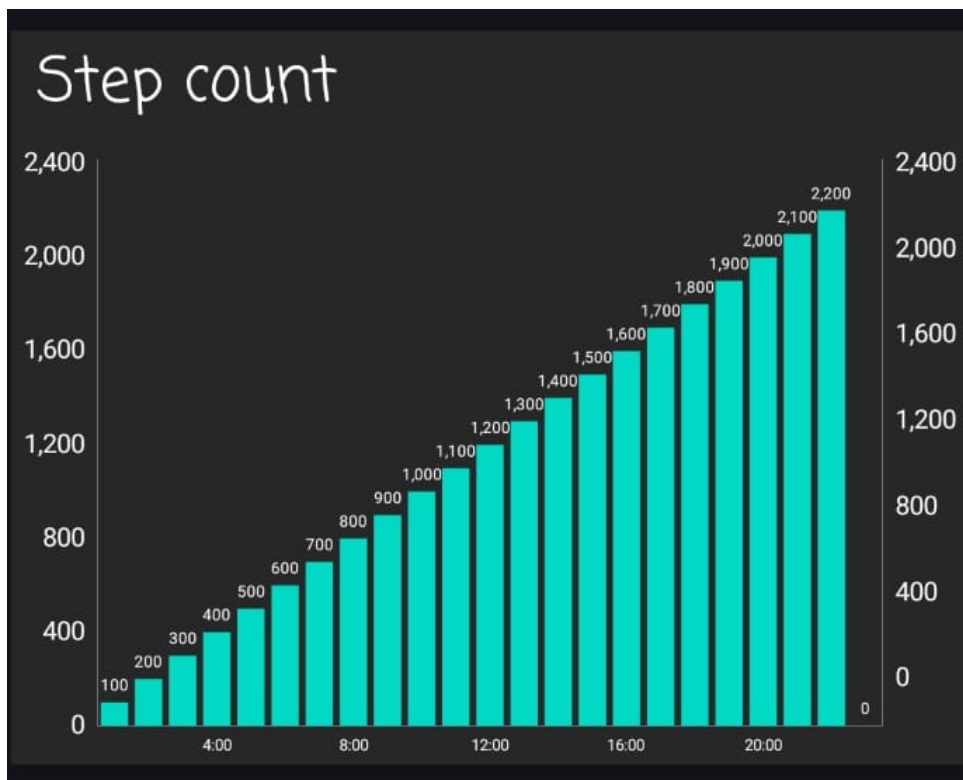


Obrázek A.27: Aplikace - Domovská stránka

A.

