



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

APLIKACE POČÍTAČE V ULTRALEHKÉM LETOUNU

APPLICATIONS OF COMPUTERS IN ULTRAMICRO AIRCRAFT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. DANIEL ŽŮREK

VEDOUcí PRÁCE

SUPERVISOR

prof. Dr. Ing. PAVEL ZEMČÍK

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2016/2017

Zadání diplomové práce

Řešitel: **Žůrek Daniel, Bc.**

Obor: Počítačové a vestavěné systémy

Téma: **Aplikace počítače v ultralehkém letounu**

Applications of Computers in Ultramicro Aircraft

Kategorie: Uživatelská rozhraní

Pokyny:

1. Prostudujte literaturu na téma provoz ultralehkých letounů a letounů všeobecného letectví, dále prostudujte možnosti využití "embedded systémů" a mobilních systémů v letadlech.
2. Navrhněte postup tarifikace provozu letadel/vrtulníků se zaměřením na snadnou instalaci a nízké pořizovací náklady na techniku v letadle.
3. Popište postup zjišťování parametrů provozu a diskutujte dosažitelné vlastnosti řešení.
4. Implementujte systém a demonstруйте jeho funkčnost na vhodném příkladu použití.
5. Diskutujte dosažené výsledky a možnosti pokračování práce.

Literatura:

- Dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Body 1-3 zadání

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Zemčík Pavel, prof. Dr. Ing.,** UPGM FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 24. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Diplomová práce se zabývá možnostmi pro tarifkaci provozu ultralehkých letadel nebo vrtulníků. Cílem je vytvořit systém pro zjišťování provozních parametrů letadla, s hlavním zaměřením na detekci a vyhodnocení pohybu. V práci jsou popsány metody snímání otáček, měření zrychlení, způsoby měření frekvence, bezdrátová komunikace, komunikace po sériové lince, hardwarová realizace a softwarová implementace výsledného vestavěného systému, a nakonec popis implementace mobilní aplikace.

Abstract

The master thesis deals with the possibilities for monitoring the operation of ultralight aircraft or helicopters. The aim is to create a system for the determination of aircraft operating parameters with a main focus on motion detection and evaluation. The thesis describes speed sensing methods, acceleration measurement, frequency measurement methods, wireless communication, serial line communication, hardware and software implementation of the embedded system, and finally description of the implementation of the mobile application.

Klíčová slova

otáčkoměr, akcelerometr, měření frekvence, UART, I2C, Bluetooth, mikrokontrolér, Android

Keywords

tachometer, accelerometer, frequency measurement, UART, I2C, Bluetooth, microcontroller, Android

Citace

ŽŮREK, Daniel. *Aplikace počítače v ultralehkém letounu*. Brno, 2017. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Dr. Ing. Pavel Zemčík

Aplikace počítače v ultralehkém letounu

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana prof. Dr. Ing. Pavla Zemčíka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Daniel Žůrek
24. května 2017

Poděkování

Chci poděkovat vedoucímu této práce prof. Dr. Ing. Pavlu Zemčíkovi za poskytnutí cenných rad a pomoci při vypracování tohoto textu. Dále bych chtěl poděkovat Martinu Škrobánkovi za pomoc a rady při realizaci a návrhu plošného spoje. Nakonec chci poděkovat své rodině za podporu po celou dobu mého studia.

Obsah

1	Úvod	2
2	Shrnutí současného stavu	3
2.1	Otáčkoměry	3
2.2	Měření frekvence	18
2.3	Akcelerometry	20
2.4	Mobilní zařízení	25
2.5	Sériová komunikace	30
3	Analýza a návrh řešení	36
3.1	Analýza současného stavu	36
3.2	Návrh řešení	37
4	Řešení a implementace	43
4.1	Hardwarová část řešení	43
4.2	Softwarová část řešení	49
5	Testování	69
5.1	Tvorba testovacího signálu	69
5.2	Testování mobilní aplikace	70
6	Závěr	72
	Literatura	73
	Přílohy	74
	Seznam příloh	75
A	Obsah přiloženého CD	76

Kapitola 1

Úvod

Velký pokrok informačních technologií lze v současné době vidět ve skoro každém aspektu lidské činnosti. Vývoj v elektronice vede ke snižování ceny, zvyšování výkonu, snižování příkonu, nebo ke zmenšování rozměrů elektronických součástek a přístrojů. Tím dochází k obrovskému rozšíření jednotlivých technologií do celé společnosti. Dalším důsledkem vývoje je vytváření aplikací nebo zařízení, které jsou mnohem efektivnější a prostorově úspornější než tomu bylo v minulosti.

Vzorovým příkladem je využití těchto technologií v letectví, kde je dnes trendem slučování jednotlivých palubních přístrojů do jednoho celku. Výsledkem tedy není více různých měřících zařízení rozmístěných na palubní desce, ale snaha o vytvoření komplexního měřícího systému. S rozvojem českého amatérského letectví se dostala do oblíbenosti především malá ultralehká letadla. Piloti dnes už nejsou nuceni pořizovat si svá vlastní letadla, ale většina aeroklubů nabízí možnost si takové letadlo zapůjčit. Cena zapůjčení letadla se potom odvíjí od počtu nalétaných hodin. Většina půjčoven tuto dobu stanoví podle času vzletu a přistání daného letadla, nebo je tuto dobu nucen kontrolovat pilot. Tento systém však není příliš efektivní a může vést k nepřesnostem. Hlavním cílem této diplomové práce je, na základě několika provozních údajů letadla, konkrétně počtu otáček motoru letadla a jeho zrychlení, určit celkovou dobu letu. Toho je dosaženo vytvořením měřícího systému, schopného tyto údaje snímat a vyhodnocovat. Měřící zařízení, bude umístěno na palubě letadla. Vstupem zařízení je frekvence otáček motoru a údaj o jeho zrychlení. Tato data se vyhodnotí a odešlou pomocí bezdrátového vestavěného Bluetooth modulu do mobilního telefonu. Další částí měřícího systému je mobilní aplikace, která umožní zobrazení naměřených výsledků a měření doby letu.

Úvodní kapitola této diplomové práce se věnuje shrnutí informací o způsobech snímání otáček, měření frekvence a snímání zrychlení. Dále je popsán vývoj aplikací pro mobilní zařízení. Závěr kapitoly se soustředí na popis komunikace u počítačových periférií. Následující kapitola obsahuje analýzu a návrh výsledného systému pro tarifkaci letadla. Zde je uveden rozbor informací z předešlé kapitoly a jejich následného uplatnění v této práci. V další kapitole je uveden popis hardwarové realizace výsledného zařízení spolu s popisem softwarové implementace řídicí aplikace pro mikrokontrolér a mobilní aplikaci. Poslední kapitola se zabývá testováním a vyhodnocením.

Kapitola 2

Shrnutí současného stavu

Tato kapitola obsahuje souhrn vybraných informací z oblasti měření zrychlení, snímání otáček a měření frekvence, vývoje mobilních aplikací a způsobů sériové komunikace. Informace byly použity k vypracování této práce a k vytvoření výsledného měřicího systému. První část kapitoly se věnuje otáčkoměrům. Ve druhé části je popisována technologie pro měření zrychlení a frekvence. Poslední část obsahuje popis mobilní platformy Android a technologie pro sériovou komunikaci.

2.1 Otáčkoměry

Jedním z parametrů výsledného zařízení pro určení trvání doby letu je frekvence otáček motoru letadla. Zařízení bude tedy vykazovat funkcionalitu otáčkoměru.

Otáčkoměry jsou měřicí přístroje, které se používají k měření otáček motoru, nebo jiných rotačních těles. Klasifikace otáčkoměrů se může lišit na základě fyzikálního principu měření, měří-li časovou či frekvenční metodou, nebo podle připojení otáčkoměru k snímanému objektu. Nejobecnější rozdělení otáčkoměrů je tedy dle připojení na dotykové (kontaktní), nebo bezdotykové (bezkontaktní). Jednotkou pro měření rychlosti otáčení se uvádí počet otáček za minutu (ot/min - RPM). Otáčkoměry jsou široce využívány v automobilové nebo letecké technice.

2.1.1 Kontaktní měření otáček

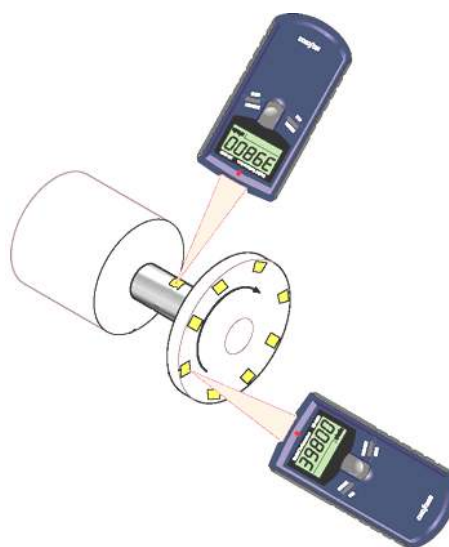
Rotační pohyb je u kontaktních otáčkoměrů snímán v ose otáčení, kdy se snímá například pohyb řemenice nebo hřídele. V tomto případě se často využívá soustružnického vpichu, který zajišťuje stabilnější upevnění snímací hlavy otáčkoměru. Další způsob, jak měřit rotační pohyb, je využitím obvodové rychlosti, kdy se přiloží snímací kolečko k obvodu rotující řemenice nebo kola. V tomto druhém případě lze měřit nejen rychlost otáčení, ale i obvodovou rychlost. Toho se využívá pro měření rychlosti posuvu u součástí, které pracují na základě lineárního pohybu, například dopravníkový pás. Kontaktní otáčkoměry bývají vybaveny výměnnými koncovkami různého typu v závislosti na měřících podmínkách. Povrch těchto nástavců je obvykle gumový, pro zajištění přesného a spolehlivého pohybu bez prokluzování [10].



Obrázek 2.1: Kontaktní měření otáček. Vlevo - osa otáčení, Vpravo - obvodová rychlost.¹

2.1.2 Bezkontaktní měření otáček

Bezkontaktní otáčkoměry, na rozdíl od kontaktních, nevyžadují pro měření dotyk se snímaným objektem. Snímání pohybu je realizováno pomocí světelného paprsku, nebo použitím magnetických součástí. Bezdotyková technologie měření otáček je ve většině případů rychlejší, jednodušší a praktičtější oproti dotykové. Velkou výhodou je, že rotační senzor není nikdy přímo vystaven vlivům okolí a nevykazuje žádné mechanické opotřebení. Životnost senzoru se neuvádí na cykly, ale na průměrnou dobu do selhání (Mean Time To Failure - MTTF). Avšak bezkontaktní metoda měření není vždy ta nejvýhodnější. V závislosti na měřicích podmínkách je někdy vhodnější, nebo jediné možné, měření pomocí kontaktních otáčkoměrů. Například při měření obvodové rychlosti při malém průměru hřídele, či při uspořádání, nevhodném pro bezdotykové měření otáček[10].



Obrázek 2.2: Ukázka bezkontaktního měření otáček.²

¹Převzato z: www.onosokki.co.jp/English/hp_e/products/application/tach/tach_a_1_1.htm

²Převzato z: www.onosokki.co.jp/English/hp_e/products/application/tach/tach_a_1_2.htm

Z dosud popsaných způsobu měření otáček, nelze jednoznačně určit, která z těchto metod je nejvýhodnější. Obě tyto metody se vzájemně doplňují, a právě z tohoto důvodu jsou pro univerzální použití vhodné otáčkoměry, které kombinují kontaktní i bezkontaktní způsob měření.

Následující část kapitoly je věnována rozdělení otáčkoměrů podle použití odlišných fyzikálních metod při snímání otáčení. Na základě odlišných fyzikálních principů můžeme otáčkoměry rozlišit do následujících kategorií:

- Mechanické
- Spojité indukční
- Impulsní
- Stroboskopické

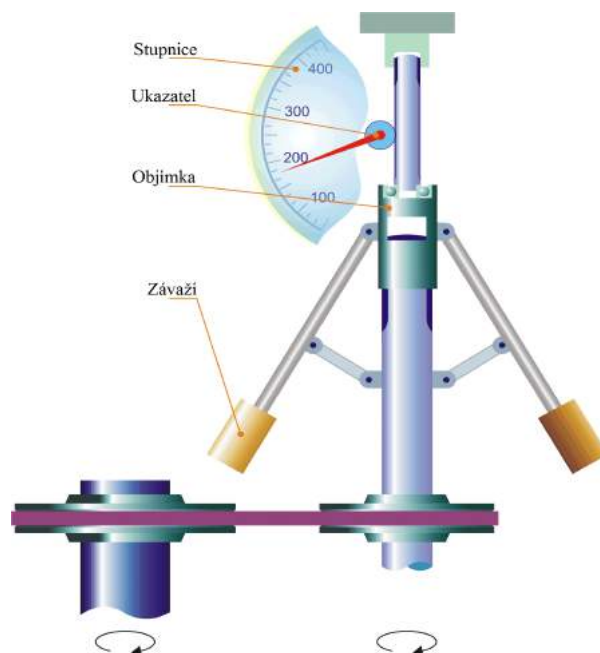
2.1.3 Mechanické otáčkoměry

Mechanické otáčkoměry využívají účinky odstředivé síly na rotující hmotu, jejíž pohyb se převádí na snímač polohy. Mechanické otáčkoměry znázorňují frekvenci otáčení přímo na stupnici přístroje. Nevýhodou těchto otáčkoměrů je fakt, že mechanická závaží mají určitou setrvačnost, což má nepříznivý vliv na přesnost měření. Mechanické otáčkoměry nejsou schopny určit směr otáčení[10].

Odstředivý otáčkoměr

Odstředivé otáčkoměry jsou založeny na principu známém jako Wattův odstředivý regulátor, který byl zaveden v roce 1782, a používal se na regulaci přívodu páry u parních strojů. Odstředivé otáčkoměry jsou tedy jedny z prvních používaných otáčkoměrů.

Pro odstředivý otáčkoměr jsou typická dvě závaží, připevněná na ramenech s pružinou, a dvě objímky. Otáčkoměr pracuje na principu měření silového účinku odstředivé síly, která působí na obě závaží. Závaží se od sebe v důsledku rotace hřídele vzdalují, dokud nedojde k vyrovnání odstředivé síly a síly pružiny. Se zvyšující se rychlostí otáčení hřídele roste i odstředivá síla, která působí na obě závaží. Síla, vytvořená rotací závaží, se přenáší přes ramena na objímky, které se k sobě začínou přibližovat. Pružina, umístěná mezi objímkami, zajišťuje zpětný tah působící proti odstředivé síle závaží, a brání tedy volnému pohybu obou objímek. Zatímco při zvyšování otáček odstředivá síla přemáhá tuhost pružiny, tak při opětovném snižování otáček pružina obě objímky opět oddaluje, a tím je vrací do původní pozice. Tímto dochází k regulaci otáčení, která se odvíjí právě od tuhosti pružiny. Nevýhodou těchto otáčkoměrů byla poruchovost zapříčiněná nedostatečným mazáním[15][11].



Obrázek 2.3: Modernější konstrukce odstředivého otáčkoměru.³

2.1.4 Spojité indukční otáčkoměry

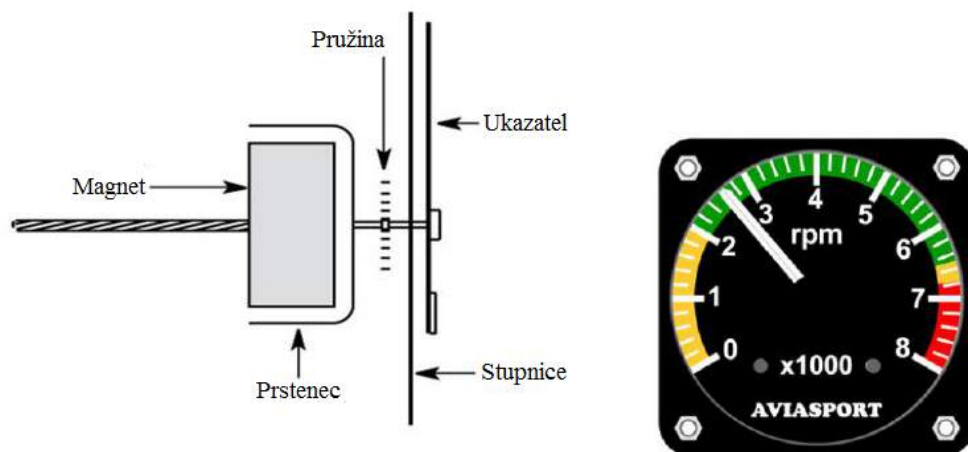
Spojité indukční otáčkoměry mohou být magnetické nebo elektrodynamické, a to na stejnosměrný nebo střídavý proud.

Elektrodynamické otáčkoměry jsou jednou z nejdůležitějších skupin otáčkoměrů. Jsou založeny na indukčním zákonu, kdy generují napětí přímo úměrné otáčivé rychlosti. Pohybující se vodič v magnetickém poli indukuje na svých koncích napětí. Podle funkce je můžeme rozdělit na generátory stejnosměrné (tachodynamy) a střídavé (tachoalternátory)[15][11].

Magnetické otáčkoměry

Magnetické otáčkoměry využívají účinek vířivých proudů indukovaných v kovovém prstenci magnetickým polem při otáčení magnetu. Hlavním prvkem otáčkoměru je otočný magnet s hřídelí. Hřídel otáčí magneticky polarizovaný kotouč, jehož magnetický tok se uzavírá přes feromagnetický plášť. Ve vzduchové mezeře mezi nimi je hliníkový prstenc, v němž se za rotace indukují vířivé proudy. Jejich magnetické pole je unášeno ve směru rotace kotouče, spirálová pružina, umístěna mezi ručičkou a prstencem, vyvozuje direktivní moment. Ukazatel spojený s prstencem se proto natočí o úhel úměrný úhlové rychlosti kotouče. Pružina zabraňuje kompletnímu otáčení prstence. K ustálení ručičky dojde ve chvíli, kdy je indukovaná magnetická síla v rovnováze se silou spirálové pružiny. Magnetické otáčkoměry jsou citlivé již od malých otáček, ovšem jejich nevýhodou je vazba na teplotní změny. To je dáno teplotní závislostí válce, který smršťováním mění své magnetické vlastnosti, čímž se mění vířivé proudy a zkreslují se tak zobrazované otáčky. Magnetické otáčkoměry se využívají pro přímé měření otáček, např. v automobilech (tachometr) nebo malých letadlech[10][15][11].

³Převzato z: <http://vlm.sjf.stuba.sk/podstranky/otacanie%20mechanicke.html>

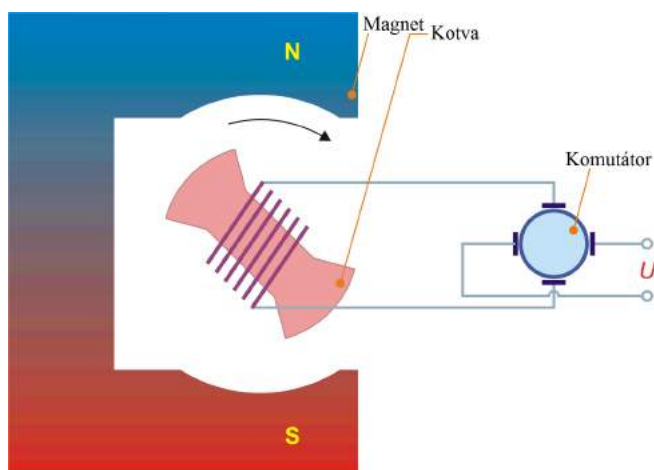


Obrázek 2.4: Magnetický otáčkoměr s ukázkou výstupu v letadle.⁴

Tachodynamo

Tachodynamo patří mezi nejnámější senzory pro snímání rychlosti. Jsou konstruovány jako dynamo, která generují stejnosměrné napětí, nebo jako alternátory pro generování střídavého napětí. Stator tachodynamo je tvořen permanentním magnetem, v jehož poli se otáčí rotor s vinutím vyvedeným na komutátor. Z komutátoru se pomocí sběračů odeberá stejnosměrné napětí, které je přímo úměrné frekvenci otáčení. Výsledná magnetická indukce, vznikající ve vzduchové mezeře, má téměř obdélníkový průběh.

Tachodynamo se používají ke snímání úhlové rychlosti. Jejich nevýhodou je velké zvlnění a problém s komutací kvůli jiskření a opotřebení. Výhodou je vysoká citlivost. Tachodynamo se běžně používá pro měření frekvence do 6000 ot/min[10].



Obrázek 2.5: Schéma tachodynamo.⁵

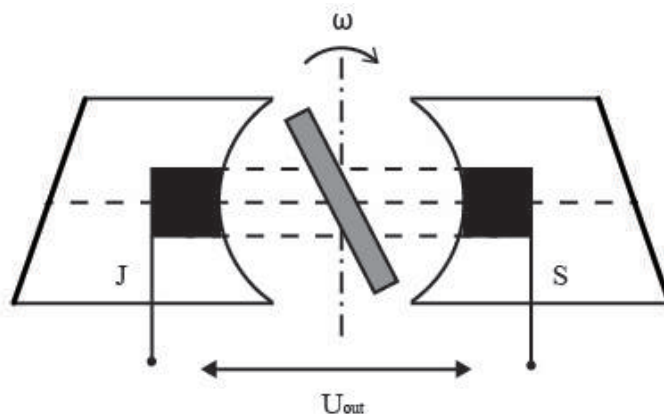
⁴Převzato z: www.datwiki.net/page.php?id=4985&find=magnetic-drag%20tachometer&searching=yes

⁵Převzato z: vlm.sjf.stuba.sk/podstranky/otacanie%20analog.html

Tachoalternátory

Tachoalternátory mohou být konstruovány několika způsoby. První možností je stator tvořený z permanentního magnetu, kdy otáčením rotorových cívek dochází k indukci napětí. Druhý typ má rotor tvořený permanentním magnetem, a ve statorových cívkách se indukují napětí úměrné otáčkám magnetu (tzv. bubínkový rotor).

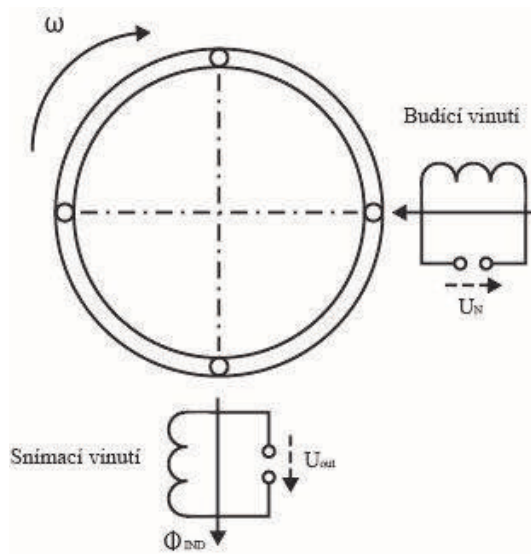
U tachoalternátorů, s rotujícím permanentním magnetem, se v nepohyblivém vinutí statoru indukují střídavé napětí, úměrné otáčkám magnetu. Změnou rychlosti otáčení měníme napětí a také kmitočet, který je závislý na počtu pólových dvojic rotoru a počtu otáček za minutu. Díky tomu je možné pro měření počtu otáček využít měřič kmitočtu. Jelikož se kmitočet při změně magnetického pole, teploty nebo odporu nemění, na rozdíl od napětí, můžeme toho využít pro přesnější měření otáček[10].



Obrázek 2.6: Tachoalternátor s otočným magnetem.⁶

Tachoalternátor s bubínkovým rotorem se skládá ze dvou cívek, které jsou vůči sobě otočené o 90°. První cívka je trvale připojena na síť střídavého napětí a plní funkci budicího vinutí. Druhá cívka je snímací, zde dochází k indukci výstupního napětí a frekvenci, která je stejná jako frekvence budicího napětí. Toto napětí je při stálém buzení úměrné úhlové rychlosti rotoru. Tachoalternátory se používají k měření frekvence do 8000 ot/min[10].

⁶Převzato z: www.snimace.xf.cz/otackomery-mechanicke-indukcni.php



Obrázek 2.7: Tachoalternátor s bubínkovým rotorem.⁷

2.1.5 Impulsní otáčkoměry

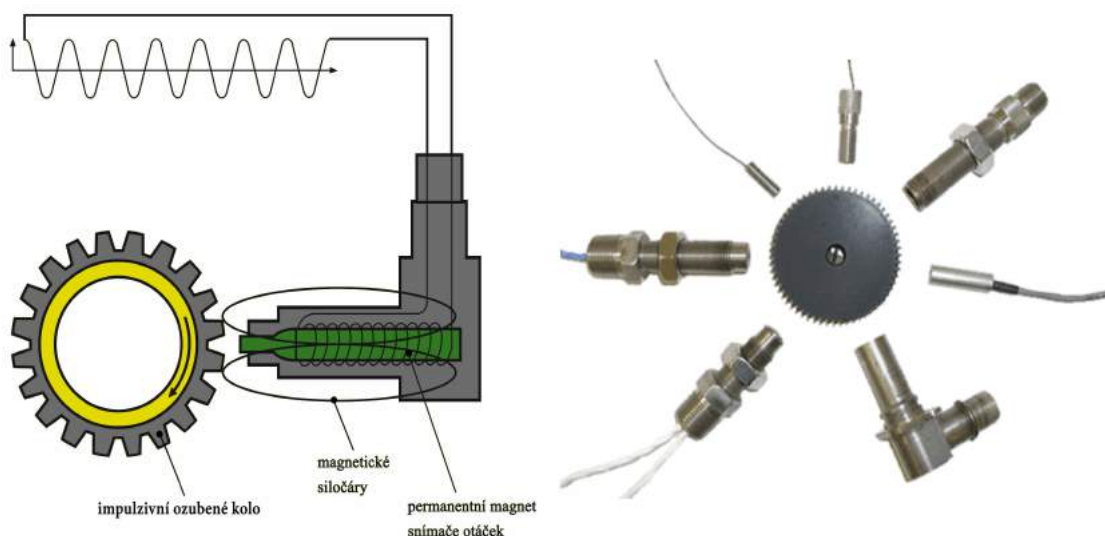
Impulsní otáčkoměry snímají značky na rotujícím objektu a vyhodnocují počet pulzů za jednotku času. K detekci polohy značky lze využít kontaktní snímač (např. jazýčkové relé), indukčnostní snímač, optoelektronický snímač a další. Obecně jsou impulsní otáčkoměry tvořeny snímacím kotoučem s dělením (zuby, otvory atd.), a z vlastního snímacího prvku, který snímá značky na kotouči a převádí je na elektrický signál. Výhodou snímačů je digitální výstupní signál a výsledná přesnost.

Indukční otáčkoměry

Indukční snímače, nebo také pick-up senzory, využívají pro měření indukční zákon. Vychází tedy ze skutečnosti, že v cívce, vystavené proměnnému magnetickému poli, se indukuje střídavé napětí. Snímač je tvořen cívkou, jejíž jádro je tvořeno permanentním magnetem. Při zasouvání a vysouvání magnetu do cívky a zpět se indukuje napětí, závislé na rychlosti pohybu, a úměrné časové změně magnetického toku.

V praxi se mohou použít dva způsoby konstrukce indukčního otáčkoměru. První možností je malý permanentní magnet, umístěný na rotující hřídeli v blízkosti pevného vinutí. Při rotaci se indukují napěťové impulsy, které jsou úměrné rychlosti otáčení. Ve druhém konstrukčním řešení je permanentní magnet součástí snímače a tvoří magnetický tok cívkou. Ke změně magnetického toku dojde přiblížením feromagnetického materiálu (kov) ve tvaru zubu, který rotuje okolo snímače. Zub se tedy střídavě přibližuje a oddaluje od snímače. Přiblížením zubu ke snímači dochází k uzavření magnetických siločar, a cívkou začne procházet velký magnetický tok. Při oddálení dojde k porušení siločar v důsledku velké vzduchové mezery mezi snímačem a obíhajícím zubem. Takto neustále se střídající magnetický tok vytváří indukované napětí v cívce. Indukované napětí je tím větší, čím větší jsou otáčky[10][15][11].

⁷Převzato z: www.snimace.xf.cz/otackomery-mechanicke-indukcni.php



Obrázek 2.8: Druhý typ indukčního snímače, vpravo ukázka různých provedení v praxi.⁸

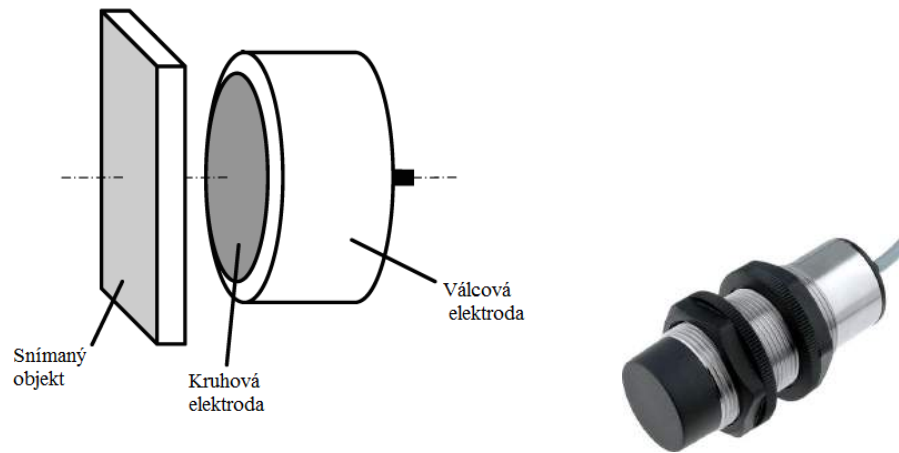
Indukční otáčkoměry jsou hojně využívány v automobilovém průmyslu. Frekvence otáčení může být měřena např. na klikové hřídeli motoru. Indukčního snímače využívají i systémy ABS. Každé kolo je vybaveno indukčním snímačem u brzdového kotouče, který předává informace řídicí jednotce. Pokud řídicí jednotka dostane signál, že je kolo blokováno, krátkodobě sníží tlak v brzdovém systému, a tím uvede kolo znovu do pohybu[10].

Kapacitní otáčkoměry

Kapacitní snímače umožňují bezkontaktně detekovat, popřípadě měřit, přiblížení předmětů na vzdálenost až několika milimetrů. Jejich hlavní výhodou je možnost snímat prakticky libovolný materiál. Montáž snímačů je obvykle shodná s indukčními senzory.

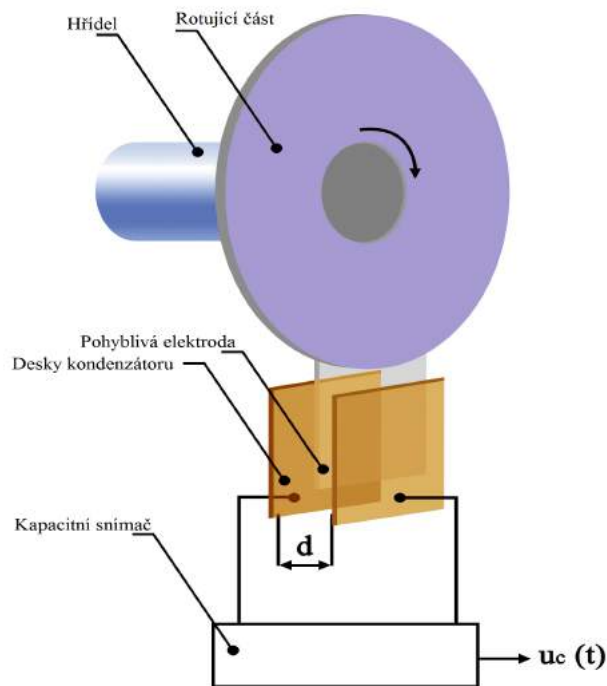
Kapacitní snímače vyhodnocují změnu permitivity prostředí vyvolanou předmětem, který vstoupí do elektrického pole kondenzátoru. To znamená, že kapacitní snímače mohou snímat nejen kovové, ale i nevodivé materiály, jejichž permitivity jsou dostatečně velké. Tímto se liší od indukčních snímačů, které jsou schopny reagovat pouze na elektricky vodivé materiály. V kapacitních senzorech je jako snímací prvek používán „otevřený“ kondenzátor. Ten se skládá ze dvou elektrod ve tvaru destiček, a dielektrika s nevodivým nebo mírně vodivým mediánem mezi nimi. První kruhová elektroda tvoří hlavní aktivní prvek kapacitního snímače. Druhá elektroda tvoří válcové pouzdro, které funguje zároveň i jako stínění. Takto vytvořený kondenzátor má svou základní kapacitu, která se mění se změnou dielektrika – přiblížením snímaného objektu[15][11].

⁸Převzato z: www.autolexicon.net/obr_clanky/cs_abs_002.jpg



Obrázek 2.9: Kapacitní snímač, vpravo ukázka provedení v praxi.⁹

Pro měření otáček pomocí kapacitního snímače lze zkonstruovat následující systém. Pohyblivá elektroda je připojena k měřené části a při otáčení hřídele se vsune mezi desky měřícího kondenzátoru, čím se změní permitivita dielektrika a kondenzátor změní svou kapacitu. Tato změna se projeví jako jeden impuls. Při otáčení snímač kapacity generuje impulsy, ze kterých lze vyvodit frekvenci otáčení^{[15][11]}.



Obrázek 2.10: Kapacitní otáčkoměr.¹⁰

⁹Převzato z: www.kam.sjf.stuba.sk/katedra/publikacie/leonardo/ucebnica/16s.pdf

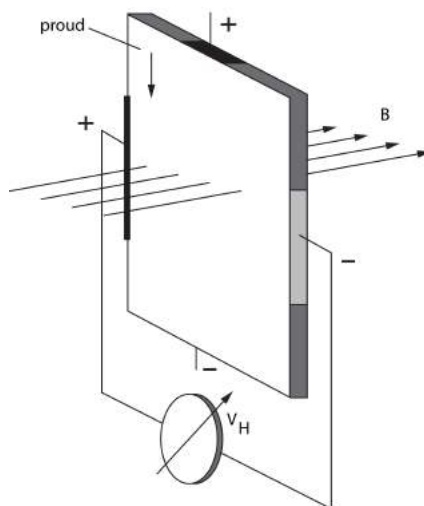
¹⁰Převzato z: [//vlm.sjf.stuba.sk/podstranky/otacanie%20impulz.html](http://vlm.sjf.stuba.sk/podstranky/otacanie%20impulz.html)

Magnetické otáčkoměry

Magnetické snímače vytvářejí elektrický impuls na základě Hallova jevu. Tento fyzikální jev byl objeven v roce 1879. Hallův jev byl zpočátku sice omezený, ale velmi cenný nástroj pro studování elektrické vodivosti v kovech, polovodičích a dalších vodivých materiálech. Hallovy senzory jsou dnes používány k detekci magnetického pole, polohy, nebo třeba rozmístění objektů.

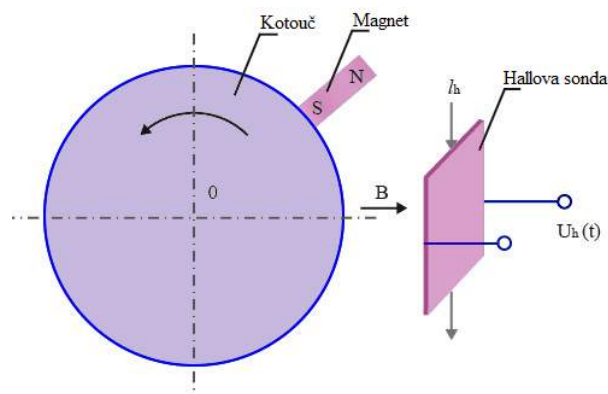
Prochází-li v jedné ose polovodičového plátku proud, a kolmo na rovinu působí magnetické pole, vzniká v kolmém směru na směr proudu Hallovo napětí, které je úměrné velikosti magnetického pole.

Hallův snímač (sonda) je elektronická součástka, jejíž činnost je založena na výše popsaném jevu. Sonda je tvořena úzkou polovodičovou destičkou, skrz kterou prochází proud. Vložíme-li destičku do magnetického pole, které působí kolmo na protékající proud v destičce, začne skrz destičku proudit indukční tok, a tímto je generováno Hallovo napětí. Velikost získaného napětí není však příliš velká, a proto se před vyhodnocením ještě zesiluje[8].



Obrázek 2.11: Princip Hallova jevu[8].

Magnetický otáčkoměr lze zkonstruovat podobně, jako tomu bylo u indukčních otáčkoměrů. Na rotující hřídel je upevněn permanentní magnet, který se otáčí spolu s ní. Hallův snímač je tvořen destičkou připojenou na zdroj konstantního proudu. Magnet se pohybuje v blízkosti snímače a působí na něj svým magnetickým polem. Při rotaci hřídele dochází k vytváření impulsů Hallova napětí[8].



Obrázek 2.12: Magnetický snímač otáček využívající Hallův jev. ¹¹

2.1.6 Fotoelektrické otáčkoměry

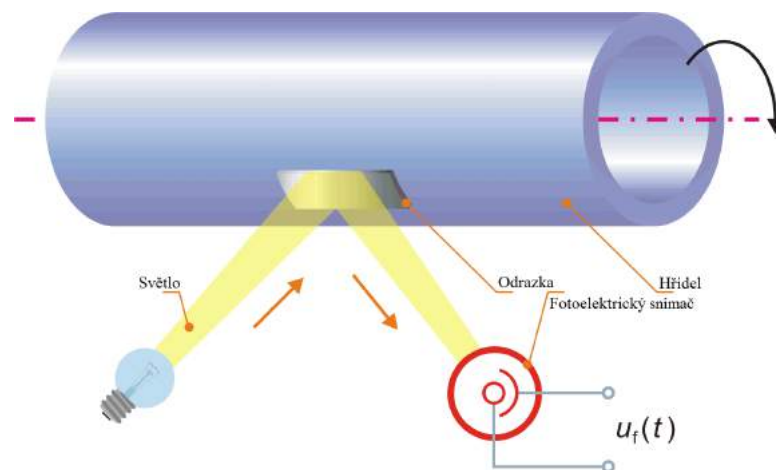
Fotoelektrické (optické) snímače pracují na principu detekce vráceného světelného paprsku, emitovaného vlastním snímačem nebo vysílačem. Zdrojem světla je buď infračervená či červená LED dioda nebo laserový paprsek. Světlo je vždy pulzně modulované pro eliminaci vlivu okolního světla. Fotoelektrické snímače sestávají z dvojice přijímač a vysílač světelného signálu. Vysílač emituje nepřetržitě světelný paprsek a přijímač tento paprsek zachycuje, přičemž dochází k fotoelektrickému jevu, kdy dopadem světla na fotoelektrický materiál dochází ke změně světelné energie na elektrickou. To se projeví jako změna elektrických vlastností nebo vodivosti či změna napětí. Jako detektor světla se nejčastěji používá fotodiody nebo fototranzistor. Vzhledem k uspořádání můžeme fotoelektrické otáčkoměry dělit do dvou skupin:

- s modulací optické cesty (Reflexní závory)
- s přerušením optické cesty (Jednocestné závory)

Reflexní závory

Reflexní závory vyzářují světlo, které se po odrazu od předmětu (lesklého) nebo odrazky vrací zpět. K aktivování výstupu tak dochází při přerušování optické dráhy objektem, který světlo neodráží. Tohoto principu lze pro měření otáček využít umístěním speciální odrazky na rotující součást. Rotující odrazka v každém průchodu odrazí paprsek nazpět. Vrácený paprsek je zachycen fotocitlivým prvkem snímače, a světelné impulsy jsou převedeny na impulsy elektrické[15][11].

¹¹Převzato z: [//vlm.sjf.stuba.sk/podstranky/otacanie%20impulz.html](http://vlm.sjf.stuba.sk/podstranky/otacanie%20impulz.html)



Obrázek 2.13: Otáčkoměr na principu reflexní závory.¹²

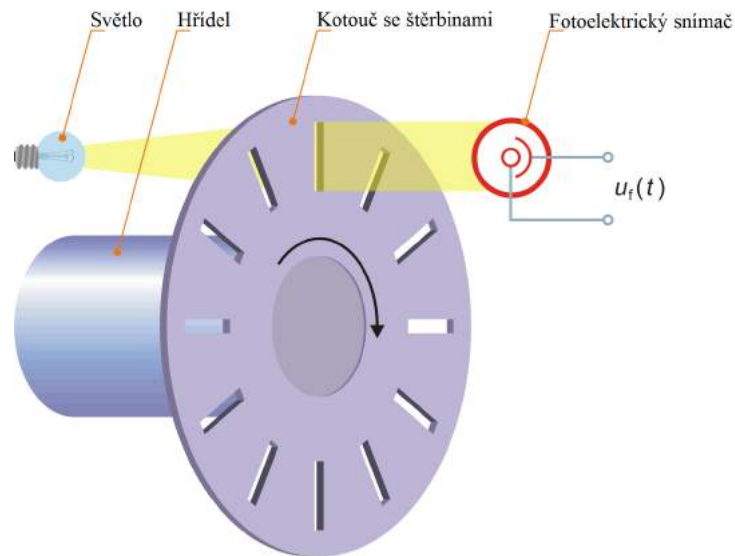
Výhodou optických snímačů je umožnění konstrukce stále menších snímačů s vysokou rozlišovací schopností, limitovanou jevy při ohybu světla. Další výhodou je vysoká necitlivost vůči elektromagnetickým polím a galvanickému oddělení snímače a objektu[15][11].

Jednocestné závory

Jednocestné závory pracují na velmi jednoduchém principu snímání předmětu. Světlo, které vysílá zdroj, se dostává k přijímači. Pokud se jeho dráha přeruší nějakým objektem, výstup senzoru se aktivuje. Pro správnou funkci musí být splněno několik podmínek. Přijímač a vysílač musejí být správně seřizeny bezpodmínečně v ose proti sobě. V cestě mezi přijímačem a vysílačem nesmí být žádná jiná překážka než snímáný předmět. Emitované světlo musí být modulováno. Splněním těchto podmínek můžeme docílit snímání předmětů až na vzdálenost 200 metrů.

V otáčkoměrech založených na jednocestných závorách je rotující hřídel spojená s kotoučem, ve kterém je jedna nebo více štěrbin. Při otáčení kotouče se fotoelektrický snímač střídavě osvětluje záblesky ze světelného zdroje. Na výstupu snímače získáváme elektrický signál ve tvaru pulsů. Frekvenci otáčení lze určit pomocí čítače pulsů[15][11].

¹²Převzato z: vlm.sjf.stuba.sk/podstranky/otacanie%20impulz.html



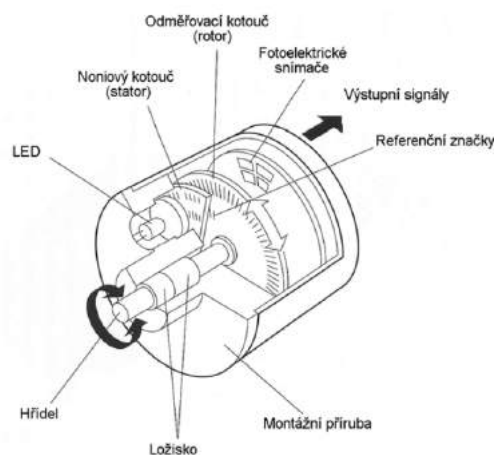
Obrázek 2.14: Otáčkoměr na principu jednocestné závory.¹³

Inkrementální snímače

Inkrementální snímače, někdy též označovány jako rotační enkodéry, jsou elektromechanická zařízení, která slouží ke snímání polohy, rychlosti nebo měření otáček motorů. Název inkrementálních snímačů je odvozen od jejich výstupu, což je sled impulsů inkrementujících obsah čítače. Princip těchto snímačů je založen na jednocestné závore, tedy ve clonění světelného toku mezi zdrojem světla a fotodetektorem. Celý systém sestává ze dvou rotačních disků, zdroje světla a fotodetektoru.

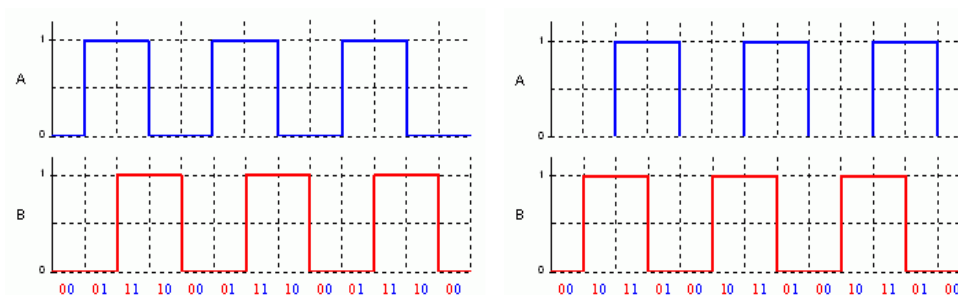
První odměřovací disk (rotor) je spojen s hřídelem a obsahuje střídající se světlopropustné a světlonepropustné úseky. Druhý disk (stator) je zkonstruován stejným způsobem, je umístěn před nebo za prvním diskem, a navzájem tvoří soustavu clonek. Štěrby na obou discích při rotaci přerušují světelný paprsek, přičemž na fotodetektoru vznikají elektrické impulsy. Důležitým faktorem je pootočení obou disků navzájem o 90° , tedy o $1/4$ periody. Otáčením hřídele se vytváří dva signály obdélníkového průběhu. Zpracováním prvního signálu je možné získat informaci o rychlosti otáčení, zpracováním druhého signálu získáme údaj o směru otáčení. Abychom byli schopni určit zároveň i absolutní polohu, musíme přidat třetí referenční signál (značku), který bude indikovat počátek soustavy. Nevýhoda takto vytvořených snímačů spočívá v tom, že změna obsahu čítače rušivými signály je korigovatelná až po dosažení referenční značky^{[15][11]}.

¹³Převzato z: vlm.sjf.stuba.sk/podstranky/otacanie%20impulz.html



Obrázek 2.15: Schéma inkrementálního snímače.¹⁴

Směr otáčení lze u inkrementálních snímačů určit na základě vzájemného posunutí signálů získaných při generování pulsů.



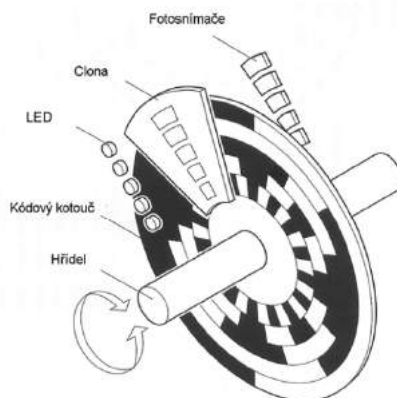
Obrázek 2.16: Vlevo směr otáčení po směru hodinových ručiček, vpravo proti směru.¹⁵

Absolutní snímače

Absolutní snímače získáme, pokud v inkrementálním čidle místo kotouče s ryskami použijeme tzv. kódovací kotouč, na němž jsou průhledná a neprůhledná políčka v několika řadách, a každá řada má svůj snímač. Každé řadě pak odpovídá jeden bit výstupního slova. Použitím kotoučů s různými kódy, můžeme získat výstupní slovo v binárním, Grayově nebo jiném kódu. Grayovo kódování se často využívá, jelikož při přechodu z jednoho stavu do druhého, se mění stav informace pouze v jednom bitu. Sousední polohy se tedy liší pouze v jednom bitu. Díky odlišnému způsobu kódování a rozdílnému výstupnímu signálu jsou absolutní snímače schopné zaznamenávat správné údaje i po ztrátě napájecího napětí[10].