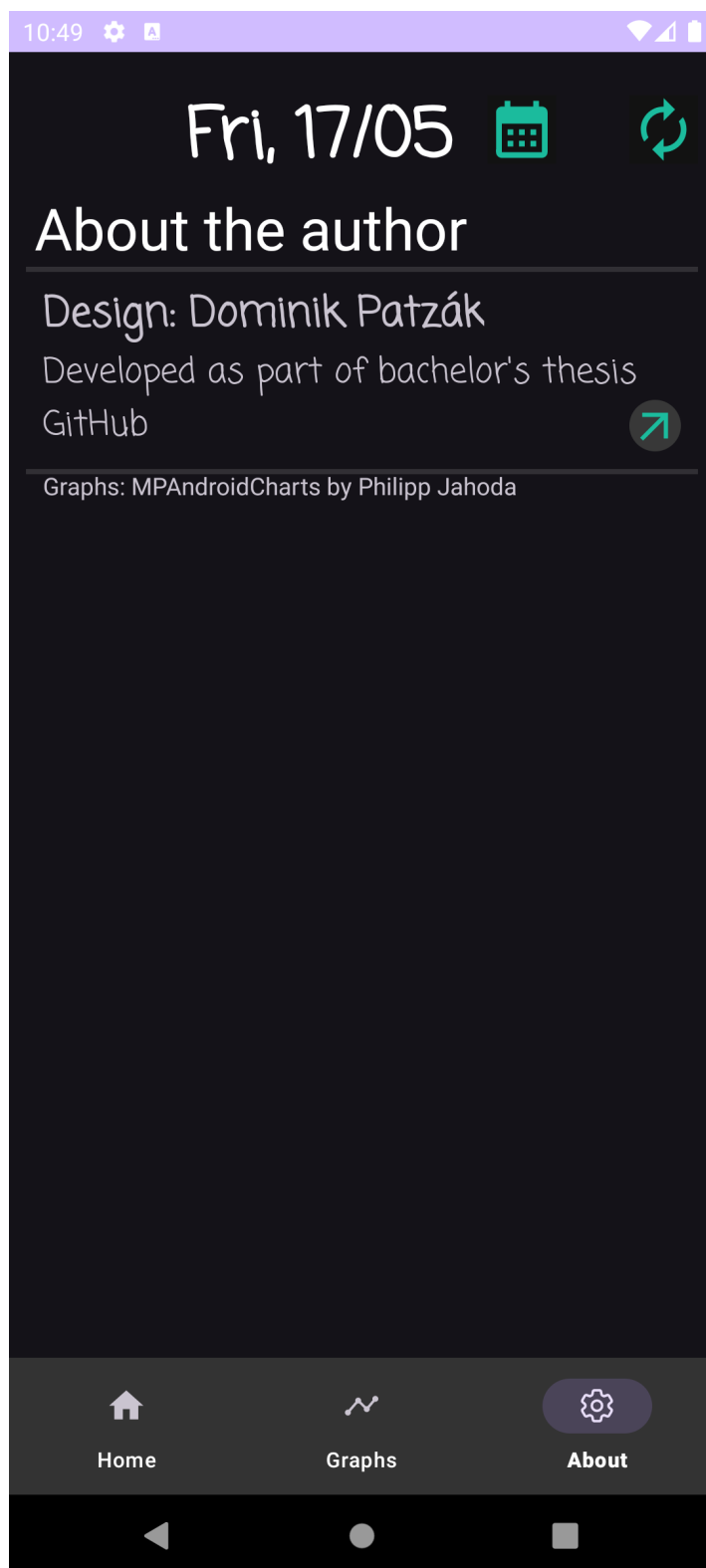


Obrázek A.28: Aplikace - Grafy



Obrázek A.29: Vizualizace ušlých kroků

A.



Obrázek A.30: Aplikace - Informace

Číslo pinu	Jméno	Funkce
1	<i>VBAT</i>	Vstup pro záložní napájení z baterie
2	<i>RCC_OSC32_IN</i>	Vstup krystalu 32.768 kHz
3	<i>RCC_OSC32_OUT</i>	Výstup krystalu 32.768 kHz
6	<i>NRST</i>	Reset procesoru
12	<i>USART2_TX</i>	UART - vysílač
13	<i>USART2_RX</i>	UART - přijímač
14	<i>HC05_MODE</i>	Bluetooth - režim: programování/data
15	<i>HC05_RESET</i>	Bluetooth - Reset modulu
16	<i>MAX30101_INT</i>	Přerušení oxymetru
17	<i>TP_RST</i>	Výstup pro reset dotykového snímače
18	<i>TP_INT</i>	Přerušení z dotykového snímače
19	<i>TIM_CH4</i>	PWM výstup pro podsvícení displeje
20	<i>RST</i>	Reset displeje
21	<i>DC</i>	Režim: programování/data displeje
22	<i>CS</i>	Výstup - Umožnění komunikace s displejem
25	<i>ADXL345_INT</i>	Přerušení akcelerometru
26	<i>SPI2_SCK</i>	SPI - hodiny
27	<i>SPI2_MISO</i>	SPI - Výstup displeje, příjem MCU
28	<i>SPI2_MOSI</i>	SPI - Výstup MCU, příjem displeje
30	<i>I2C2_SCL</i>	<i>I²C</i> - hodiny
32	<i>I2C2_SDA</i>	<i>I²C</i> - datová linka
33	<i>BAT_INT</i>	Přerušení z MP2667
34	<i>EEPROM_WP</i>	Umožnění zápisu do paměti EEPROM
37	<i>SYS_JTMS – SWDIO</i>	SWD - programovací vstup
39	<i>SYS_JTCK – SWCLK</i>	SWD - hodiny
41	<i>SYS_JTDO – TRACESWO</i>	SWD - programovací výstup
42	<i>HC05_STATUS</i>	Bluetooth - stav připojení k telefonu
43	<i>HC05_DISC</i>	Bluetooth - odpojení
44	<i>USART1_TX</i>	UART - vysílač
45	<i>USART1_RX</i>	UART - přijímač
46	<i>BOOT</i>	Výběr, odkud se BOOTuje program
47	<i>I2C1_SCL</i>	<i>I²C</i> - hodiny
48	<i>I2C1_SDA</i>	<i>I²C</i> - datová linka

Tabulka A.1: Výčet pinů procesoru