¹⁴Převzato z: docplayer.cz/docs-images/17/166871/images/12-0.png

¹⁵Převzato z: robotika.vosrk.cz/guide/sensors/decode/cs



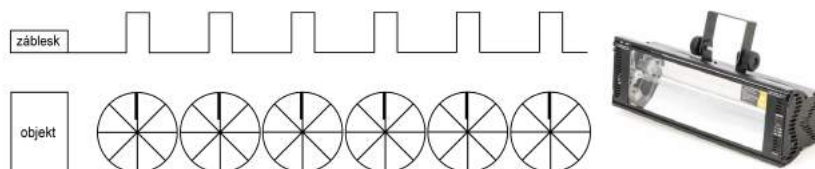
Obrázek 2.17: Absolutní snímač polohy.¹⁶

2.1.7 Stroboskopické otáčkoměry

Stroboskopické otáčkoměry jsou bezkontaktní přístroje, které k měření využívají stroboskopického jevu. K tomuto jevu dochází v případě, kdy reprezentujeme spojitý periodický pohyb (otáčení) diskretními, nespojitými vzorky (záblesky). Stroboskopický jev můžeme pozorovat, pokud je spojitý periodický děj zprostředkován prostřednictvím konečného a vhodného počtu vzorků tak, že frekvence pozorovaného pohybu se přiblíží frekvenci vzorkovací, nebo jejímu násobku[10].

Stroboskop

Hlavní částí stroboskopu je stroboskopická výbojka (halogenová), která vysílá intenzivní, regulovatelné, velmi krátké záblesky přerušovaného světla, díky zdroji o proměnném kmitočtu. Snímaná součástka je vystavena zábleskům, přičemž jejich frekvence se postupně zvyšuje, a v důsledku toho se rotující část začne opticky „zpomalovat“. Zpomalování i frekvence záblesků roste, dokud nedojde ke stroboskopickému jevu, kdy se frekvence výbojky a rotující části vyrovnají. V tomto stavu se díky setrvačnosti lidského oka součástka jeví jako nehybná. Umístíme-li na snímanou část značku, bude při stroboskopickém jevu zůstat na stejném místě. V případě, že frekvence záblesků je větší než frekvence rotující části, bude se zdát, že se značka pohybuje ve směru otáčení. Na druhou stranu, pokud frekvence výbojky bude menší, značka se bude pohybovat proti směru otáčení. Jelikož ke znehybnění dochází s každým celočíselným násobkem vzorkovací frekvence, lze pomocí stroboskopů, kromě frekvence otáčení, měřit i fázi a zpoždění[10].



Obrázek 2.18: Princip stroboskopického jevu, vpravo provedení v praxi.¹⁷

¹⁶Převzato z: images.slideplayer.cz/41/11198611/slides/slide_10.jpg

¹⁷Převzato z: www.lastra.si/9027-home_default/stroboskop-stairville-1500-dmx.jpg

2.2 Měření frekvence

Díky nepřebernému množství využití je měření frekvence důležitou součástí mnoha technických odvětví. Od jednoduchých měřících obvodů po složitější systémy, jako například tachometry v autech, nebo měřiče otáček motorů. Tato část bude věnována popisu několika metod pro měření frekvence.

$$f = \frac{1}{T} \quad (2.1)$$

Frekvence a čas jsou dvě navzájem svázané veličiny, jelikož frekvence je definována jako počet opakování za jednotku času a je převrácenou hodnotou doby periody 2.1. Pro analogové nebo digitální křivky lze tedy převrácením periody signálu získat danou frekvenci. Čím kratší je doba periody, tím větší je frekvence, a naopak. Ve fyzice lze periodicitu pozorovat u rotací, kmitů nebo vlnění. Fyzikální jednotkou frekvence je Hertz.

Metody pro měření frekvence lze rozdělit do dvou skupin, na analogové a digitální. Analogové metody jsou dnes na ústupu a jsou postupně nahrazovány metodami digitálními. Přesto však pro ně lze najít uplatnění, a to díky nízké ceně a absenci napájecího napětí u některých metod. Dále se však budeme zabývat pouze digitálními metodami pro měření frekvence[9].

2.2.1 Digitální metody

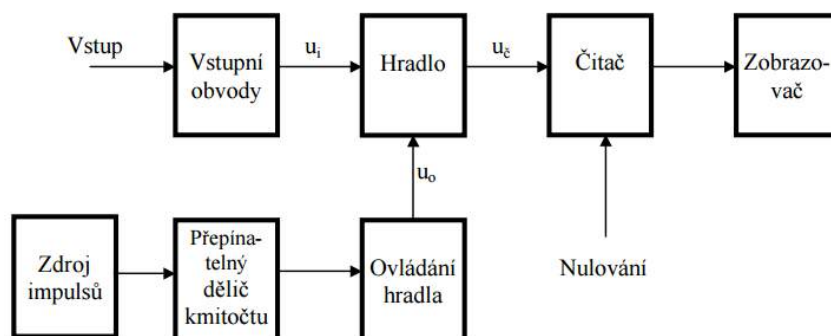
S digitálním měřením frekvence se setkáváme velmi často. Je to z důvodů pohodlného měření, vysoké přesnosti a stále nižší ceně měřících přístrojů. Tyto přístroje jsou stále zdokonalovány, přičemž cílem je dosáhnout co největšího frekvenčního rozsahu, vysoké přesnosti při malé době měření, a snadné obsluze. I když má číslicové měření velkou řadu výhod, může u zkreslených nebo zašuměných signálů vést ke zcela chybným výsledkům. Je proto vhodné zkontrolovat výsledky měření, například osciloskopem[9][7].

Přímé měření frekvence

Přímé číslicové měření kmitočtu vychází z toho, že kmitočet je převrácená hodnota periody. Metoda je duální k měření periody. Nejběžnějšími přístroji pro přímé měření jsou čítače. Čítáním se určuje počet period měřeného signálu, které spadají do měřícího intervalu známé délky.

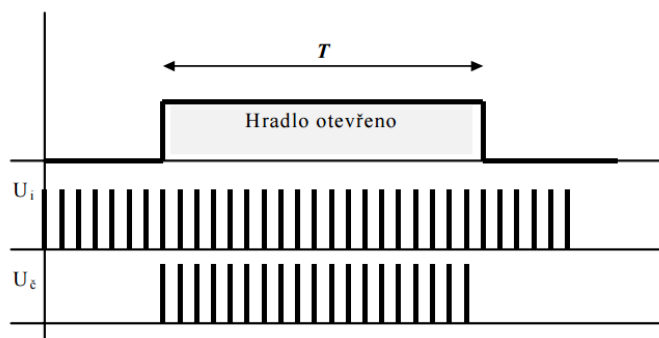
$$f = \frac{N}{T} \quad (2.2)$$

V rovnici 2.2 udává proměnná N počet period (impulsů). U jednodušších přístrojů je hodnota periody celistvý násobek 1s a dělení se řeší jen posunem desetinné tečky na zobrazovacím displeji. U přístrojů složitějších lze interval periody spojitě měnit, a přístroj realizuje dělení[7].



Obrázek 2.19: Blokové schéma číslicového měřiče otáček.

U číslicových měřičů frekvence je úroveň trvale nastavena na nulu. Někdy se vstupní citlivost reguluje automaticky podle špičkové hodnoty vstupního napětí. Volba hrany u měření kmitočtu je nepodstatná. Na rozdíl od měřiče časového intervalu však každý impuls ze vstupních obvodů odpovídá jedné periodě vstupního signálu u_i a přichází na vstup hradla. Hradlo je řízeno obvody ovládání hradla a je otevíráno po dobu T , která je vymezena zdrojem impulsů a následným děličem kmitočtů. Zdroj impulsů je řešen jako krystalem řízený oscilátor. Dělič kmitočtu je přepínatelný v dekadických násobcích, vymezuje intervaly např. 10s-1s-0, 1s-10ms-1ms... a určuje rozsah měření. Současně se změnou rozsahu se posouvá i desetinná tečka na numerickém zobrazovači, případně se mění řád jednotky[7].



Obrázek 2.20: Průběh napětí na digitálním měřiči kmitočtu.

Průběhy napětí v číslicovém měřiči kmitočtu jsou uvedeny na Obrázku 2.20. Pokud je doba T proměnná (volitelná spojitě obsluhou), musí se nezávisle tento interval číslicově měřit, aby ho bylo možno použít při výpočtu frekvence.

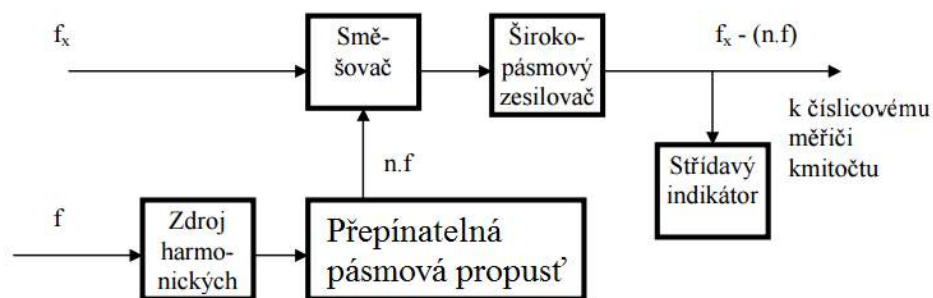
Popsaným způsobem nelze měřit kmitočty vyšší, než je mezní kmitočet čítače. Při jeho překročení již čítač některé pulsy uč nezaregistruje, a přístroj udává nepřesné výsledky. Dále je nutno si uvědomit, že tento přístroj udává průměrnou frekvenci za dobu T [7].

Nepřímé měření frekvence

Nepřímé metody číslicového měření mají za cíl odstranit nevýhody přímého číslicového měření frekvence. Při měření nízkých kmitočtů je výhodné, místo měření frekvence, číslicově měřit periodu signálu a výpočtem převést na frekvenci. Měření pak trvá jen dobu periody. Některé přístroje automaticky při měření nízkých frekvencí volí tuto metodu. Měření kmi-

točtů vyšších, než je mezní kmitočet čítače, lze taky provádět nepřímou. Jednou z možností je předřazení rychlého děliče kmitočtu na principu čítače. Ten samozřejmě obsahuje i potřebné rychlé vstupní obvody. Takový předdělič je v některých přístrojích vestavěn, a má i samostatný vstup. Tímto způsobem navíc není změněna relativní přesnost měření.

Další možností nepřímého měření spočívá v použití směšování k přeměně kmitočtu. Jde o doplněk, označován jako kmitočtový měnič[7].



Obrázek 2.21: Blokové schéma kmitočtového měniče.

Měření signálu o kmitočtu f_x je po případném zesílení přiveden do směšovače, kde vznikne rozdíl $f_x - (n.f)$, který se měří číslicovým měřičem kmitočtu. Signál o kmitočtu $n.f$ se získá filtrací ze zdroje vyšších harmonických pulsů, na jehož vstupu je signál o kmitočtu f . Volba n se děje přepínáním pásmové propusti za současného sledování velikosti střídavého napětí na výstupu měniče pomocí střídavého indikátoru. Ten má frekvenční rozsah přibližně takový, jako následující číslicový měřič kmitočtu. Kmitočtový dělič bývá řešen jako zásuvný modul do jednotky číslicového měřiče kmitočtu, a při jeho použití se na displeji přístroje odečítá skutečný vstupní kmitočet f_x . Tímto způsobem lze měřit kmitočet i přes 20 GHz[7].

2.3 Akcelerometry

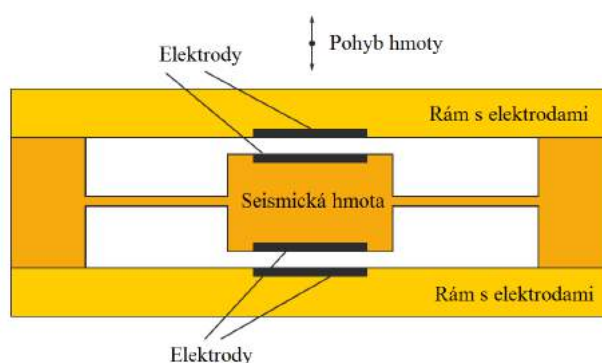
Zrychlení je dynamická vlastnost těles, jelikož dle 2. Newtonova zákona vyžaduje působení určité síly, například gravitační. Nehybné těleso žádné působení síly nevyžaduje, stejně tak jako změna jeho polohy, která závisí na rychlosti, pokud neexistuje síla opačná, jako například tření. Změna polohy, rychlost a zrychlení jsou mezi sebou propojené, což dokazuje i vztah, kdy zrychlení lze určit jako derivaci rychlosti podle času. Zrychlení je vektorová fyzikální veličina, která se běžně udává v násobcích tíhového zrychlení. Jednotka tíhového zrychlení je g (Země – cca 9,81 m/s).

Akcelerometry jsou elektrická zařízení, která umožňují měřit velikost zrychlení. Pomocí akcelerometrů lze měřit dynamické zrychlení, což je síla vytvořená změnou rychlosti pohybujícího se tělesa, nebo zrychlení statické, kdy se jedná o sílu vzniklou působením gravitace. Obecně akcelerometry pomáhají určitému systému lépe chápat své okolí. Při měření dynamických zrychlení můžeme například analyzovat směr pohybu. Toho je využíváno třeba v informatice. Laptopy dnes obsahují akcelerometry, které chrání harddisk před poškozením. V případě upuštění notebooku akcelerometr zaregistruje náhlý volný pád a vypne pevný disk. Čtecí hlavy tedy nenarazí na plotny disku. Dle vnitřní struktury můžeme akcelerometry dělit na kapacitní, piezoelektrické, piezorezistivní a tepelné[8][14].

2.3.1 Kapacitní akcelerometry

Princip kapacitních akcelerometrů je založen na detekci pohybu seismické hmoty, která zaostává za pohybem pouzdra, ve kterém je uložena, a které je spojeno se zkoumaným objektem. Na tomto základu jsou kapacitní akcelerometry využívány jako snímače posuvu, které jsou schopny měřit mikroskopické pohyby při silných vibracích nebo lineárním zrychlování.

Kapacitní snímač je tvořen minimálně ze dvou elektrod. První elektroda je napevno připojena k pouzdru akcelerometru, a druhá elektroda je přichycena na seismické hmotě, která se v pouzdru volně pohybuje. Tyto dvě elektrody tvoří kondenzátor, jehož hodnota se odvíjí právě od velikosti zrychlení. Zlepšením rozlišovací schopnosti kapacitních akcelerometrů je dosaženo vytvořením druhého kondenzátoru v téže struktuře. Při pohybu hmoty dochází ke zvětšování jedné kapacity, a spolu s tím hodnota druhé kapacity klesá. Rozdíl obou kapacit je následně vyhodnocen[8].



Obrázek 2.22: Schéma kapacitního akcelerometru.¹⁸

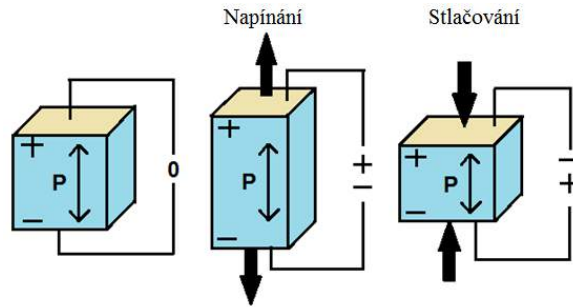
Kapacitní akcelerometry jsou nejčastěji konstruovány pomocí MEMS (Micro Electro Mechanical Systems) technologie. Technologie je založená na umístění elektronických i mikro mechanických prvků na křemíkové bázi. Mechanické prvky mají za úkol převést mechanický podmět na elektrickou hodnotu. Elektrické integrované obvody provádí řízení a vyhodnocení.

Kapacitní akcelerometry mají vysokou přesnost i citlivost, jsou ovšem vhodné jen pro nízké rozsahy zrychlení[14].

2.3.2 Piezoelektrické akcelerometry

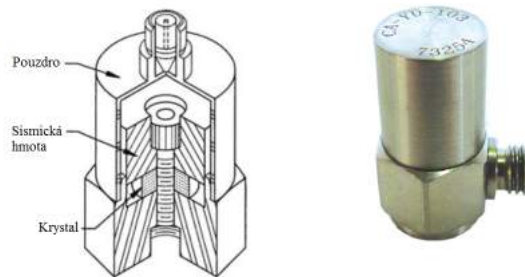
Piezoelektrické akcelerometry jsou založeny na piezoelektrickém jevu. Tento jev má přirozené uplatnění ve snímání vibrací a zrychlení. Piezoelektrický jev vzniká při deformaci piezoelektrických krystalů, což jsou krystaly, které nemají střed symetrie. Nejznámějším takovým krystalem je křemen. Začne-li na krystal působit deformační síla, dojde k vychýlení opačných nábojů v krystalické mřížce. Elektrická těžiště kladných i záporných iontů v krystalu se od sebe vzdálí a na určitých plochách krystalu se objeví elektrický náboj. Výsledkem piezoelektrického jevu je přeměna mechanické energie na energii elektrickou, v krystalických materiálech, složených z elektrických dipólů[8][14].

¹⁸Převzato z: automatizace.hw.cz/mereni-a-regulace/mereni-vibraci-pro-diagnostics-opotrebeni-stroju.html



Obrázek 2.23: Podstata piezoelektrického jevu. ¹⁹

Piezoelektrické akcelerometry jsou tvořeny seismickou hmotou a piezoelektrickým materiálem. Při vystavení snímače zrychlení (vibracím, otřesům atd.) je seismickou hmotou generována síla, která působí na piezoelektrický materiál. Působením síly dochází k piezoelektrickému jevu a generování elektrického náboje, který je úměrný působící síle[8].



Obrázek 2.24: Schéma piezoelektrického snímače, napravo provedení v praxi[8][14].

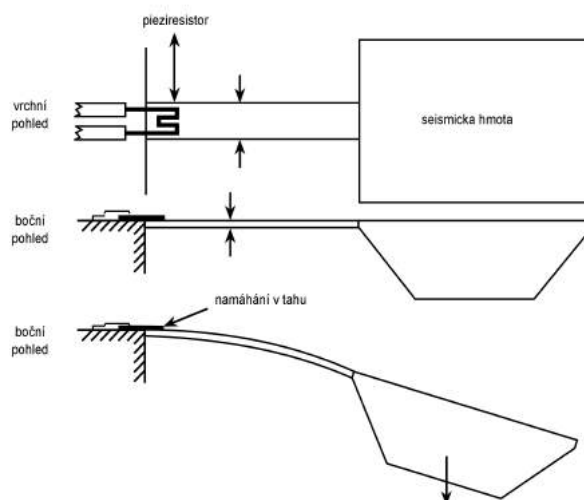
Piezoelektrické akcelerometry pracují na frekvenci od 2 Hz až do 5 kHz, nelze je tedy použít pro měření statického zrychlení. Dále disponují dobrým potlačením mimošného šumu, vysokou linearitou a širokým rozsahem provozních teplot (až do 120°C)[8].

2.3.3 Piezorezistivní akcelerometry

Piezorezistivní akcelerometry využívají piezorezistivního jevu. Tento jev je obdobně jako piezoelektrický založen na deformaci materiálu. V předešlé kapitole vyvolávala deformace krystalové mřížky elektrické napětí. U piezorezistivních snímačů vyvolává působení vnější síly na krystalickou mřížku změnu měrného elektrického odporu.

Senzory obsahují namísto piezoelektrického materiálu piezorezistivní, který převádí sílu způsobenou pohybem seismické hmoty na změnu odporu. Vlivem zrychlení dochází k působení síly, která změní (ohne) polohu nosníku z piezorezistivního materiálu. Tímto ohybem se mění odpor, a na základě této změny jsme schopni určit velikost působeného zrychlení. Pro odstranění negativních vlivů se v integrovaných piezorezistivních akcelerometrech využívá zapojení do Wheatstonova můstku[8].

¹⁹Převzato z: www.intechopen.com/source/html/39166/media/image4.png



Obrázek 2.25: Princip piezorezistivního snímače.²⁰

Výhodou piezorezistivních akcelerometrů je schopnost měřit i konstantní zrychlení (0 až 13 kHz). Nároky, kladené na dnešní piezorezistivní křemíkové akcelerometry, jsou především malá velikost, stabilní výstup přes celý rozsah provozních teplot, velká přesnost, vysoká spolehlivost, snadná integrace senzoru a potřebné elektroniky, a nízká cena[8].

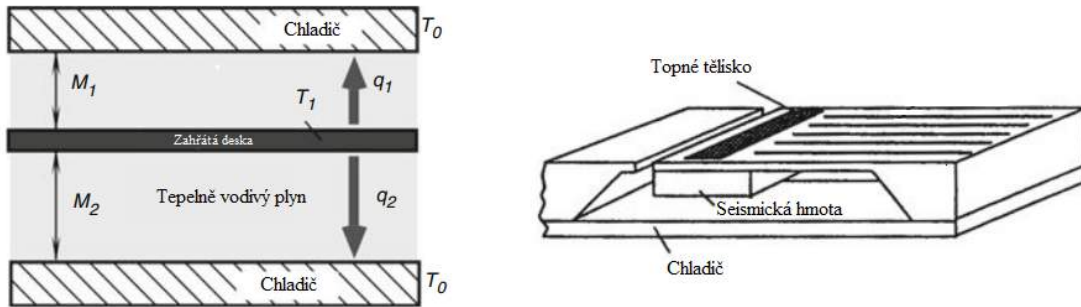
2.3.4 Tepelné akcelerometry

Principem tepelných akcelerometrů je šíření tepla určitou látkou, a snímání rozložení teploty v okolí zdroje tepla. Senzory můžeme rozdělit podle látky, kterou zahříváme, na akcelerometry s vyhřívanou deskou nebo zahřátým plynem.

Vyhřívaná deska

Akcelerometry, využívající techniku zahřáté desky stejně tak, jako všechny výše popsané akcelerometry, obsahují seismickou hmotu. Hmota je podepřena tenkým nosníkem a umístěna v blízkosti chladiče nebo mezi dvěma chladiči. Prostor mezi seismickou hmotou a chladičem je vyplněn tepelně vodivým plynem. Hmota je zahřívána topným tělískem na předem stanovenou teplotu. Ve stavu, kdy nepůsobí žádné zrychlení, je mezi hmotou a chladičem vytvořen tepelně rovnovážný stav. Množství tepla odvedeného chladičem je funkcí vzdálenosti mezi seismickou hmotou a chladičem[8].

²⁰Převzato z: www.ece.montana.edu/seniordesign/archive/FL06/mems_mask/



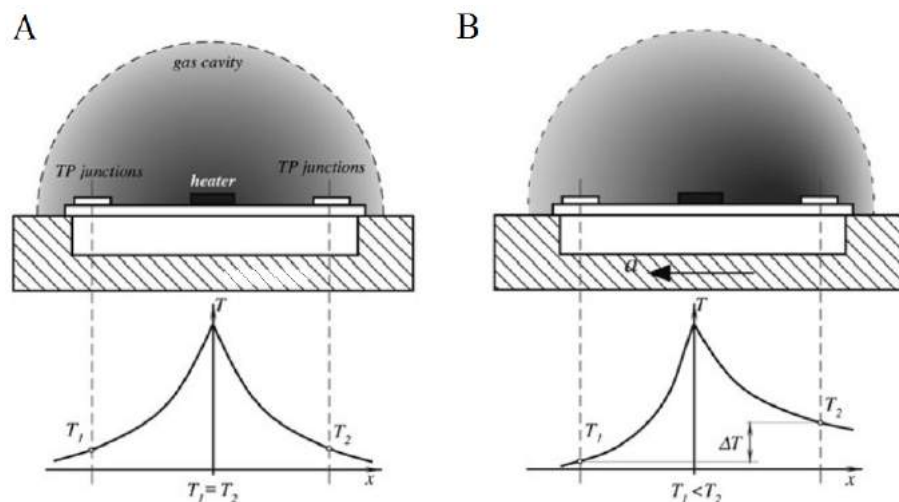
Obrázek 2.26: Tepelný akcelerometr s vyhřívanou deskou M_1 , M_2 – vzdálenost hmoty a chladiče, q_1 , q_2 – vyzářené teplo, T_1 , T_0 - teplota[8].

Snímaná teplota je převedena na elektrický signál, ze kterého lze vyvodit hodnotu zrychlení. Citlivost akcelerometrů je menší než u kapacitních nebo piezoelektrických, nicméně jsou více náchylné na vlivy okolní teploty, a na elektrostatický nebo elektromagnetický šum[8].

Zahřátý plyn

Hlavní rozdíl mezi těmito akcelerometry je ve využití zahřátého plynu namísto seismické hmoty. Tento typ tepelného snímače byl vytvořen společností MEMSIC. Technologie je založena na CMOS architektuře a jedná se o dvouosý měřicí systém. Principem je přenos tepla na základě vynuceného tepelného proudění. Proudění může být přirozené, vyvolané například gravitací, nebo zapříčiněné vnější silou, například ventilátorem. U akcelerometrů se zahřátým plynem je takováto síla vyvolána právě zrychlením. Sensor měří vnitřní změny šíření tepla v komoře s uzavřeným plynem. Funkčně se jedná o ekvivalent k akcelerometrům obsahujícím seismickou hmotu. Jednou z největších výhod, při použití plynu místo pevné hmoty, je odolnost vůči šokovému zatížení (až do 50,000 g), což vede ke snížení poruchovosti.

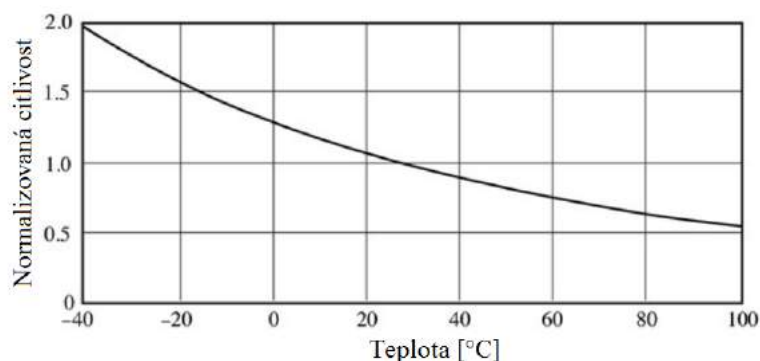
Snímač je tvořen topným tělískem, komorou naplněnou plynem, a čtyřmi rovnoměrně rozmístěnými teplotními senzory, zapojenými sériově. Teplotní senzory jsou rozloženy do všech čtyřech stran okolo topného tělíska, čímž je zajištěno snímání ve dvou osách. Při nulovém zrychlení je teplotní rozložení plynu v komoře symetrické kolem topného tělíska, takže teplota je na všech senzorech stejná. Výstupem je tedy nulové napětí. Typicky je tělísko zahříváno na hodnotu kolem 200°C. Plyn dosahuje největší teploty přímo u tělíska, ale rapidně se ochlazuje směrem k levým a pravým teplotním sensorům. Pokud na plyn žádná síla nepůsobí, má teplota v komoře symetrické kuželovité rozložení, kde jsou teploty na všech senzorech stejné (Obrázek 2.27 A)[8].



Obrázek 2.27: Teplotní rozložení v tepelném akcelerometru (A - klid, B - zrychlení) T_1 , T_2 , T - teploty, a - zrychlení[8].

Zrychlením dochází k tepelnému proudění, které způsobí asymetrické teplotní rozložení (Obrázek 2.27 B). Na obrázku lze vidět působení zrychlení, které způsobí posun molekul plynu k pravému teplotnímu snímači. Zahřátý plyn předá senzoru tepelnou energii, která je převedena na elektrickou. Zároveň nastane i úbytek napětí na levém teplotním senzoru, a rozdíl obou napětí bude přímo úměrný působenému zrychlení.

Akcelerometry se zahřátým plynem jsou schopny měřit zrychlení od 1 g až do 100 g. Jsou tedy schopny měřit jak dynamické (vibrace), tak statické (gravitace) zrychlení. Citlivost akcelerometrů opět závisí na okolní teplotě, je tedy nutné tyto jevy kompenzovat[8].



Obrázek 2.28: Ukázka změny citlivosti tepelných akcelerometrů v závislosti na okolní teplotě[8].

2.4 Mobilní zařízení

Mezi dnes nejpoužívanější operační systémy pro mobilní telefony patří Android od společnosti Google a iOS od firmy Apple. Celosvětově dnes na trhu s 65,19% převládá systém Android. Výhodou Androidu je například otevřenost systému, velké množství bezplatných aplikací, uživatelská flexibilita atd. Nevýhodou je fragmentace systému. Mobilní telefony dnes čím dál více připomínají stolní počítače, což lze nejlépe vidět na parametrech vnitř-

ního hardwaru. Velkou výhodou mobilních telefonů je široká komunita vývojářů. Vzniká spousta nových aplikací na synchronizaci dat, sociální sítě, jízdní řády atd. Právě díky počtu aplikací, které jsou "skoro na vše", mají mobilní zařízení větší míru využití než pouze na telefonování a SMS zprávy. V následující části textu bude popsán dnes nejrozšířenější mobilní systém Android.

2.4.1 Platforma Android

Android je open-source mobilní operační systém založený na Linuxovém jádře. Primárně byl vyvíjen pro mobilní telefony a tablety, ale dnes je rozšířen ve spoustě dalších zařízeních jako televize, routery nebo herní konzole. V roce 2005 byla společnost koupena firmou Google a od té doby se dále rozvíjí. Nejnovější verzí systému je dnes Android Nougat.

Architekturu systému lze rozdělit do následující pěti vrstev[2]:

- Základem platformy Android je upravené Linuxové jádro. Jádro pracuje na nejnižší vrstvě a tvoří přechod mezi hardwarem a softwarem. Z tohoto důvodu obsahuje ovladače k periferiím, jako například Bluetooth, USB, WiFi nebo fotoaparát. Dále se jádro stará o správu paměti, plánování procesů, bezpečnostní funkce a řízení spotřeby.
- Další vrstvu tvoří abstraktní hardwarová část. Zde jsou jednotlivá hardwarová rozhraní zpřístupněná vyššímu Java API. Vrstva obsahuje moduly knihoven, kde každá implementuje rozdílné rozhraní pro specifický typ hardwaru. Pokud vyšší vrstva vyžaduje určitý hardware, je jí pro práci s tímto rozhraním zaslána patřičná knihovna. K výběru je zde například modul pro Bluetooth nebo audio vstup.
- O úroveň výš se nacházejí nativní knihovny, napsané v jazyce C nebo C++. Těchto knihoven využívá řada základních služeb, jako například předešle popsaná vrstva. Mezi knihovny patří Libc, WebKit nebo OpenGL a Android nabízí přístup k těmto knihovnám i aplikacím. Na stejné úrovni se nachází i Android Runtime (ART), který je dostupný pro zařízení od verze 5.0 a výš. Jedná se o nový virtuální stroj, vytvářející běhové prostředí pro aplikace napsané v Javě. Používá včasnou (just-in-time) a dopřednou (ahead-of-time) kompilaci, optimalizuje garbage collector a přidává lepší debugovací prostředky.
- Předposlední vrstvu tvoří Java API Framework. Všechny možnosti vytváření aplikací pro Android jsou dostupné přes kolekci aplikačních služeb napsaných v Javě. K těmto službám se řadí například správce notifikací, správa aktivit nebo prvky uživatelského rozhraní.
- Poslední, nejvyšší vrstvu, tvoří samotné aplikace. Systém obsahuje množinu základních aplikací, například kalendář, aplikaci pro sms, správce kontaktů a další. Tyto aplikace slouží jak uživatelům, tak vývojářům, kdy poskytují data pro jejich aplikace.



Obrázek 2.29: Vrstvový model architektury systému Android.²¹

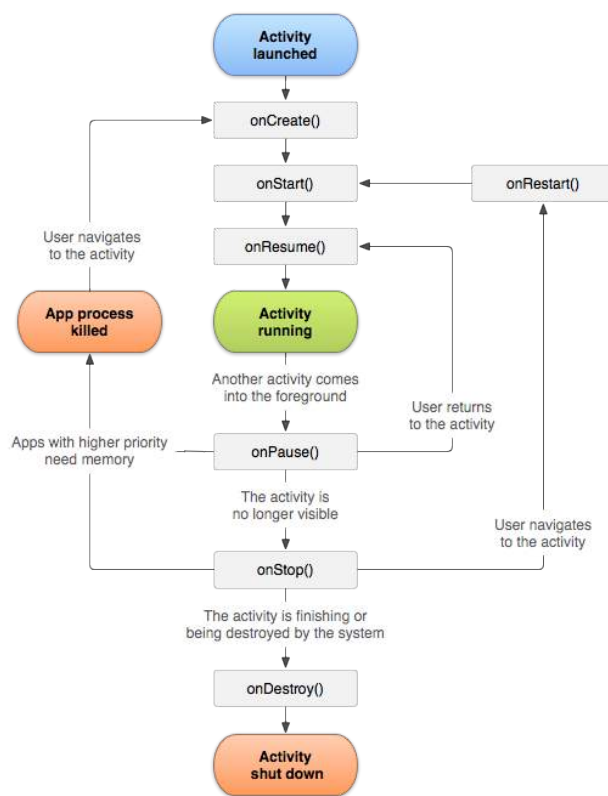
Tvorba aplikací

Aplikace pro platformu Android jsou psané v programovacím jazyku Java. Android SDK (Software Development Kit) přeloží výsledný kód spolu se všemi použitými zdrojovými soubory (obrázky, knihovny atd.), a vytvoří APK balíček (Android Package). Tento balíček obsahuje veškerý obsah aplikace a slouží jako instalační soubor. Aplikace jsou tvořeny z několika komponent, kde každá z nich může sloužit jako vstupní bod pro uživatele nebo systém. Existují čtyři hlavní komponenty^[1]:

- *Activity* - Jedná se o hlavní prvek pro komunikaci s uživatelem. Aktivity představují samostatné obrazovky, se kterými uživatel pracuje. Aplikace obsahuje zpravidla více aktivit a dohromady tvoří výsledné uživatelské rozhraní. Základ tvoří jedna hlavní aktivita, spouštěná při startu aplikace, která může vytvářet aktivity další. Každá aktivita monitoruje co zrovna uživatel vidí, obsahuje informaci o předchozí pozastavené

²¹Převzato z: developer.android.com/guide/platform/index.html

aktivitě, ke které by se uživatel mohl vrátit, vpomáhá aplikaci v případě jejího ukončení a poskytuje způsob, jakým může uživatel přecházet mezi jednotlivými aktivitami.



Obrázek 2.30: Životní cyklus aktivity v systému Android.²²

Aktivity jsou v systému spravovány formou zásobníku. Při vytvoření nové aktivity dojde k jejímu přidání na vrchol zásobníku, a předešlá aktivita se dostane do popředí, až po skončení nové aktivity. Životní cyklus aktivit je tvořen čtyřmi stavy (obrázek 2.30). První stav je v případě, kdy je aktivita na popředí a běží (Running). Přenesením na pozadí dojde k pozastavení (Paused), přičemž aktivita nezanikne a uchovává si všechny svoje hodnoty. Může však být ukončena systémem, například kvůli nedostatku paměti. Překrytím aktivity jinou aktivitou, dojde k úplnému pozastavení aktivity (Stop). V tomto stavu je často ukončena systémem. Jestliže aktivita správně skončí, nebo je ukončena systémem, ocitne se ve stavu ukončení (Destroy). Objekt aktivity obsahuje callback funkce (`onCreate()`, `onResume()`, `onStop()` atd.), které může využít programátor pro pokročilé nastavení chování aktivity v daném stavu.

- *Services* - Services představují obecný proces, který běží na pozadí a neobsahuje uživatelské rozhraní. Pro představu se může jednat o přehrávání hudby, zatímco je uživatel v jiné aplikaci, nebo stahování dat přes internet, kdy nedochází k blokování uživatele v interakci s aktivitou. Services může spouštět aplikace nebo systém.
- *Broadcast receivers* - Slouží k zasílání událostí skrze systém mimo standardní běh aplikací. Broadcast receivers je schopen doručit vyslanou událost i aplikaci, která mo-

²²Převzato z: developer.android.com/reference/android/app/Activity.html

mentálně neběží. Příkladem může být aplikace budíku. Uživatel nastaví dobu buzení, a do doby, než se aplikace aktivuje, není nutné aby běžela. Pomocí Broadcast receivers si mohou aplikace vyměňovat události mezi sebou.

- *Content providers* - Spravuje data aplikací a umožňuje je uložit do souborového systému, do databáze, či na web. Aplikace jsou schopny data mezi sebou sdílet.

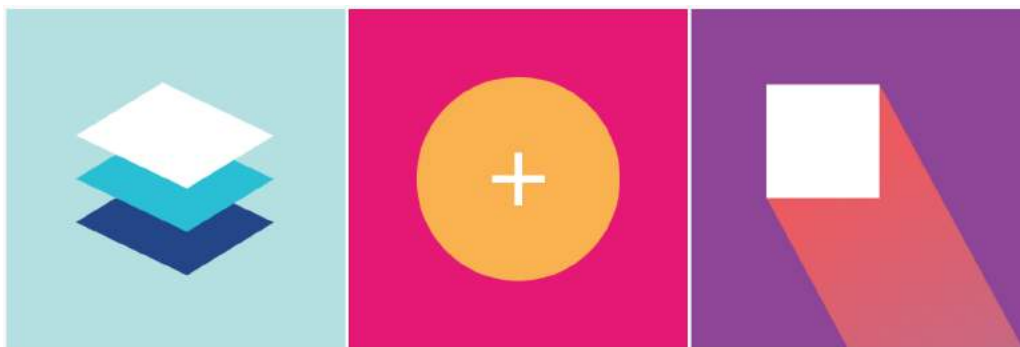
Vývojové prostředky

Vývojovým prostředím pro aplikace systému Android je Android Studio, vytvořené přímo společností Google. Výhodou Android Studia je, že již v základu obsahuje SDK (Software Development Kit) pro platformu Android. Tento balíček obsahuje potřebné nástroje pro jednoduchou tvorbu aplikací, od přednastavených šablon vzhledu jednotlivých aktivit, po emulátor mobilních zařízení. Pro jednoduché ladění aplikací existuje Android ADB (Android Debug Bridge). Jedná se o příkazový nástroj ke komunikaci s připojeným telefonem nebo spuštěným emulátorem. Pokud chceme vytvářenou aplikaci testovat přímo na našem telefonu skrze ADB, musí mít mobilní zařízení povolený režim pro vývojáře a debugování přes USB. Takto nastavený telefon připojíme přes USB k počítači s Android Studiem a můžeme výsledek programování vyzkoušet naživo. Jelikož aplikace pro Android jsou psané v programovacím jazyku Java, musí být dále nainstalováno JDK (Java Development Kit). Android Studio je vývojové prostředí založené na IntelliJ IDEA firmy JetBrains. Alternativou k Android Studio je prostředí Eclipse s požadovanými pluginy.

Grafické uživatelské rozhraní

Grafické uživatelské rozhraní (GUI) je vše, co uživatel v aplikaci vidí, a s čím může pracovat. Android nabízí celou řadu předem připravených UI komponent, jako jsou strukturované objekty rozložení a ovládací prvky. Dále jsou k dispozici pokročilé grafické prostředky, jako například dialogová okna, oznámení a nabídky, které nám umožňují vytvořit požadované grafické rozhraní.

V roce 2014 představila společnost Google nový grafický styl pro Android, který rychle převzaly jeho hlavní aplikace, jako Gmail, Google Drive, Google Maps atd. Myšlenka je taková, že prvky uživatelského prostředí reagují na dotek velmi intuitivním způsobem, animace jsou jemnější a barvy jsou jasnější a pastelovější, aby vynikly důležitější informace. Tento grafický styl byl nazván Material Design.



Obrázek 2.31: Material Design se snaží lépe a intuitivně popsat objekt, funkčnost a pohyb.²⁴

²⁴Převzato z: material.io/guidelines/#introduction-goals

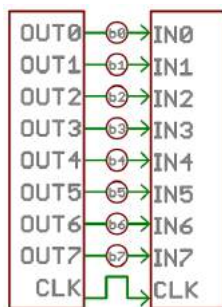
Nosným prvkem Material Designu je tzv. Material, který je založen na chování listu papíru a inkoustu. Na rozdíl od skutečného papíru se však digitální Material umí inteligentně roztahovat a přetvářet. Můžeme i říct, že má fyzický povrch a hrany. Důležitým prvkem u Material Designu je jeho 3D prostředí, ve kterém se nachází Material, světlo a stíny. Material se skutečně pohybuje v nějaké výšce, reprezentované na ose Z (standardní je 1dp). Stíny jsou pak vytvářeny výškovým rozdílem mezi překrývajícími se materiály.

Mezi další vlastnosti Materialu patří neproniknutelnost, nelze jej prokládat, nelze jej ohnout, může se v jakoukoli chvíli objevit i zaniknout, může se rozpojit, nebo slít v jeden celek a další.

Dodržování pravidel vytváření GUI pro platformu Android není nikde vyžadováno, jako je tomu např. u iOS, avšak jejich dodržáním přispíváme k dobré konzistenci aplikace. Díky tomu, že v současné době existuje velká spousta aplikací, které zásady Material Designu dodržují, a snahou Googlu tento styl aplikovat i na webové služby, nemají uživatelé Androidu problém, se rychle zorientovat v ovládní nové aplikace[5].

2.5 Sériová komunikace

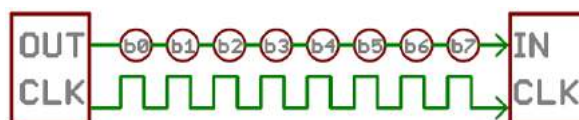
V počítačové komunikaci existují dva typy přenosů - sériový a paralelní. V paralelním přenosu se přenáší více bitů v jeden okamžik. Takové typy přenosů obvykle vyžadují široké datové sběrnice o 8, 16 a více vodičích. Výhodou paralelní komunikace je přenos celého binárního slova v jednu chvíli, což má za následek vyšší přenosovou rychlost než při sériové komunikaci. Nevýhodou jsou větší náklady. Paralelní sběrnice se využívají především k propojení vnitřních komponent počítače, kde je dostatečný prostor pro vodiče.



Obrázek 2.32: Schéma provedení paralelní sběrnice.²⁵

Při sériovém přenosu jsou jednotlivé datové bity postupně přenášeny po jednom vodiči. Sériového přenosu se využívá především u připojení vstup/výstupních zařízení, například k mikrokontroléru, kde počet spotřebovaných pinů může být kritický. V takovém případě se obětuje rychlost komunikace pro větší funkcionalitu. Sériově jsou data přenášena v počítačových sítích. Všechny druhy bezdrátového přenosu jsou rovněž sériové. V práci je dále věnována pozornost sériovému způsobu komunikace[13].

²⁵Převzato z: learn.sparkfun.com/tutorials/serial-communication



Obrázek 2.33: Schéma provedení sériové sběrnice.²⁶

V průběhu let vznikl nespočet sériových protokolů, které pokrývaly potřeby různých vestavěných systémů. Mezi nejznámější patří rozhraní USB nebo Ethernet. Další známé rozhraní jsou například SPI, UART nebo I²C. Každé z těchto sériových rozhraní může být rozděleno na synchronní a asynchronní[13].

2.5.1 Asynchronní přenos

Asynchronní přenos dat se musí obejít bez podpory externího hodinového signálu. Tato metoda je výhodná z důvodu minimalizace počtu použitých vodičů. Na druhou stranu se musí vložit více úsilí při spolehlivosti přenosu a příjmu dat.

Jelikož zde není synchronizační signál, musí k synchronizaci dat dojít jiným způsobem. Přenos je rozdělen do sekvencí (Frame), kde každá sekvence obsahuje startovací a ukončovací posloupnost, podle kterých dojde k synchronizaci přijímače a vysílače. Celková sekvence se skládá ze:

- synchronizačních bitů (Start a Stop)
- datových bitů
- paritních bitů

V klidovém stavu je sériová sběrnice v logické úrovni 1. Generování Start bitu změni úroveň na logickou 0. Následně se vysílají datové bity, za kterými se mohou vyskytnout bity paritní. Konec přenosu jednoho rámce je signalizován Stop bitem, což je změna zpět na klidovou úroveň[13][12].



Obrázek 2.34: Datový rámec sériové komunikace.²⁷

Počet jednotlivých bitů může být samostatně nakonfigurován, a proto se přenosová sekvence na dvou sériových asynchronních sběrnících může mírně lišit. Důležité je, aby jak přijímač, tak vysílač, sdíleli stejnou přenosovou rychlost, a měli shodně nastavené parametry přenosu. Přenos rámce začíná asynchronně, avšak zpracování na straně vysílače i přijímače probíhá synchronně. Synchronnost v tomto případě znamená, že frekvence hodinových generátorů jsou na obou stranách stejné. Tyto frekvence musí zůstat dostatečně přesné, alespoň po dobu přenosu jednoho znaku. Pak se mohou libovolně odlišovat, jelikož synchronizaci zajistí sestupná hrana nového Start bitu.

²⁶Převzato z: learn.sparkfun.com/tutorials/serial-communication

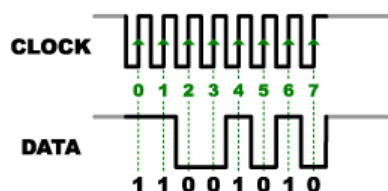
²⁷Převzato z: learn.sparkfun.com/tutorials/serial-communication

Jeden z parametrů pro zajištění správné asynchronní sériové komunikace je rychlost přenosu. Rychlost se udává v jednotkách Bd (Baud), a značí periodu vzorkovacího intervalu, který je generován při odesílání a příjmu dat. Rychlost přenosu musí být nastavena na přijímači a vysílači ještě před začátkem komunikace. Převrácená hodnota rychlosti nám udává dobu, která je potřebná pro odeslání jednoho bitu po sběrnici. Například pro rychlost 19,200 Bd trvá přenos jednoho bitu 52 μ s. Aby se vyrovnalo zpoždění mezi oběma komunikujícími stranami, vzorkuje přijímač jednotlivé bity uprostřed jejich intervalu[13][12].

2.5.2 Synchronní přenos

U synchronního sériového přenosu se, na rozdíl od asynchronního, využívá externí synchronizační signál. Data jsou vždy spárována s hodinovým signálem, takže všechna zařízení na této sběrnici sdílejí společné hodiny.

Synchronní přenosy neobsahují žádné synchronizační Start ani Stop bity, a proto přijímač i vysílač musí celou dobu pracovat naprosto synchronně. Nejrozšířenějším způsobem, jak přenášet synchronizaci směrem k přijímači, je vyhradit další samostatný vodič pro synchronizační signál, který je souběžný s datovým.



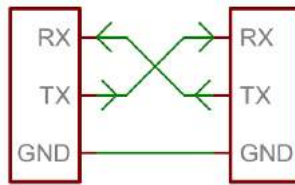
Obrázek 2.35: Příklad možnosti zaslání synchronizačního signálu přijímacímu zařízení.

Další možností, jak přenést synchronizaci k přijímači, je zkombinovat synchronizační signál s datovým. Tímto je docíleno, že se vše přenáší po jednom společném vodiči. Jeden způsob provedení vkládá za každý datový bit jeden synchronizační, kdy dochází ke dvěma změnám přenášeného signálu. První změna určuje data, druhá hodiny. Tento postup však vede ke snížení rychlosti přenosu, jelikož datový bit je pouze každý druhý. Další způsob je synchronizace přímo z dat. Jsou-li posílané bity reprezentovány změnou mezi dvěma úrovněmi, lze k synchronizaci využít tuto hranu. Problém v tomto případě nastává, objeví-li se dlouhá posloupnost jedniček nebo nul. Po delší dobu nedochází ke změnám úrovně, a přijímač se tak nemá podle čeho synchronizovat. Řešení poskytuje metoda *Bit Stuffing*, která po dohodnuté sérii jedniček nebo nul vkládá hodnotu opačnou, a tím synchronizuje přijímač. Tento princip využívá například rozhraní USB.

Synchronní přenos je, oproti asynchronnímu, v provedení přímočařejší a většinou i rychlejší, ale také vyžaduje jeden další vodič mezi komunikujícími zařízeními. Další výhodou je, že na straně příjemce není nutný oscilátor, který by vzorkoval příchozí signál[13][12].

2.5.3 Rozhraní UART

Rozhraní UART (Universal Asynchronous Receiver Transmitter) slouží pro sériovou asynchronní komunikaci (viz. 2.5.1) a podporuje full-duplex přenos. Může tedy v jednom okamžiku data přijímat i vysílat.

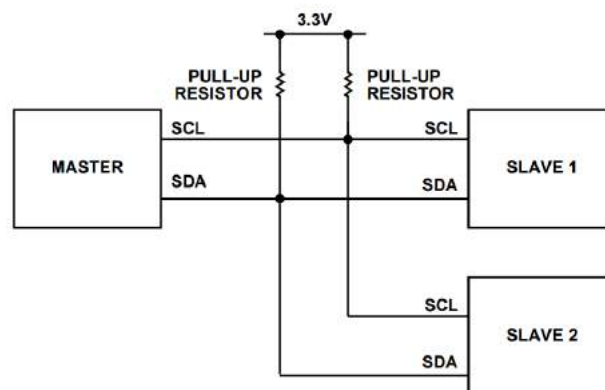


Obrázek 2.36: Obecné schéma sériového rozhraní pro příjem a odesílání dat.²⁸

Rozhraní UART vysílá data na pinu označeném jako Tx, a přijímá na pinu Rx. Komunikace mezi zařízeními pak probíhá změnou z klidové úrovně (logická 1). Transakce je zahájena zasláním Start bitu, následovaného bity datovými, konec je určen pomocí Stop bitu[12].

2.5.4 Sběrnice I²C

Sběrnice I²C (Inter-integrated Circuit) je synchronní sériové rozhraní (viz. 2.5.2) pro half-duplex komunikaci. V jeden okamžik může zařízení na této sběrnici pouze číst nebo zapisovat. Komunikační protokol sběrnice je navržen tak, aby umožnil více *Slave* zařízením komunikovat s jedním nebo více *Master* zařízením. V roli Master je většinou mikrokontrolér. Sběrnici vyvinula firma Philips v roce 1982 pro komunikaci mezi jejich integrovanými obvody. Obdobou pro I²C je rozhraní TWI od společnosti Atmel[13][4].

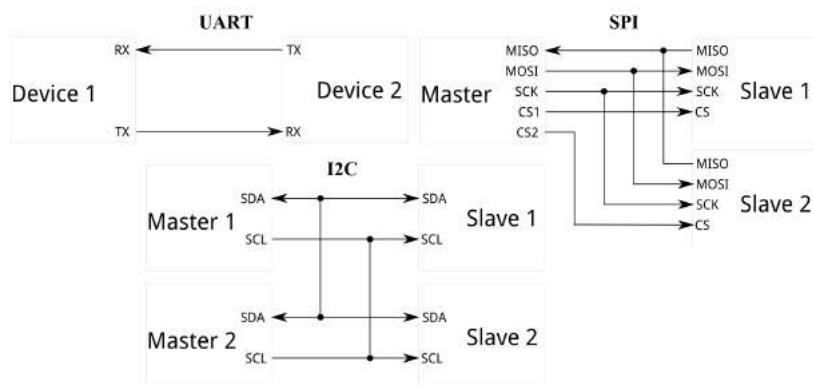


Obrázek 2.37: Diagram zapojení na sběrnici I²C.²⁹

Výhodou klasických asynchronních rozhraní (např. UART) je, že využívají ke komunikaci pouze dva vodiče (Tx a Rx). Avšak nutnost generování hodinového signálu u přijímače i vysílače, složitost provedení a fakt, že dokáží správně komunikovat pouze mezi dvěma propojenými zařízeními, je oproti synchronním sběrnícím znevýhodňuje. Nevýhodou u synchronních rozhraní, jako je například sběrnice SPI, je velký počet propojovacích vodičů (u SPI 4) a omezení na pouze jedno řídicí Master zařízení, ke kterému je připojeno více Slave obvodů. Rozhraní I²C kombinuje vlastnosti z obou popsaných sběrnic. K přenosu vyžaduje, podobně jako asynchronní sběrnice, pouze dva vodiče. Může komunikovat s více zařízeními, na jedné lince a na rozdíl od SPI, může v zapojení existovat více Master obvodů (Multi-master)[13][4].

²⁸Převzato z: learn.sparkfun.com/tutorials/serial-communication

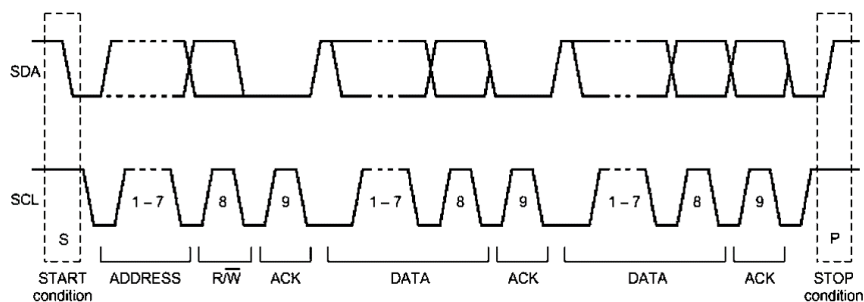
²⁹Převzato z: www.analog.com/media/en/technical-documentation/application-notes/AN-895.pdf



Obrázek 2.38: Rozdíl v provedení mezi UART, SPI a I²C.³⁰

Každá I²C sběrnice sestává ze dvou linek, signál SCL je hodinový a SDA datový (Obrázek 2.37). Synchronizační hodinový signál je vždy generován určeným Master obvodem. Na rozdíl od rozhraní UART nebo SPI je I²C sběrnice s otevřeným kolektorem, což znamená, že může při komunikaci stáhnout úroveň napětí do 0, ale už není schopna toto napětí zvednout zpět do aktivní úrovně. Pro obnovu napětí na požadovanou úroveň slouží externí pull-up rezistory, připojené na linku SCL i SDA. S rostoucí frekvencí komunikace musí být hodnota těchto odporů nižší. Pro standardní frekvenci přenosu 100 kHz, je doporučená hodnota odporů 4,7 kΩ. Vylepšené verze sběrnice I²C umožňují frekvenci 400 kHz i 1 MHz.

Komunikace na I²C je komplexnější oproti UART nebo SPI sběrnicím. Přenos je rozdělen na adresové a datové rámce. V adresovém bloku Master stanoví adresu zařízení, se kterým si přeje komunikovat. Za adresovým rámcem následuje jedna nebo více datových sekvencí. Data jsou vystavena na linku SDA těsně poté, co dojde k poklesu úrovně na SCL. Vzorkování dat probíhá na každou nástupnou hranu hodinového signálu[4].



Obrázek 2.39: Časování na sběrnici I²C.³¹

K zahájení komunikace dochází tehdy, pokud Master zařízení stáhne úroveň SDA a ponechá SCL signál v aktivní úrovni. Této konfiguraci linek SCL a SDA se říká *Start Condition*. Slave zařízení je Startem uvědoměno o novém příchozím spojení. V případě, kdy dvě Master zařízení zažádají o přidělení sběrnice, je linka přidělena tomu, které vystaví Start jako první.

³⁰Převzato z: learn.sparkfun.com/tutorials/i2c

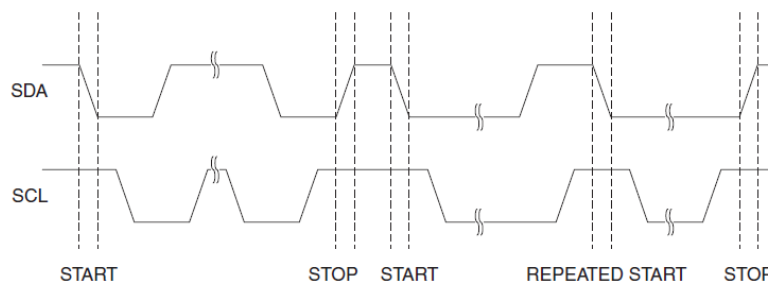
³¹Převzato z: maxembedded.com/2014/02/inter-integrated-circuits-i2c-basics/

Jako první se vždy k Slave zařízení odesílá adresový rámeček. Adresa na sběrnici I²C má 7 bitů ve speciálních případech 10. V základu tedy můžeme adresovat až 128 zařízení. Přenos adresy je od MSB k LSB. Adresa je následována jedním bitem určujícím požadovanou operaci (čtení nebo zápis). Posledním přenášeným bitem v adresové sekvenci je potvrzovací NACK/ACK. Potvrzovací bit se vysílá u každého adresového i datového rámečku. Jakmile je posláno prvních osm bitů, přebere kontrolu nad SDA linkou příjemce. Pokud přijímač nestáhne datový signál do 0 před příchodem potvrzovacího bitu, dojde k chybě při přenosu, a komunikace je zastavena.

Po odeslání adresového rámečku je zaslán rámeček datový. Master pokračuje v generování hodinového signálu a data jsou vložena na SDA buď Master nebo Slave zařízením. Počet datových rámečků může být libovolný a většina Slave zařízení si automaticky inkrementuje vnitřní registr, což značí, že data přijatá nebo odeslaná budou zařazena do následujícího registru v pořadí.

Jakmile skončí přenos dat, Master zařízení vygeneruje *Stop Condition*. Jedná se o změnu z nízké (0) úrovně na aktivní (1). Na SDA lince je tato změna provedena až po SCL. Na sběrnici by tedy nemělo docházet ke změnám úrovně na lince SDA v případě, kdy je SCL aktivní, aby tato změna nebyla u přijímače chápána jako Stop.

Speciálním případem při komunikaci na I²C sběrnici je *Repeated Start*. V některých případech je důležité, aby si Master vyměnil se Slave zařízením několik zpráv najednou. Při této výměně však nesmí dojít k zásahu ze strany ostatních Master obvodů. Pro vytvoření Repeated Start je úroveň SDA signálu změněna na 1, zatímco SCL zůstává v 0. SCL je poté změněna na 1 a v tomto stavu je SDA stažena zpět na 0 jako u klasického Start. V tomto případě nebyl nikdy vygenerován Stop, a předchozí komunikace nebyla ukončena. Sběrnice tedy zůstává přidělena původnímu Master zařízení a komunikace může pokračovat dle standardního protokolu[4].



Obrázek 2.40: Generování Repeated Start na sběrnici I²C.³²

³²Převzato z: www.avrbeginners.net/architecture/twi/twi.html

Kapitola 3

Analýza a návrh řešení

Kapitola analyzuje informace z kapitoly 2. Na základě analýzy je popsán návrh řešení, pro vytvoření systému schopného tarifkovat provoz ultralehkého letadla z údajů o zrychlení a frekvenci otáček motoru.

3.1 Analýza současného stavu

Pro tarifkaci ultralehkých letadel se používá doba letu. Doba trvání letu si může pilot změřit sám, nebo lze na základě provozních parametrů, konkrétně otáček motoru a zrychlení, určit okamžik, kdy letadlo vzlétne a přistane. Interval mezi vzlétnutím a přistáním letadla se pak změří automaticky.

Otáčkoměry popsané v kapitole 2.1.3 a 2.1.4 spadají do kategorie kontaktního měření otáček (2.1.1). Nejsou příliš vhodné, jelikož vyžadují poměrně značné mechanické úpravy v konstrukci motoru i letadla. Bylo by zapotřebí přímé spojení s rotující hřídelí, nebo jinou částí motoru.

Impulsní a fotoelektrické otáčkoměry (kapitola 2.1.5 a 2.1.6) jsou mnohem více použitelné metody ke snímání otáček, aspoň v našem případě, než mechanické a spojitě. Metody z kategorie indukčních, kapacitních, magnetických a reflexních závor nabízejí, oproti výše zmíněným, jednodušší provedení způsobu měření. Tyto metody vyžadují umístění nějaké "značky" na list vrtule a k této značce přidat odpovídající čidlo. Například v případě magnetických otáčkoměrů se na list vrtule přidá magnet a do jeho blízkosti se umístí Hallova sonda. Rotací vrtule dochází ke generování Hallova napětí, a z vytvářených pulsů lze odvodit frekvenci otáček. Podobný způsob lze uplatnit i v případě indukčních a kapacitních snímačů. Při použití reflexních závor, plní značka na vrtuli funkci odražeče světelného paprsku. Čidlo se skládá ze zdroje světelného paprsku (např. laser) a fotodiody, která odražený paprsek přemění na napětí. Ovšem i tyto bezkontaktní (viz. 2.1.2) metody způsobu snímání a měření frekvence otáček motoru vyžadují velké zásahy do konstrukce letadla. Bylo by však lépe se jim vyhnout.

Pokud nechceme pro měření otáček provádět konstrukční úpravy na letadle, musíme pracovat s prostředky, které nabízí samotné letadlo. Po konzultaci s vedoucím práce bylo zjištěno, že letadlo disponuje servisním konektorem, který na výstupu nabízí signál od zapalovacích cívek motoru letadla. Pokud by byl tento signál přiveden na vstup elektronického zařízení, mohla by se díky pulsům od zapalování, měřit frekvence otáčení. V práci je využita tato možnost.

Podle informací z kapitoly 2.3 je nejlepší volbou pro měření zrychlení použít kapacitní akcelerometry (viz. 2.3.1). Tyto akcelerometry jsou vhodné pro měření nízkých zrychlení, které v ultralehkém letadle dosahují maximálně -2 až +4 g. Dále nabízejí vysokou přesnost a citlivost. Piezoelektrické akcelerometry (viz. 2.3.2) nejsou schopny měřit statické zrychlení, a citlivost tepelných akcelerometrů (viz. 2.3.4) může být ovlivněna teplem od motoru letadla.

Jak bylo popsáno výše, je k měření frekvence otáček využít signál od zapalovacích svíček motoru letadla. Signál bude, spolu s údaji o zrychlení z kapacitního akcelerometru, přiveden na vstup elektronického obvodu, který údaje vyhodnotí. Měřicí zařízení bude umístěno na palubě letadla. Další vlastností měřiče, je schopnost měřené hodnoty nějakým způsobem předat pilotovi, a určit dobu letu pro následnou tarifkaci. Nabízí se možnost získané údaje odeslat pomocí bezdrátového přenosu do mobilního zařízení. Mobilním zařízením je mobilní telefon, který má pilot k dispozici po celou dobu letu. Existuje několik mobilních operačních systémů. Díky rozšířenosti systému, velké komunitě vývojářů a předešlým zkušenostem s programováním, bude pro komunikaci se snímačem zvolen mobilní telefon s operačním systémem Android (viz. 2.4).

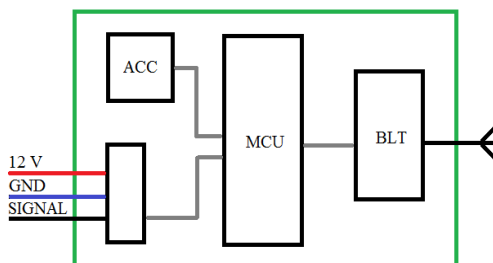
Komunikace v rámci měřicího obvodu a mobilního zařízení musí vycházet ze zásad počítačové komunikace popsaných v kapitole 2.5. Bezdrátový přenos dat do mobilního telefonu je vždy sériový. Formát přenášených dat o zrychlení záleží na typu použitého akcelerometru. Pro úsporu místa na měřicím zařízení, bude preferován sériový způsob komunikace.

3.2 Návrh řešení

Dle analýzy provedené v kapitole 3.1, se celkový systém pro tarifkaci letadla sestává ze dvou částí. První částí je elektronické zařízení, které snímá údaje o otáčkách motoru a zrychlení letadla, a tato data vysílá pomocí bezdrátové technologie. Druhá částí je aplikace v mobilním telefonu, která přijme údaje vysílané měřicím zařízením, a na základě těchto hodnot změří pilotovi dobu letu.

3.2.1 Měřicí zařízení

Měřicí zařízení pro snímání otáček a zrychlení obsahuje tři vstupy, napájení obvodu, uzemnění a signál od zapalovacích cívek. Zrychlení je snímáno pomocí vestavěného modulu s akcelerometrem. Možností je, pro řízení snímače, využít mikrokontrolér, který do systému přinese flexibilitu a možnost rychle reagovat na případné změny v návrhu. Na vstupy mikrokontroléru budou přivedeny měřené údaje o otáčkách a zrychlení. Mikrokontrolér na výstupu obsahuje modul pro bezdrátovou komunikaci, který data odešle do mobilního telefonu.



Obrázek 3.1: Blokový diagram navrhovaného měřicího zařízení pro snímání otáček motoru a zrychlení letadla.

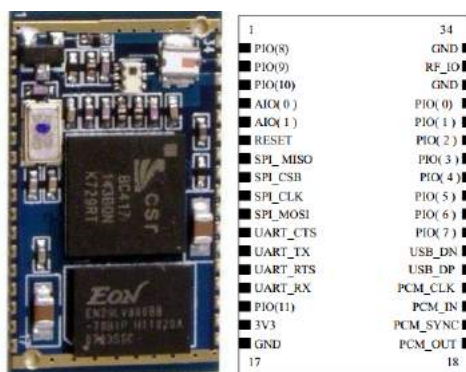
Modul akcelerometru musí, při použití mikrokontroléru, obsahovat digitální výstupy pro připojení k procesoru. Další podmínkou je měření zrychlení ve všech třech osách, kvůli možnému pohybu letadla ve vzduchu. Omezením celého měřicího systému je velikost palubního napětí, kterým bude obvod napájen. Napětí na palubě letadla je 12 V. Výběr rádiového modulu pro bezdrátovou komunikaci je ovlivněn přijímači dostupnými v dnešních mobilních telefonech. Do úvahy připadá využití Bluetooth nebo WiFi modulů, které jsou v telefonech standardem. V práci je použita technologie Bluetooth, kvůli nižší energetické spotřebě, a kvůli jednoduchosti komunikačního protokolu. Důraz je kladen rovněž na spotřebu výsledného zařízení, která by měla být co nejmenší. Proto je důležité vybrat vhodný mikrokontrolér, nejlépe s architekturou procesoru typu ARM. Mikrokontrolér dále musí obsahovat dostatečný počet sériových vstupů a výstupů, jelikož akcelerometr i Bluetooth modul budou komunikovat přes sériové rozhraní. Rovněž signál od zapalovacích cívek motoru letadla bude přiveden na jeden ze sériových vstupů.

3.2.2 Popis vybraných modulů

Elektronické moduly pro zhotovení měřicího zařízení byly vybrány na základě vlastností, popsanych v kapitole 3.1 a 3.2.1. Jedná se o Bluetooth modul BTM-112, který slouží pro bezdrátový přenos dat do mobilního zařízení, akcelerometr MMA8452Q a řídicí mikrokontrolér s procesorem ADuC7020.

Bluetooth BTM-112

Jak již bylo zmíněno v kapitole 3.1, bude pro zaslání dat ze snímače do mobilního telefonu využito technologie Bluetooth. Přenos bezdrátovou sítí vyžaduje dvě zařízení, přijímač a vysílač. Bluetooth přijímač je integrován v mobilním telefonu. Je tedy potřeba pořídit vysílač. Pro tento účel jsou vhodné již sestavené Bluetooth moduly, kterých je na trhu dostatečné množství. Výběr vhodného modulu pak závisí na rozměrech, ceně, dosahu signálu, komunikačním protokolu, nebo na rychlosti přenosu. V práci byl zvolen modul BTM-112.



Obrázek 3.2: Bluetooth modul BTM-112[3].

Modul v Bluetooth síti funguje jako server a je schopen přijmout požadavky od klienta. Klientem může být notebook nebo mobilní telefon. Mezi vlastnosti modulu patří vysílací frekvence 2,4 GHz, napájecí napětí 3,3 V s vnitřním regulátorem na 1,8 V. Obousměrnou komunikaci zajišťuje podpora full-duplex přenosu. Modul podporuje standard 2,0 + EDR, má nízký příkon a komunikace je možná až se sedmi klienty. Výstupní výkon modulu je

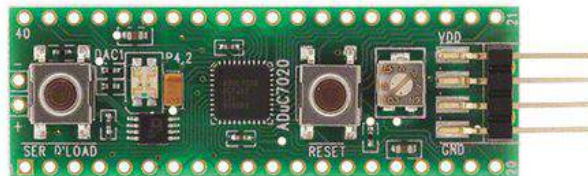
maximálně 4 dBm, což s anténou zajišťuje dosah kolem 100 m, bez antény jednotky metrů. Hmotnost modulu je 2,24 g a provozní teplota je od -40°C do 85°C. K dispozici je komunikační rozhraní UART, PCM, SPI a USB[3].

Ovládání modulu je možné prostřednictvím jeho sériového UART rozhraní. Jelikož většina počítačů dnes neobsahuje sériové porty, je nutné přidat redukcí pro převod z UART na USB. Po zapnutí a spárování modulu s notebookem, se BTM-112 objeví jako nový virtuální COM port s přiděleným číslem. Tímto způsobem lze s modulem komunikovat pomocí softwaru pro sériovou komunikaci. Pro naprogramování modulu byl použit volně dostupný program *Hercules SETUP utility*, který má funkci terminálu. Nastavení modulu lze měnit zasláním AT příkazů. Je možné změnit párovací PIN kód (základní je 1234), jméno zařízení, rychlost komunikace atd. Základní nastavení sériového rozhraní UART modulu je:

- Baud Rate: 19 200 Bd
- Datové bity: 8
- Paritní bity: 0
- Stop bit: 1
- Flow control: H/W or none

Mikrokontrolér EVAL AduC7020 MKZ

Řídící jednotkou celého snímače je mikrokontrolér s procesorem ADuC7020. Procesor bude na svých vstupech dostávat signál ze zapalovacích cívek motoru letadla a údaje o zrychlení od akcelerometru. Získané údaje zpracuje a odešle přes sériové rozhraní UART na Bluetooth modul BTM-112.



Obrázek 3.3: Mikrokontrolér EVAL ADuC7020 MKZ.¹

Procesory řady ADuC702x jsou plně integrované, obsahují 12-bitový analogově-digitální převodník, schopný vzorkovací rychlosti až 1 MSPS. ADuC7020 je procesor s jádrem ARM7 a 16/32 bitovou RICS architekturou. Frekvenci generuje on-chip oscilátor a PLL s maximální hodnotou 41,78 MHz. Maximální výkon procesoru je 41 MIPS. Obsahuje 8 kB SRAM a 62 kB nevolatilní Flash/EE paměti. Procesor obsahuje celou řadu periférií, jako například UART, I²C nebo SPI, a již disponuje obvodem pro nahrávání programu do paměti. Vstupní napětí je regulované na 3,6 V. Provozní teploty se pohybují od -40°C do 125°C. Pokud procesor pracuje na frekvenci 41,78 MHz, je jeho spotřeba 120 mW.

Dle informací z kapitoly 3.2.1 bude signál ze zapalovacích cívek motoru letadla přiveden na sériový vstup mikrokontroléru. V kapitole 2.5 jsou popsány dva způsoby sériové komunikace - synchronní a asynchronní. Synchronní způsob není příliš vhodný, jelikož signál z

¹Převzato z: img1.findic.com/eval-aduc7020mkz-cVTq2y8g-0q381Q3Yo.jpg

cívek povede pouze jedním vodičem na vstup mikrokontroléru, a nebude jak vytvořit patřičný synchronizační signál. Řešením je použit asynchronní způsob komunikace, a na vstup mikrokontroléru zasílat patřičné datové rámce (viz. 2.5.1).

Aby byl procesor schopen registrovat příjem nového znaku, musí signál z cívek splňovat podmínky komunikačního protokolu pro asynchronní sériové rozhraní. V rámci domluvené rychlosti přenosu mezi přijímačem (MCU) a vysílačem (signál z cívek), musí přenos jednoho bitu trvat vypočtený interval, aby byl na straně příjemce správně vzorkován. Jelikož je Bluetooth modul BTM-112 nastaven na rychlost přenosu po sériové lince na 19,200 Bd, musí být na stejnou rychlost nastaven i procesor. Z této rychlosti plyne, že interval po který je přijímán nebo vyslán 1 bit je $52 \mu s$ (viz. rovnice 3.1).

$$\frac{1}{19200} \doteq 0.000052s \quad (3.1)$$

Sériový přenos může mít až 8 datových bitů, rozsah posílaných hodnot je tedy 0 až 255. Jediná informace, kterou chceme ze signálu od zapalovacích svíček získat, je frekvence příchozích pulsů, ze kterých se odvodí frekvence otáček motoru letadla. Je tedy důležité správně zaslat znak na vstupní pin rozhraní UART mikrokontroléru, ale už není důležitá jeho hodnota. Pro tento účel postačí vygenerovat Start bit, který značí začátek přenosu změnou úrovně z 1 na 0 a zbytek datových bitů spolu se Stop bitem nechat v logické úrovni 1. Tímto bude docíleno správné odeslání jednoho znaku po sériové sběrnici s hodnotou 255.

Idle	Start	Data	Data	Data	Data	Data	Data	Data	Data	Stop
		0	1	2	3	4	5	6	7	

Obrázek 3.4: Rámec zaslaný na UART vstup mikrokontroléru.

Akcelerometr MMA8452Q

Modul akcelerometru slouží pro měření zrychlení a spolu s údaji o otáčkách rozhodne o stavu letadla. Z analýzy v kapitole 3.1 plyne, že modul bude tvořen kapacitním akcelerometrem, který je schopný sériové komunikace s mikrokontrolérem. Na základě těchto parametrů byl zvolen modul s akcelerometrem MMA8452Q. MMA8452Q je nízkoenergetický, kapacitní (viz. 2.3.1), digitální, 3-osý (x, y, z) akcelerometr s komunikačním rozhraním I²C (viz. 2.5.4). Akcelerometr nabízí dynamický rozsah měření v ± 2 g, ± 4 g a ± 8 g. Velikost dat je 8 nebo 12 bitů. Údaje mohou být zpracovány vysokofrekvenčním filtrem, který odfiltruje konstantní gravitační zrychlení. Frekvence získání nových dat je od 1,56 Hz do 800 Hz. Napájecí napětí je do 3,6 V[6].



Obrázek 3.5: Modul s akcelerometrem MA8452Q.²

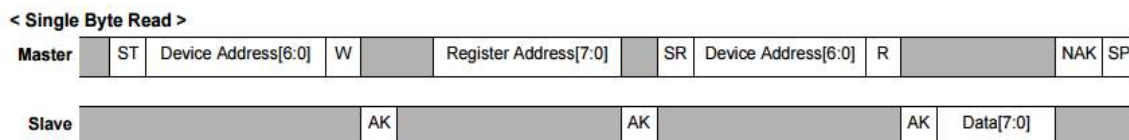
²Převzato z: www.sparkfun.com/products/13926

Jelikož akcelerometr komunikuje přes sběrnici I²C, musí se řídit komunikačním protokolem popsaným v kapitole 2.5.4. Akcelerometr má 7-bitovou adresu, kde prvních 6 bitů je pořad stejných (001110), a sedmý bit je určen hodnotou na pinu SA0. Pokud je na tomto pinu hodnota vstupního napětí, je poslední adresový bit v 1, celková adresa zařízení je tak 0x1D. Přivedením země (GND) na pin SA0 se k adrese přidá bit s hodnotou 0 a adresa je tedy 0x1C. Z výroby je adresa akcelerometru nastavena na hodnotu 0x1D, která v práci zůstala nezměněna. Konečná hodnota zasílaná v adresovém rámci (viz. 2.5.4) na rozhraní I²C je určena typem prováděné operace. Čtecí operace přidá k adrese zařízení bit s hodnotou 1, zapisovací bit s hodnotou 0. V našem případě je konečná hodnota pro čtení 0x3B a pro zápis 0x3A[6].

Command	[7:2] Device Address	[1] SA0	[7:1] Device Address	R/W	[7:0] 8-bit Final Value
Read	001110	0	0x1C	1	0x39
Write	001110	0	0x1C	0	0x38
Read	001110	1	0x1D	1	0x3B
Write	001110	1	0x1D	0	0x3A

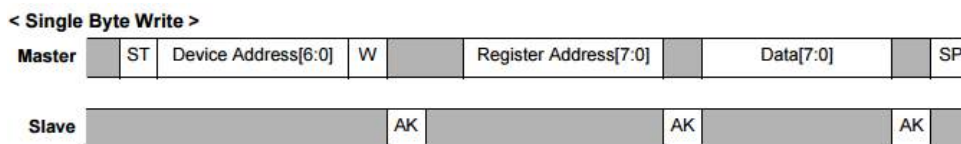
Tabulka 3.1: Jednotlivé adresy akcelerometru vzhledem k prováděným operacím.

Jednotlivé registry akcelerometru podporují jak čtení/zápis jednoho bytu dat, tak i přenos několika bytů za sebou. Pro čtení z registru je nejdříve na sběrnici vystavena adresa zařízení pro operaci zápisu (0x3A), za ní následuje adresa registru, ze kterého se bude číst, dále následuje adresa zařízení pro operaci čtení (0x3B). Po takto vyslaném požadavku akcelerometr zašle obsah daného registru. Operace pro čtení vyžaduje generování Repeated Start (viz. 2.5.4).



Obrázek 3.6: Diagram pro čtení dat z MMA8452Q[6].

Zapisovací operace je z hlediska odesílaných rámců jednodušší. Začátek operace opět vyžaduje adresu zařízení pro zápis (0x3A), následuje registr, do kterého chceme data zapsat, a nakonec zašleme samotná data.

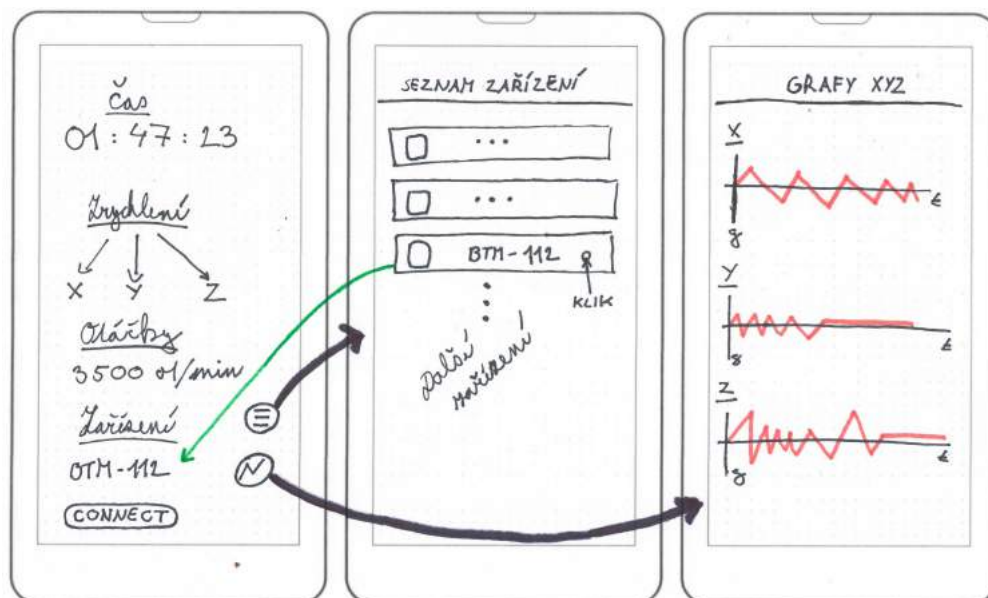


Obrázek 3.7: Diagram pro zápis dat do registru MMA8452Q[6].

3.2.3 Mobilní aplikace

Hlavním účelem mobilní aplikace je změřit dobu letu pro následnou tarifkaci, a patřičně tuto informaci zobrazovat pilotovi. Doba letu se změří na základě dat přijatých od měřícího

zařízení na palubě letadla (viz. 3.2.1). Aplikace nesmí být náročná na obsluhu a nejlépe by neměla nevyžadovat žádnou interakci s pilotem, kromě připojení na snímač zrychlení a otáček. Důležité informace, zobrazované pilotovi, jsou uplynulý čas letu, hodnota zrychlení a frekvence otáček.



Obrázek 3.8: Náčrtek jednotlivých obrazovek mobilní aplikace.

Aplikace se pomocí bezdrátového Bluetooth adaptéru, vestavěného v mobilním telefonu, připojí na Bluetooth modul ve snímacím zařízení. Díky vytvořenému spojení, bude moci mobilní aplikace přijímat data od snímacího zařízení, a na jejich základě tarifkovat provoz letadla.

Návrh aplikace je tvořen ze tří obrazovek. Hlavní obrazovka ukazuje dobu letu, údaje o zrychlení a otáčkách, získané od měřícího zařízení, a jméno zařízení, na které je připojena. Druhá obrazovka obsahuje seznam všech Bluetooth zařízení, která jsou s telefonem již spárována. Pilot si vybere zařízení s označením BTM-112. Poslední obrazovka slouží pro vykreslení grafů z údajů o zrychlení v jednotlivých osách.

Mobilní aplikace je vytvářena pro platformu Android a vychází z informací popsanych v kapitole 3.2.3.

Kapitola 4

Řešení a implementace

Kapitola obsahuje podrobné informace o vývoji a realizaci hardwaru snímacího zařízení. Bude zde popsáno elektrické zapojení jednotlivých součástí a integrovaných obvodů. Zařízení snímá signál od zapalovacích svíček motoru letadla, ze kterého je vypočtena frekvence otáčení motoru. Dále je snímána hodnota zrychlení z vestavěného akcelerometru. Údaje o otáčkách a zrychlení jsou bezdrátově přeneseny do mobilního telefonu. Kapitola je dále věnována popisu implementace mikrokontroléru, který řídí snímací zařízení, a popisu mobilní aplikace, jejímž úkolem je navázání spojení, přenos dat a zobrazení měřených hodnot.

4.1 Hardwarová část řešení

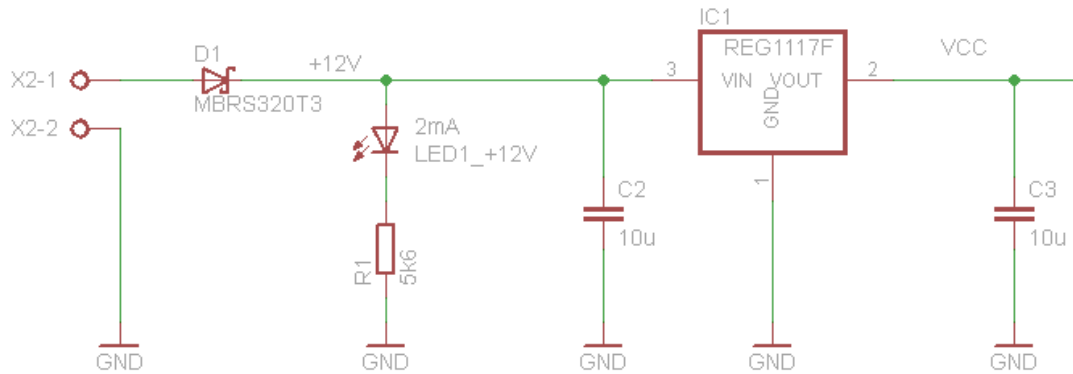
V hardwarové části je popis návrhu a procesu výroby měřícího zařízení, které snímá údaje o otáčkách motoru a zrychlení letadla. První část kapitoly se zabývá návrhem elektrického obvodu snímače. Ve druhé části je popsán proces výroby desky plošného spoje, která je osazena jednotlivými vybranými součástkami. Celkové blokové schéma obvodu odpovídá diagramu 3.1.

Srdcem zařízení je mikrokontrolér s procesorem ADuC7020. Na synchronní sériové rozhraní procesoru (sběrnice I²C) je napojen modul akcelerometru MMA8452Q. Na UART vstup procesoru je přiveden signál od zapalovacích cívek. Tento signál je pomocí obvodu s monostabilním klopným obvodem vytvarován tak, aby splňoval formát komunikace na sériové lince. Procesor vyčítá oba údaje a následně je přeposílá na své výstupní UART rozhraní. Na sériový výstup procesoru je napojen Bluetooth modul BTM-112, který data bezdrátově odesílá. Celý systém je napájen palubním napětím 12 V. Jelikož všechny integrované obvody pracují na napětí 3,3 V, je zde vytvořen napájecí stabilizační obvod. Samotný signál z cívek může obsahovat vysoké špičky napětí, a je tedy nutné celkový obvod chránit. K tomu slouží ochranný obvod mezi vstupem snímače a monostabilním obvodem.

4.1.1 Popis elektrického obvodu

V této části práce je popsáno zapojení jednotlivých elektronických komponent snímače. Celkové schéma zapojení obsahuje moduly popsané v části 3.2.2 spolu s napájecím obvodem, ochranným obvodem a obvodem pro generování pulsů pomocí monostabilního klopného obvodu.

Napájecí obvod



Obrázek 4.1: Schéma zapojení napájecího obvodu.

Jak již bylo zmíněno v kapitole 3.2.2, pracují všechny vybrané moduly s napájecím napětím 3,3 V. Jelikož velikost palubního napětí letadla, na které bude snímač napojen, je 12 V, musí dojít ke stabilizaci napětí z 12 V na požadovanou hodnotu 3,3 V. Pro stabilizaci napětí slouží integrovaný stabilizátor NCP1117. Jedná se o lineární regulátor, který je schopen jak fixního, tak nastavitelného výstupního napětí. Maximální hodnota vstupního napětí je 20 V.



Obrázek 4.2: Použitý stabilizátor NCP1117.¹

Při použití stabilizátoru je nutné si uvědomit, že se chová jako předřadný rezistor, a při stabilizaci napětí dochází k zahřívání součástky v důsledku ztrátového výkonu. Z tohoto důvodu se na stabilizátory přidává kovová destička, která odpadne teplo vyzařuje do okolí. Vhodné je na tuto destičku namontovat chladič, nebo ji připájet k větší kovové ploše, pro lepší odvod tepla.

$$P = (U_{in} - U_{out}) \times I_{out} \quad (4.1)$$

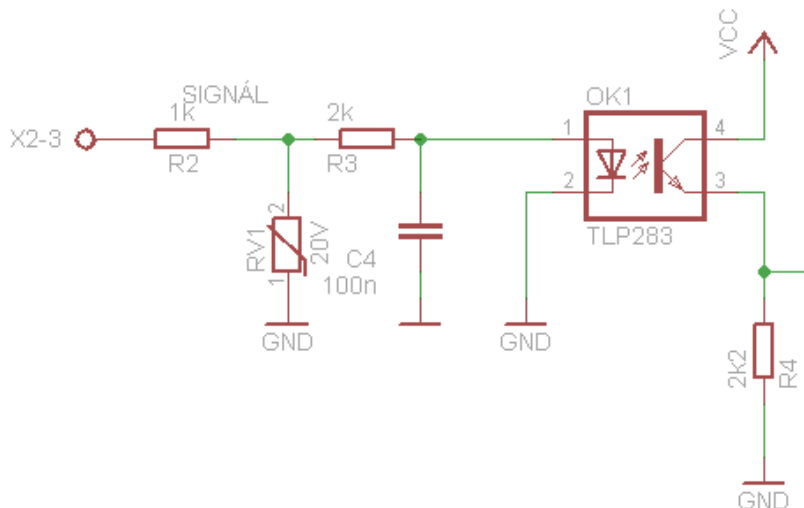
Vztah pro ztrátový výkon je dán rovnicí 4.1, kde U_{in} je vstupní napětí, U_{out} je výstupní napětí a I_{out} je výstupní proud.

Stabilizační obvod pro vstupní napětí (4.1) tvoří stabilizátor NCP1117, který má na vstupu i výstupu připojeny filtrační (bypass) blokovací kondenzátory C2 a C3. Kondenzátor C2 slouží pro stabilitu regulátoru, je-li umístěn několik centimetrů od zdroje proudu. Tento kondenzátor snižuje citlivost obvodu a výrazně zvyšuje vstupní přechodovou odezvu. Kondenzátor C2 by měl být umístěn co nejbliž k regulátoru. Frekvenční kompenzaci pro stabilizátor zajišťuje výstupní kondenzátor C3 a jeho připojení je povinné. Slouží pro zajiš-

¹Převzato z: cz.farnell.com/productimages/standard/en_GB/GE3DPAK05-40.jpg

tění výstupní stability. Usměrňovací Schottkyho dioda D1 chrání obvod proti přepólování a přepětí.

Ochranný obvod



Obrázek 4.3: Schéma zapojení ochranného obvodu s varistorem a optočlenem.

Přepólování napájecího napětí nebo přepětí je velmi častou příčinou zničení zařízení, a proto by měl obvod obsahovat ochranné prvky. Za přepětí se považuje napětí, které přesahuje nejvyšší hodnotu provozního napětí v elektrickém obvodu. Ochranné přepětové prvky jsou zařízení nebo součástky, které slouží k potlačení napětového přepětí. U vytvořeného snímače byly použity prvky, spadající do kategorie jemné přepětové ochrany. Do této kategorie patří například varistor nebo transil.

Ochranné prvky, použité ve snímači, jsou dva - varistor a optočlen. Tyto součástky chrání obvod proti hodnotám impulsů ze zapalovacích cívek motoru, které mohou dosahovat až 100 V.



Obrázek 4.4: Použitý optočlen TLP283² s varistorem.³

Varistor je polovodičový odpor. Velikost jeho odporu je závislá na velikosti napětí. Čím vyšší napětí na varistor přichází, tím menší je jeho odpor. Této vlastnosti je využito v obvodu na obrázku 4.3. Při narůstajícím napětí signálu od cívek odpor na varistoru RV1 klesá, a proud je sveden na ochranný vodič do země.

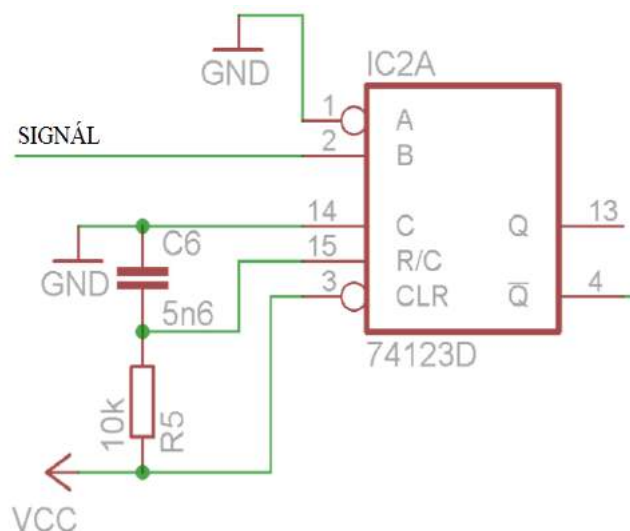
Optočlen je elektronická součástka, která se používá ke galvanickému oddělení dvou elektrických obvodů. Skládá se z LED diody a fotocitlivé polovodičové součástky, větší-

³Převzato z: icbank.com/data/ICBShop/Product/AvagoTechPhotos/516-4-DIP-SMD.jpg

³Převzato z: media.digikey.com/Renders/Cooper%20Bussmann%20Renders/TVSA_0402.jpg

nou fototranzistoru. Přivedením proudu na vstup optočlenu dochází k rozsvícení diody a k otevření fototranzistoru. Otevřený tranzistor způsobí sepnutí výstupního obvodu. Díky galvanickému oddělení lze ovládat obvody, které se mezi sebou liší napětovou úrovní v řádech stovek voltů, a nehrozí poškození oddělené části obvodu.

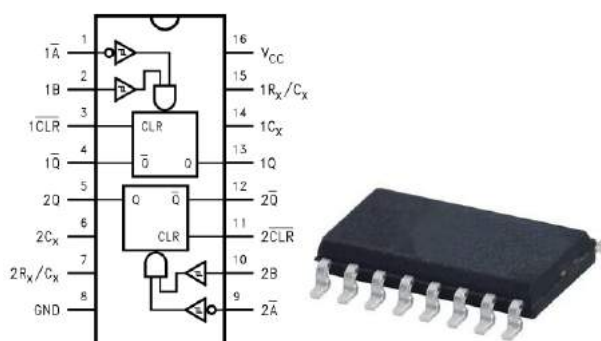
Monostabilní klopný obvod



Obrázek 4.5: Schéma zapojení monostabilního klopného obvodu 74VHC123AM.

Jak bylo popsáno v kapitole 3.2, je signál od zapalovacích cívek motoru letadla přiveden na UART vstup mikrokontroléru. Tvar vyslaného signálu nesplňuje parametry komunikačního protokolu pro asynchronní sériové rozhraní, a tudíž mikrokontrolér nezaregistruje příchozí spojení na svůj UART vstup. Signál z cívek musí být upraven tak, aby splňoval formát datového rámce pro sériové rozhraní (viz. 2.5.1).

Pro vytvoření výstupního signálu požadovaného tvaru z libovolného vstupního pulsu, slouží monostabilní klopný obvod. Jedná se o klopný obvod, který má jeden stabilní stav, v němž může setrvat libovolně dlouho, a jeden přechodný stav, ve kterém setrvává pouze předem určenou dobu.

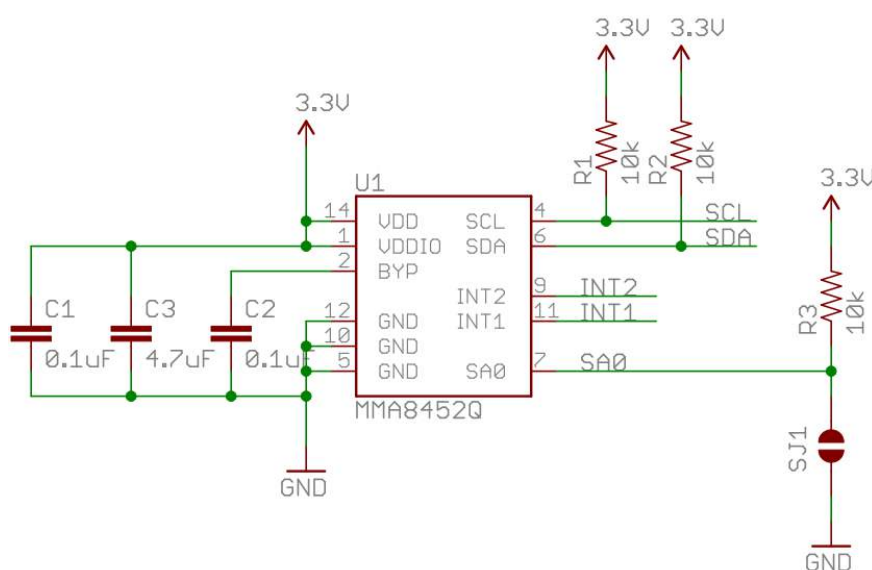


Obrázek 4.6: Použitý monostabilní klopný obvod 74VHC123AM.

Na základě těchto vlastností je vygenerován signál, který ze stavu 1 přejde do stavu 0, v tomto stavu setrvává $52 \mu\text{s}$ a vrátí se do počátečního stavu. Tímto je docíleno vygenerování jednoho datového rámce s hodnotou 255. Zvolený interval a hodnota jsou popsány v kapitole 3.2.2. Interval, ve kterém je monostabilní klopný obvod ve stavu 0, určují hodnoty kondenzátoru C6 (C_x) a odporu R5 (R_x) ve schématu na obrázku 4.6. Délka intervalu (t_w) je dána vzorcem 4.2. Výsledné délky pulsu v intervalu trvajícím $52 \mu\text{s}$, bylo docíleno použitím kondenzátoru s kapacitou $5,6 \text{ nF}$ a odporem o velikosti $10 \text{ k}\Omega$. Byl využit negovaný výstup monostabilního obvodu 74VHC123AM, jelikož počáteční stav obvodu je v 0.

$$t_w = C_x \times R_x \quad (4.2)$$

Zapojení akcelerometru

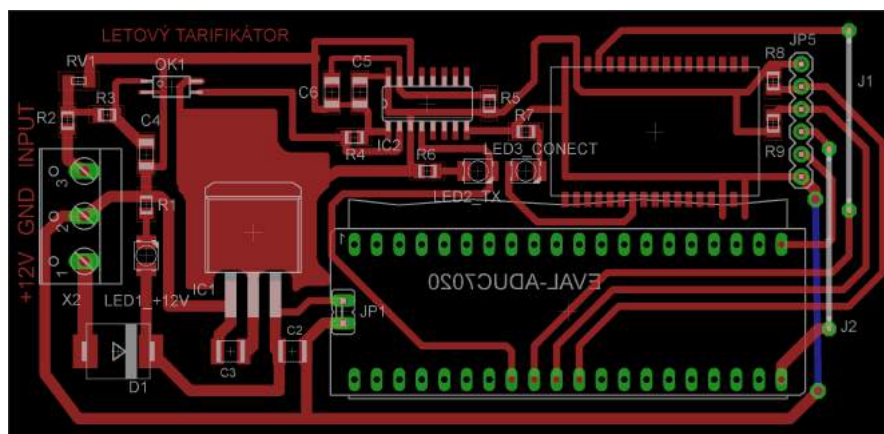


Obrázek 4.7: Schéma zapojení modulu s akcelerometrem MMA8452Q.

Akcelerometr MMA8452Q obsahuje piny pro napájení (VDD), uzemnění (GND), datovou linku sběrnice I²C (SDA), synchronizační linku (SCL), generování vnějšího přerušování (INT1 a INT2) a adresový bit zařízení (SA0). Začátek komunikace na sběrnici I²C je stanoven změnou logické úrovně z 1 do 0, proto se na linkách SDA i SCL nachází pull-up rezistory o velikosti $10 \text{ k}\Omega$, které drží obě linky ve stavu logické 1 (viz. 2.5.4). Každé zařízení na sběrnici I²C je označeno 7-bitovou adresou, díky tomu je určeno, se kterým zařízením se bude komunikovat. Pin SA0 označuje nejnižší bit adresy akcelerometru. Pokud bychom v systému chtěli mít dva akcelerometry MMA8452Q, je možné je rozlišit pomocí tohoto pinu. Pokud na SA0 přivedeme hodnotu 0 (GND), bude adresa zařízení $0x1C$, v případě 1 (VDD), bude adresa $0x1D$. Pro výběr adresy akcelerometru slouží jumper SJ1. V základu je jumper otevřen a na pin SA0 je přivedeno napájecí napětí, adresa zařízení je tedy $0x1D$. Pokud bychom chtěli změnit adresu na $0x1C$, stačí propojit dvě kovové plošky jumperu SJ1. Rozmístění blokovacích kondenzátorů C1, C2 a C3 je provedeno dle dokumentace.

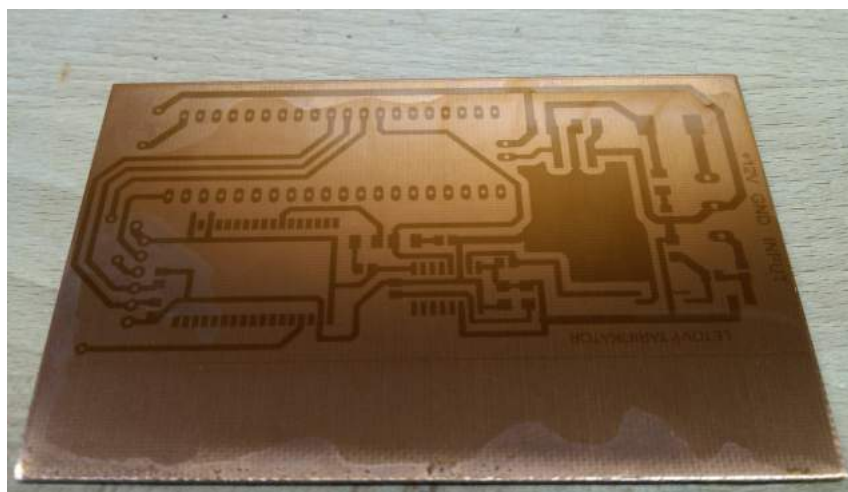
4.1.2 Výroba desky plošného spoje

Měřicí zařízení je realizováno jako elektrický obvod na desce plošných spojů. Jedná se o jednostranný plošný spoj. Obsahuje napájecí obvod, ochranný obvod, obvod pro tvarování signálu, mikrokontrolér, Bluetooth modul a modul akcelerometru. Deska byla navrhována ve vývojovém prostředí EAGLE.



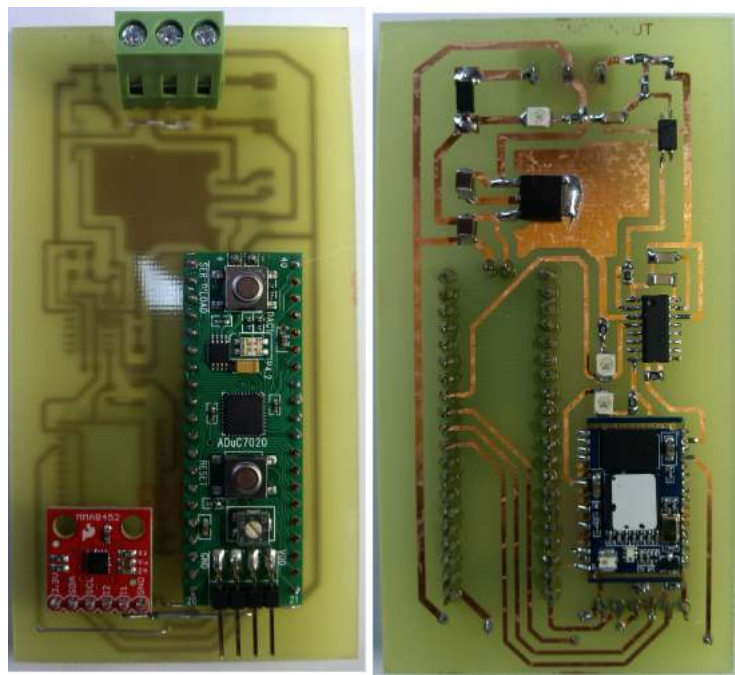
Obrázek 4.8: Celkový návrh měřicího zařízení v prostředí EAGLE.

Šablonu desky, kde jsou obsaženy pouze jednotlivé spoje, vytiskneme na papír. Papír v potřebných místech nastříkáme zpruhledňujícím sprejem a přiložíme na měděnou stranu desky. Takto nachystanou desku vložíme pod halogenovou výbojku. Světlo z lampy naruší strukturu lepidla mědi, a tam, kde měď není zakrytá šablonou, se spojí. Po zhruba pěti minutách desku opláchneme a vložíme do přípravku pro pozitivní fotoemulzi. Jedná se o 1,5% roztok hydroxidu sodného. Tímto docílíme vyvolání spojů na desce (obrázek 4.9).



Obrázek 4.9: Deska po použití fotoemulze.

Deska se následně vloží do leptacího roztoku s chloridem železitým (30-50%). Zde dojde k vyleptání osvětlené mědi s narušeným lepidlem. Na desce zůstane jen měď, která tvoří cesty plošného spoje. Po osazení plošného spoje jednotlivými součástkami, je deska vložena do ultrazvukové čističky, která odstraní nečistoty vzniklé od tavidla při pájení.



Obrázek 4.10: Výsledné měřicí zařízení (pohled shora a zdola).

4.2 Softwarová část řešení

Softwarová část řešení obsahuje popis implementace mikrokontroléru a mobilní aplikace. Řídící mikrokontrolér snímacího zařízení komunikuje s akcelerometrem MMA8452Q přes jeho sériovou sběrnici I²C. Tato komunikace vyžaduje naprogramovat v procesoru funkce pro zápis a čtení dat akcelerometru. Procesor při zapnutí zařízení nakonfiguruje akcelerometr na požadované parametry, a zbytek času tráví v dotazovací smyčce, ve které čte hodnoty v registrech pro jednotlivé osy zrychlení. Asynchronně ke čtení dat akcelerometru, může přijít na UART vstup mikrokontroléru signál od zapalovacích cívek motoru. Je tedy nutné, kromě sběrnice I²C, naprogramovat i funkci rozhraní UART (vstup i výstup). Pokud dojde k přijetí dat z akcelerometru nebo motoru letadla, vyvolá se na vstupních pinech procesoru přerušování. Data budou zpracována v implementované obslužné přerušovací rutině. Správně přijatá data procesor odešle na své výstupní UART rozhraní.

Mobilní aplikace využívá pro spojení s měřicím zařízením vestavěný Bluetooth modul v mobilním telefonu. Nejdříve se provede zapnutí a inicializace bezdrátového adaptéru. Následně se vytvoří spojení, kdy mobilní telefon pracuje jako klient, který se připojuje na server (modul BTM-112). Po vytvoření spojení, začne aplikace přijímat data, odeslaná ze snímače. Z přijatých dat od zapalovacích cívek se vypočte celková frekvence otáčení motoru. Pokud frekvence dosáhne určité hodnoty, lze konstatovat, že letadlo vzletlo. Rozhodnutí o stavu letadla se, kromě frekvence motoru, provede i na základě hodnot zrychlení. Splní-li se podmínky pro vzlet od otáček a zrychlení, spustí se měřič, který změří celkovou dobu letu.

Celkově tedy:

- Software mikrokontroléru inicializuje a řídí přenos s akcelerometrem, Bluetooth modulem a vstupním signálem. Implementuje funkce pro čtení/zápis na svých sériových rozhraních a funkcí obsluhy přerušování.

- Program mobilní aplikace naváže spojení v Bluetooth síti s modulem BTM-112. Přijme údaje o zrychlení a frekvenci motoru. Data vyhodnotí a na jejich základě změří dobu letu.

4.2.1 Implementace mikrokontroléru

Následující část práce je zaměřena na popis implementace nejdůležitějších funkcí řídicího procesoru snímacího zařízení, pro komunikaci s akcelerometrem, Bluetooth modulem a signálem z cívek. Mikroprocesorový modul s procesorem ADuC7020 je programován ve vývojovém prostředí Keil μ Vision 3 v jazyce C s využitím patřičných registrů a knihoven.

Inicializace komunikačních rozhraní a přerušení

Důležitým krokem v počáteční fázi implementace procesoru, je nastavení jeho frekvence. Na této hodnotě závisí například rychlost sériové komunikace nebo generování vnitřního přerušení od časovače. Mikroprocesor ADuC7020 disponuje vnitřním oscilátorem o frekvenci 32,768 kHz. Tento oscilátor je určen k poskytnutí hodinového signálu celému mikrokontroléru. Pro dosažení požadované frekvence je použita technika Fázového závěsu (PLL), kdy je frekvence oscilátoru vynásobena 1275-krát. Tímto je dosažena maximální hodnota frekvence procesoru 41,78 MHz. Maximální frekvence může být dále dělena pro získání dalších dílčích frekvencí. Celkově je k dispozici osm frekvencí od 41,78 MHz po 0,326 MHz.

K určení požadované frekvence procesoru slouží sada Power control registrů (POWCON). Zápisem hodnot do těchto registrů se určí mód procesoru (Sleep, Active atd.) a velikost děliče hodinového signálu (CD). Zápisová sekvence musí mít vždy následující tvar:

```
POWKEY1 = 0x01;
POWCON = user_value; // Vybraná hodnota z tabulky v dokumentaci
POWKEY2 = 0xF4;
```

V práci je frekvence procesoru nastavena na jeho maximální hodnotu 41,78 MHz. Jelikož mikroprocesor komunikuje s modulem BTM-112 pomocí rozhraní UART a s akcelerometrem MMA8452Q přes rozhraní I²C, je nejdříve nutné správně nakonfigurovat vstup/výstupní piny a nastavit požadovanou rychlost komunikace. Celková komunikace je řízena pomocí přerušení od jednotlivých periférií. Dále je tedy zapotřebí vybrat požadovanou přerušovací rutinu (Interrupt Service Routine), která bude vykonána v případě přerušení, a povolit přerušení od daného typu sběrnice.

```

1 /**
2 * Function for UART and I2C configuration
3 **/
4 void init_i2c_uart(){
5
6     GP1CON = 0x2211; // Set UART tx & rx pins on P1.0 and P1.1 = 0x0011
7                     // and I2C on P1.2(SCL) and P1.3(SDA)      = 0x2200
8
9     // Start setting up I2C at 100kHz
10    I2C1CFG = 0x82; // Master Enable, Enable Generation of Master Clock
11    I2C1DIV = 0xCFCF; // 0xCFCF = 100kHz, 0x3232 = 400kHz
12
13    // Start setting up UART at 19200 bauds
14    COMCON0 = 0x080; // Setting DLAB
15    COMDIV0 = 0x44; // Setting DIV0 and DIV1 to DL calculated 19200
16    COMDIV1 = 0x000;
17    COMCON0 = 0x007; // Clearing DLAB
18    COMIEN0 = 0x01; // Enable UART RX interrupt
19
20    // Interrupt config
21    IRQ = My_IRQ_Function; // Interrupt routine
22    IRQEN = SM_MASTER1_BIT + UART_BIT; // Interrupt enabled
23    GP4DAT = 0x04000000; // P4.2 LED is turned on
24 }

```

Obrázek 4.11: Ukázka inicializačního programu pro rozhraní UART a I²C.

Vývojová deska obsahuje celkem 40 vstup/výstupních pinů, které jsou rozděleny do pěti portů (Port 0 až 4). Pro zajímavost, všechny piny obsahují interní pull-up rezistory, které mohou být v případě potřeby použity. Konfigurace rozhraní se provádí zápisem do konfiguračního registru GPxCON. Vstupy i výstupy obou použitých sběrnic jsou umístěny v Portu 1, konfigurován bude registr GP1CON (obrázek 4.11).

Mikrokontrolér disponuje dvěma samostatnými, totožnými I²C rozhraními (I2C0 a I2C1), kde každé z nich podporuje jak roli Master, tak i Slave. V práci je využito I2C1. Při komunikaci na sběrnici I²C je mikroprocesor v roli Master (viz. 2.5.4), což znamená, že vytváří a řídí synchronizační signál SCL. Frekvence hodinového signálu je nastavena na standardních 100 kHz, avšak ADuC7020 podporuje i Fast mode přenos (400 kHz).

$$f_{SERIALCLOCK} = \frac{f_{UCLK}}{(2 + DIVH) + (2 + DIVL)} \quad (4.3)$$

Výsledná frekvence přenosu vychází z rovnice 4.3, kde f_{UCLK} je frekvence hodin před děličem, DIVH je horní byte hodnoty periody a DIVL je spodní byte periody. Pro frekvenci 100 kHz se musí obě hodnoty DIVH i DIVL rovnat 0xCF. Nastavení těchto vlastností dochází zápisem do konfiguračního registru I2C1CFG a registru I2C1DIV.

Tovární nastavení Bluetooth modulu udává použitou rychlost komunikace na 19,200 baudů, je tedy zapotřebí nastavit stejnou hodnotu i v případě UART rozhraní mikroprocesoru, aby docházelo ke správnému vzorkování dat na obou stranách, a ke správným výsledkům při I/O operacích. Požadovaná rychlost je vypočtena dle vzorce:

$$BaudRate = \frac{41.78Mhz}{2^{CD} - 16 \times 2 \times DL} \quad (4.4)$$

Baud Rate	CD	DL	Actual Baud Rate	% Error
9600	0	0x88	9600	0
19,200	0	0x44	19,200	0
115,200	0	0x0B	118,691	3
9600	3	0x11	9600	0
19,200	3	0x08	20,400	6.25
115,200	3	0x01	163,200	41.67

Tabulka 4.1: Jednotlivá nastavení Baud Rate generátoru pro rozhraní UART.

Sériová komunikace je konfigurována pomocí COMxxxx registrů. V případě rychlosti UART rozhraní jsou to registry COMDIV0 a COMDIV1.

Posledním krokem inicializace je nastavení přerušení, kdy dochází k volání obslužné rutiny, která zpracovává přijaté data na vstupních rozhraních. Procesor využívá dva typy přerušení IRQ a FIQ. První z nich je klasické přerušení, které slouží k obsluze obecných událostí, ať už externích nebo interních. Druhý typ je rychlé přerušení (Fast Interrupt reQuest). Hlavní rozdíl mezi těmito dvěma typy přerušení je událost, na kterou se vztahují. FIQ přerušení jsou navržena tak, aby byla použita pro rychlou obsluhu vybraných (programátorem) kritických přerušení, kdy FIQ má větší prioritu než IRQ. Typicky programátor definuje přerušení jako IRQ, ale pro přerušení s nízkou dobou odezvy, které musí být vyřízeno prioritně, jej lze nastavit jako FIQ. Jádro ARM procesoru neposkytuje samostatný přerušovací vektor pro jednotlivé periferie (časovač, UART atd.). Na všechna přerušení se musí nahlížet stejně, a je tedy jen na programátorovi, jak je rozdělí. V případě kódu na Obrázku 4.11 jsou přerušení od rozhraní UART i I²C označena jako IRQ, zápis do registru IRQEN. Funkce, která se provede v případě přerušení, je uvedena v registru IRQ. Rozsvícení LED diody na pinu P4.2 pak značí konec inicializace.

Komunikace na sběrnici I²C

Akcelerometr MMA8452Q komunikuje s procesorem přes své I²C rozhraní. Pro získání dat z jeho registrů musí být v procesoru naimplementovány funkce pro čtení a zápis. Komunikace na sběrnici I²C se drží standartu popsaného v kapitole 3.2.2. Procesor disponuje několika registry pro čtení a zápis na rozhraní I²C. Pro vystavení adresy cílového zařízení a určení čtecí/zapisovací operace slouží registr I2C1ADR. Pro čtení/zápis registry I2C1MRX/I2C1MTX, a nakonec registr I2C1CNT, kterým se určí počet přijatých bytů od Slave zařízení.

Dle kapitoly 3.2.2 je čtecí (Read) operace dosaženo Opakovaným Startem (Repeated Start), kdy po vystavení adresy registru, ze kterého chce Master zařízení číst, nenásleduje ukončení spojení (Stop Condition), ale znova se generuje začátek přenosu (Start Condition). Důležitým faktem, který je uveden v dokumentaci procesoru je to, že mikroprocesor ADuC7020 nepodporuje generování Opakovaného Startu. Pokud by program obsahoval následující posloupnost příkazů:

- adresa Slave zařízení + WriteBit (0)
- adresa registru, ze kterého chceme číst
- počet přijatých bytů
- adresa Slave zařízení + ReadBit (1)

- čtení přijatých dat

Processor pak rozdělí program na dvě samostatné operace (zápis a čtení). Zároveň nedojde k provedení ani jedné operace, jelikož v případě zápisu chybí zapisovaná data, a v případě čtení není uveden cílový registr. Processor tedy do modulu akcelerometru nic nezapíše, ani z něj nepřijme, z důvodu neplatného formátu komunikace na sběrnici. Je tedy nutné tuto situaci vyřešit, aby mohla komunikace s akcelerometrem vůbec probíhat.

Existují dvě řešení, jak zvládnout situaci, kdy neumožnění generování Opakovaného Startu ze strany procesoru v podstatě znemožňuje komunikaci z vybraným akcelerometrem. Jedno řešení je čistě softwarové, druhé je kombinací mezi hardwarem a softwarem.

První možností je nevyužívat registry procesoru připravené pro komunikaci na sběrnici I²C a celou komunikaci naprogramovat manuálně. Signály pro zahájení nebo ukončení přenosu tedy nejsou generovány automaticky procesorem, ale záleží pouze na programátorovi, kdy a jakých hodnot budou linky SCL a SDA nabývat. Příklad této metody je možné vidět u obvodu CN0133. Jedná se o aplikační ukázkou zapojení procesoru ADuC7024 a akcelerometru ADXL345 v režimu Master a Slave. Celkově je tato možnost řešení náročná na synchronizaci dat a hodin, proto je v práci použita druhá varianta řešení.

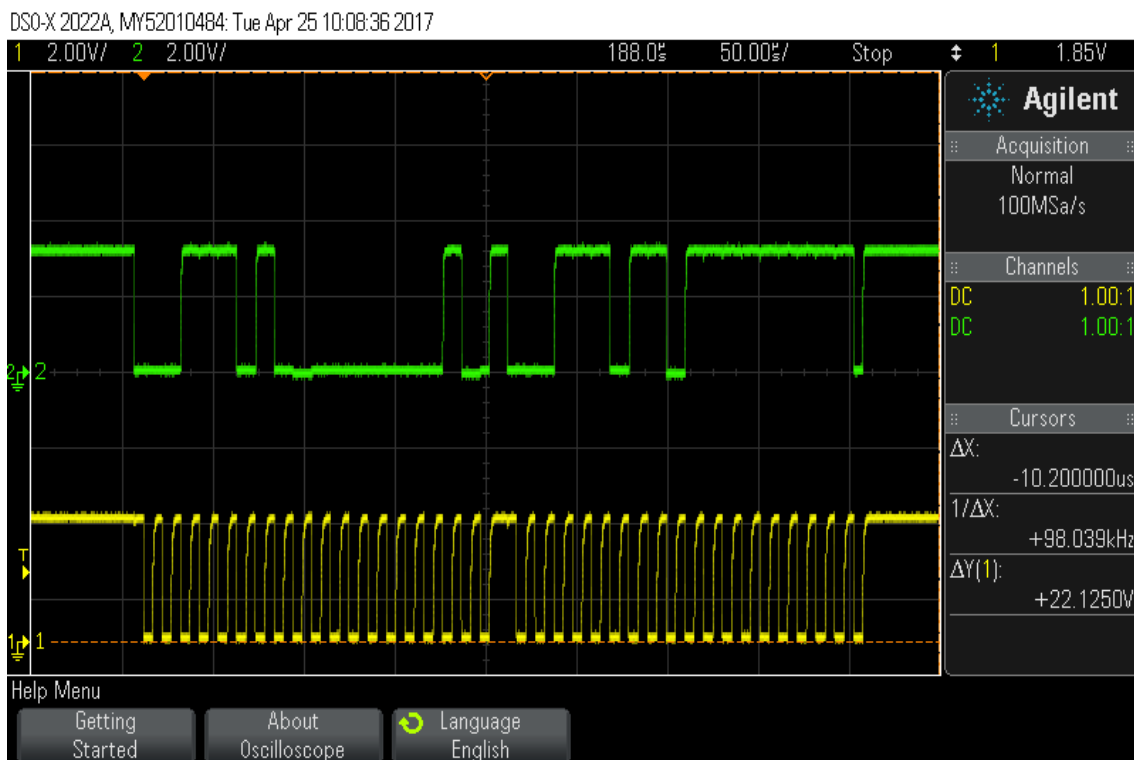
Tato varianta řešení je opět popsána výrobcem procesoru. Spočívá v přidání rezistoru o velikosti 4,7 k Ω na datovou linku SDA a rezistoru o velikosti 20 k Ω na linku SCL. Program pro čtení je nutné upravit následovně:

```

1 /**
2  *   devAddr = Device address + WriteBit(0), regAddr = Register address
3  *   Function reads 1 byte from Device, Data read in Interrupt routine
4  **/
5 void i2c_byte_read(unsigned char devAddr, unsigned char regAddr){
6
7     I2C1CCNT = 0x0; // Sets Start/Stop condition counter value to 0
8     I2C1ADR = devAddr; // Accelerometer(slave) adress(Master Write)
9     I2C1MTX = regAddr; // Accelerometr register address
10    while ((I2C1FSTA & 0x30) != 0x00) {;} // Wait for the Tx FIFO to empty
11    I2C1CNT = 0x0; // Number of bytes to be read from slave - 1
12    I2C1ADR = devAddr + 1; // Accelerometer(slave) adress(Master Read)
13    I2C1CCNT = 0x80; // Set the Start/Stop counter to re-enable
14
15    delay(DELAY_TIME);
16 }
```

Obrázek 4.12: Ukázka funkčního řešení pro Opakovaný Start na ptocesoru ADuC7020.

Důležitým bodem je nastavení hodnoty registru I2C1CCNT. Tento registr slouží k nastavení zpoždění mezi SCL a SDA. Přidáním rezistorů a zvýšením prodlevy je možné dosáhnout „schování“ Stop Condition na konci odesílaného bytu z registru I2C1MTX. Následné odeslání nové adresy vyvolá Start Condition, a na sběrnici se tento stav bude jevit jako Repeated Start. Dosažením tohoto stavu, lze nyní správně komunikovat a číst data z akcelerometru.



Obrázek 4.13: Ukázka generování Repeated Start pro čtení dat z akcelerometru (Horní: SDA, Dolní: SCL). Čtení registru obsahující zrychlení v ose X.

Na Obrázku 4.13 lze vidět na SCL lince (dolní) generování Opakovaného Startu v podobě setrvání úrovně v logické 1, a negenerování Stop Condition na SDA (horní). Absence automatického generování Opakovaného Startu je nejspíš založena na faktu, že procesor ADuC7020, zapojený ve Slave režimu na I²C sběrnici, pro čtení ze svých registrů tuto podmínku nepotřebuje.

Komunikace na sběrnici UART

Sériové rozhraní UART slouží pro komunikaci mezi procesorem a Bluetooth modulem BTM-112. Na vstup UART rozhraní taktéž přicházejí pulsy od zapalovacích cívek motoru. Je tedy nutné, podobně jako u sběrnice I²C, implementovat operace pro čtení a zápis na tomto rozhraní. Komunikace na sériové sběrnici UART je oproti rozhraní I²C podstatně jednodušší, jelikož nevyžaduje synchronizační signál. Frekvence přenosu musí být zajištěná dopředu a shodná na obou komunikujících zařízeních.

Podobně jako sběrnice I²C, má i rozhraní UART na procesoru vyčleněné registry pro čtení COMRX a zápis COMTX. Řízení komunikace probíhá kontrolováním stavu registru COMSTA0, kdy v případě příjmu dat dojde k nastavení jeho LSB, a při odesílání lze kontrolovat hodnotu bitu na pozici 0x20. K vynulování stavových bitů dojde čtením/zápisem jednotlivých registrů.

Přenos dat po sériové lince mezi monostabilním klopným obvodem (MKO), procesorem a Bluetooth modulem vypadá tak, že procesor očekává přerušení na svém UART Rx pinu od MKO. Následně procesor přečte výslednou hodnotu zapsanou v registru COMRX a na výstupní Tx pin, který je napojen na modul BTM-112, odešle zprávu ve formátu "(O:znak)".

Konfigurace akcelerometru MMA8452Q

Při zapnutí měřícího zařízení musí procesor nakonfigurovat modul akcelerometru tak, aby vůbec zasílal měřené hodnoty zrychlení. Dále je nutné nastavit formát výstupních dat a rozsah, ve kterém se zrychlení měří.

Modul akcelerometru nabízí velkou spoustu konfiguračních registrů pro pokročilé nastavení. Od nastavení detekce volného pádu po formát výstupních dat atd. Díky tomu má uživatel vysokou kontrolu nad veličinami, které bude akcelerometr snímat.

V základu je modul nastaven do SNADBY módu, kdy je umožněna změna hodnot konfiguračních registrů. Akcelerometr začne měřit zrychlení ve chvíli, kdy je uveden do tzv. Aktivního módu (ACTIVE mode). V tomto režimu není možné měnit obsah jeho registrů (kromě registru CTRL_REG1), je tedy možné registry pouze číst. Registr CTRL_REG1 je hlavní konfigurační registr, který umožňuje zápis nových hodnot i v ACTIVE režimu. Kromě registrů obsahující hodnoty zrychlení, jsou hodnoty všech registrů ponechány při přechodu z ACTIVE na STANDBY. Přepnutím mezi módy dojde zápisem na LSB registru CTRL_REG1. V tomto registru se dále nachází nastavení automatického probuzení akcelerometru ze spánkového režimu, frekvence získání nových dat, redukce šumu a nastavení formátu dat.

```
1  /**
2  * Function for Accelerometr configuration
3  * Accelerometr is set to ACTIVE mode
4  */
5  void init_acc(){
6
7      // Acc to STANDBY mode, in order to change registres
8      i2c_byte_write(ACC_ADDR, CTRL_REG1, STANDBY_MODE);
9      // Scale to 8g
10     i2c_byte_write(ACC_ADDR, XYZ_DATA_CFG, SCALE_G8);
11     // Enable Portrait/Landscape detection
12     i2c_byte_write(ACC_ADDR, PL_CFG, PL_ENABLE);
13     // Acc to ACTIVE mode, Enable FAST READ = F_READ is set, only 8-bit
14     // data
15     i2c_byte_write(ACC_ADDR, CTRL_REG1, ACTIVE_MODE);
16 }
```

Obrázek 4.14: Ukázka konfigurace akcelerometru. Nastavení rozsahu, režimu dat, detekce natočení.

V práci je akcelerometr nastaven na rozsah do ± 8 g, autodetekci natočení (Portrait/Landscape detection), a režimu 8 bitových dat (FAST READ). Citlivost modulu je rozlišena v počtu jednotek za jedno g. Pro rozsah do ± 2 g je citlivost 1024/g, pro ± 4 g je 512/g a pro ± 8 g je citlivost nastavena na 256/g. Pro maximální měřený rozsah zrychlení je tedy citlivost akcelerometru 4x horší než pro minimální rozsah, což neumožňuje snímání tak přesných hodnot, avšak při dosahování větších zrychlení, například v letadle, není tato vlastnost tak kritická.

Měřené hodnoty zrychlení jsou uloženy jako 12 bitová čísla ve dvojkovém doplňku ve dvou 8 bitových registrech. Horní byte je uložen v registru (pro osu X) OUT_X_MSB, a dolní byte v registru OUT_X_LSB (Tabulka 4.2).

0x01: OUT_X_MSB: X_MSB Register (Read Only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
XD11	XD10	XD9	XD8	XD7	XD6	XD5	XD4

0x02: OUT_X_LSB: X_LSB Register (Read Only)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
XD3	XD2	XD1	XD0	0	0	0	0

Tabulka 4.2: Formát uložení 12 bitových dat zrychlení pro osu X v registrech.

Z tabulky 4.2 pro rozložení jednotlivých bitů v registrech pro měřené 12 bitové hodnoty zrychlení vyplývá, že po přečtení a uložení správné hodnoty, například do 4 bytového integeru, je zapotřebí provést sérii bitových posunů. Důležité je taky zachování znaménkového bitu.

Druhý bit registru CTRL_REG1 slouží pro nastavení formátu 8 bitových dat, kdy jsou hodnoty zrychlení uloženy jen v horním byte registru pro danou osu. Postačí tedy číst pouze jeden registr pro získání hodnoty zrychlení, avšak s menší přesností, přesněji o faktor 16.

Data vyčtená z registrů pro zrychlení jednotlivých os jsou uložena v programové proměnné XYZ_DATA [], ze které je vypočteno celkové zrychlení. Jelikož je zrychlení reprezentováno jako číslo s plovoucí řádovou čárkou, musí být v programu převedeno na odpovídající datový typ se správným rozsahem do ± 8 g. Výsledek je vynásoben gravitační konstantou, aby výsledné číslo odpovídalo jednotkám v m/s. Hodnoty jsou ve výstupním formátu "[X:cislo;Y:cislo;Z:cislo]" odeslány v ISR na UART rozhraní mikroprocesoru, na které je připojen Bluetooth modul (viz. obrázek 4.15).

```

1 /**
2  * Function computes acceleration for each axis(x,y,z) and send result to
3  * UART
4  * Result is in m/s units(not in g!)
5  * Function is called in ISR(Interrupt Service Routine)
6  */
7
8 void get_xyz_data(){
9
10     char str[255];
11     int x, y, z;
12     float cx, cy, cz;
13
14     x = XYZ_DATA[0];
15     y = XYZ_DATA[1];
16     z = XYZ_DATA[2];
17     cx = ((float) x / (float)(1<<7) * (float)(G8)) * GRAVITY_CONST;
18     cy = ((float) y / (float)(1<<7) * (float)(G8)) * GRAVITY_CONST;
19     cz = ((float) z / (float)(1<<7) * (float)(G8)) * GRAVITY_CONST;
20
21     sprintf(str, "[X:%f;Y:%f;Z:%f]", cx, cy, cz);
22     SendStrUart(str);
23 }

```

Obrázek 4.15: Zpracování dat přijatých z akcelerometru. Převedení na datový typ float.

V Tabulce 4.3 lze vidět, že rozdíly měřených hodnot mezi 12 bitovým a 8 bitovým formátem dat se pohybují v řádech setin, a při použití v letadle jsou tyto rozdíly zanedbatelné.

12-bit Data	Range ±2g (1 mg)	Range ±4g (2 mg)	Range ±8g (3.9 mg)
0111 1111 1111	1.999g	+3.998g	+7.996g
0111 1111 1110	1.998g	+3.996g	+7.992g
...
0000 0000 0001	0.001g	+0.002g	+0.004g
0000 0000 0000	0.0000g	0.0000g	0.0000g
1111 1111 1111	-0.001g	-0.002g	-0.004g
...
1000 0000 0001	-1.999g	-3.998g	-7.996g
1000 0000 0000	-2.0000g	-4.0000g	-8.0000g
8-bit Data	Range ±2g (15.6 mg)	Range ±4g (31.25 mg)	Range ±8g (62.5 mg)
0111 1111	1.9844g	+3.9688g	+7.9375g
0111 1110	1.9688g	+3.9375g	+7.8750g
...
0000 0001	+0.0156g	+0.0313g	+0.0625g
0000 0000	0.000g	0.0000g	0.0000g
1111 1111	-0.0156g	-0.0313g	-0.0625g
...
1000 0001	-1.9844g	-3.9688g	-7.9375g
1000 0000	-2.0000g	-4.0000g	-8.0000g

Tabulka 4.3: Rozdíl v přesnosti mezi 8 a 12 bitovým formátem dat z akcelerometru.

Obsluha přerušení

Jednou z nejdůležitějších funkcí v implementaci procesoru je program pro obslužnou přerušovací rutinu. Tato funkce zajišťuje komunikaci mezi procesorem, akcelerometrem, signálem od cívek motoru a Bluetooth modulem. Čtení z akcelerometru je implementováno jako Poolling v hlavní smyčce procesoru.

Funkce pro obsluhu přerušení (ISR) je vyvolána po přijetí přerušení od dvou zdrojů. Prvním je úspěšné přijetí jednoho bytu dat na sběrnici I²C od akcelerometru, druhé přerušení způsobí přijetí dat na rozhraní UART od zapalovacích cívek.

V případě obsluhy přerušení od sběrnice I²C je nejdříve zkontrolován stavový registr přerušení IRQSTA. Pokud má vektor přerušení nastavený patřičný bit, je dále zkontrolován další registr I2C1MSTA, který obsahuje další dodatečné informace. Pokud tento registr obsahuje hodnotu 0x8, je přerušení vyvoláno úspěšným přečtením jednoho bytu dat na sběrnici I²C. Příznak přerušení je nulován automaticky po přečtení dat z registru I2C1MRX (viz. 4.16).

```

1  /**
2  *  Interrupt Service Routine
3  **/
4  void My_IRQ_Function(){
5
6      // Interrupt from I2C Bus
7      if ((IRQSTA & SM_MASTER1_BIT) != 0){
8
9          // Master Recieve IRQ
10         if((I2C1MSTA & 0x8) == 0x8){
11             // Read X_MSB
12             if (AXIS_INDEX == 0) XYZ_DATA[AXIS_INDEX] = I2C1MRX;
13             // Read Y_MSB
14             if (AXIS_INDEX == 1) XYZ_DATA[AXIS_INDEX] = I2C1MRX;
15             // Read Z_MSB + reset AXIS_INDEX + Compute accleration
16             if (AXIS_INDEX == 2){
17                 XYZ_DATA[AXIS_INDEX] = I2C1MRX;
18                 get_xyz_data();
19                 AXIS_INDEX = 0;
20             }
21         }
22     }
23 }

```

Obrázek 4.16: Ukázka přerušovací rutiny pro zpracování přerušení na sběrnici I²C.

Sběrnici I²C se posílají aktuální hodnoty zrychlení, a na UARTu se přijímá signál ze zapalovacích cívek motoru letadla v podobě jednoho UART rámce. V případě přijetí dat (znaku) na Rx pinu UART rozhraní, se data přečtou a jsou odeslána na výstupní Tx pin. Data z akcelerometru jsou čtena, dokud nejsou k dispozici hodnoty ze všech tří os. Tato data jsou zpracována ve funkci `get_xyz_data()` (Obrázek 4.15). Všechna přijatá data jsou odeslána na Bluetooth modul.

4.2.2 Implementace mobilní aplikace

Tato podkapitola obsahuje popis implementace mobilní aplikace pro platformu Android. Je zde uveden popis uživatelského rozhraní, komunikace v Bluetooth síti, struktur jednotlivých aktivit, vyhodnocení stavu letadla na základě zrychlení a frekvence otáček motoru, a způsob zobrazení zrychlení reprezentovaný pomocí grafů.

Mobilní aplikace slouží pro komunikaci se snímacím zařízením, a jejím hlavním úkolem je změřit dobu letu pro tarifkaci ultralehkého letadla. Doba letu je určena na základě vztahů mezi frekvencí otáček motoru a měřeného zrychlení letadla.

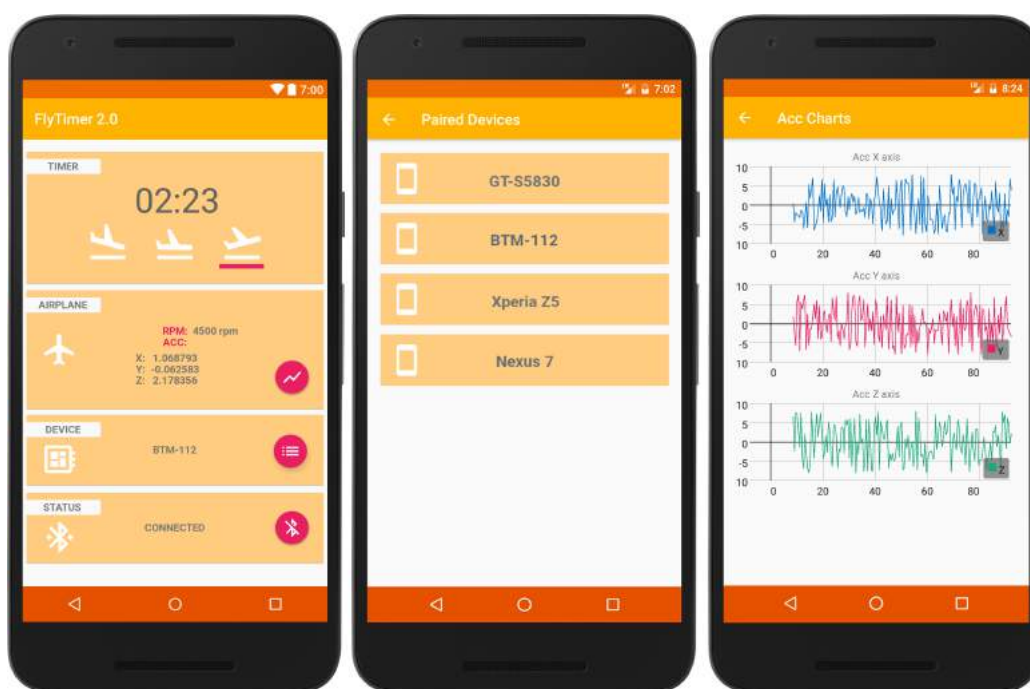
Aplikace komunikuje skrze vestavěný Bluetooth modul v mobilním telefonu. Při komunikaci v Bluetooth síti se mobilní aplikace chová jako klient (Slave), který se připojuje na server (Master). V roli serveru je Bluetooth modul BTM-112. Modul je viditelný ostatním zařízením a jako server očekává příchozí spojení od okolních Bluetooth klientů. Po spárování a navázání spojení začne server klientovi zasílat data, která přicházejí na vstup jeho sériového rozhraní. Mobilní aplikace tato data přijme a zpracuje.

Výsledné GUI aplikace

Grafické rozhraní mobilní aplikace, vytvořené v práci, se snaží dodržovat zásady Material Designu, které jsou popsány v kapitole 2.4.1, a rozvrh funkčních prvků dle 3.2.3. Styly

jednotlivých obrazovek (Activit) jsou popsány v jejich XML souborech. Konkrétně jsou to soubory `activity_main.xml`, `activity_chart.xml` a `activity_device_list.xml`. Zde jsou definovány použité komponenty a jejich atributy, jako rozložení, jejich šířka a výška, barva, případně popisek, nebo ikona.

Grafické uživatelské rozhraní hlavní obrazovky aplikace je rozděleno na čtyři hlavní sekce, a skládá se ze dvou grafických komponent. První komponenta je `cardView`, jedná se o rámeček s kulatými rohy vrhající stíny, který se většinou využívá při tvorbě položek seznamu. Jednotlivé `cardView` od sebe odlišují informační a ovládací sekce aplikace. Druhou komponentou je plovoucí tlačítko, které po kliknutí provede vyznačenou akci. Ikonky tlačítek byly zvoleny tak, aby uživatel co nejintuitivněji rozpoznal jejich funkci.



Obrázek 4.17: Výsledný vzhled uživatelského rozhraní mobilní aplikace.

První informační sekce *Timer* obsahuje časovač, který měří dobu letu a indikuje polohu letadla. Druhá sekce *Airplane* obsahuje informace o snímaných veličinách, konkrétně zrychlení ve všech osách a údaje o otáčkách motoru letadla. Zde je umístěno tlačítko pro náhled grafů, které po kliknutí, zobrazí aktivitu vykreslující grafy, na základě měřeného zrychlení. Polední dvě sekce jsou věnovány Bluetooth komunikaci se snímačem, V sekci *Device* je tlačítko pro zobrazení seznamu spárovaných Bluetooth zařízení, ze kterého si uživatel zvolí konkrétní zařízení. Tlačítko v sekci *Status* slouží pro navázání spojení se zvoleným zařízením.

Aplikace se celkově skládá ze tří obrazovek. Grafické rozhraní hlavní obrazovky je popsáno výše, dále je zde obrazovka vykreslující grafy z měřených hodnot zrychlení a poslední obrazovka vytváří a zobrazuje seznam spárovaných zařízení. Pro všechny obrazovky je použit `RelativeLayout`, `ScrollView` a spolu s obecným rozložením komponent namísto fixního má tato kombinace za následek přizpůsobení obrazovky na natočení telefonu.

Inicializace Bluetooth adaptéru

Pro příjem data odeslaných měřicím zařízením, je třeba, aby se mobilní aplikace připojila do Bluetooth sítě, která je vytvářena modulem BTM-112. Platforma Android nabízí možnost připojit se do Bluetooth sítě, která umožňuje bezdrátovou výměnu dat s ostatními Bluetooth zařízeními. Aplikační framework poskytuje přístup k funkcionalitě Bluetooth skrze Android Bluetooth API. Díky tomuto rozhraní se aplikace mohou připojovat k Bluetooth zařízením a dále využívat následující funkce:

- hledat ostatní Bluetooth zařízení v dosahu
- zobrazit nabídku spárovaných zařízení
- vytvořit RFCOMM kanály
- přenos dat
- spravovat více připojených zařízení

Aby dvě Bluetooth zařízení mohly mezi sebou posílat data, je zapotřebí vytvořit komunikační kanál. Tento kanál je vytvořen pomocí párovacího procesu, kdy jedno ze zařízení musí být viditelné pro příchozí spojení od ostatních zařízení. V našem případě zařízení, které je vždy viditelné, je Bluetooth modul BTM-112, a mobilní aplikace inicializuje spojení s tímto modulem. Po přijetí párovacího požadavku si obě zařízení vymění bezpečnostní klíč (klíč = 1234), a tímto je proces párování dokončen. Zařízení, které zahájilo komunikaci, uvolní na konci spojení komunikační kanál. Obě zařízení zůstanou spárované i po ukončení spojení, aby se mohla v budoucnu připojit automaticky. Proces párování není v aplikaci implementován, jelikož stejnou funkcionalitu poskytuje operační systém.

Aplikace pro systém Android často požadují nejrůznější povolení pro zamýšlený chod aplikace. Mezi tyto povolení patří například oznámení o poloze, přístup ke kontaktům, fotoaparátu atd. Aby aplikace mohla využívat funkce vestavěného Bluetooth modulu, musí mít v hlavním manifestu (`AndroidManifest.xml`) deklarované patřičné povolení. Povolení pro Bluetooth adaptér má následující tvar:

```
1 <uses-permission android:name="android.permission.BLUETOOTH" />
2 <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

Obrázek 4.18: Povolení pro Bluetooth adaptér v hlavním manifestu aplikace.

Před samotnou Bluetooth komunikací je třeba zkontrolovat, jestli mobilní telefon opravdu disponuje Bluetooth modulem, a pokud ano, zda-li je zapnut. Pokud modul není podporován, jsou všechny další funkce pozastaveny, a aplikace nebude jevit známky funkčnosti. V případě, že modul není aktivní, lze vyžádat jeho zapnutí, aniž by bylo zapotřebí vypínat aplikaci. Všechny úkony spojené s Bluetooth adaptérem jsou prováděny třídou `BluetoothAdapter`, která reprezentuje modul v zařízení.

```

1  bltAdapter = BluetoothAdapter.getDefaultAdapter();
2  // Kontrola zda zarizeni obsahuje Bluetooth modul
3  if (bltAdapter == null) {
4      Toast.makeText(this, "This device does not support Bluetooth.",
5          Toast.LENGTH_SHORT).show();
6      return;
7  }
8  else{
9      BluetoothOn();
10 }

```

Obrázek 4.19: Kontrola přítomnosti Bluetooth adaptéru v zařízení a jeho zapnutí.

Funkce `BluetoothOn()` (Orázek 4.19) zkontroluje stav adaptéru pomocí metody `isEnabled()`. Pokud je výsledkem funkce hodnota `false`, je modul vypnutý. Vytvořením nové třídy `Intent` s konstruktorem `BluetoothAdapter.ACTION_REQUEST_ENABLE`, která je předána jako parametr funkce `startActivityForResult()`, dojde k vytvoření nové aktivity, v tomto případě dialogového okna, které uživatele vyzve k zapnutí Bluetooth adaptéru.

```

1  /**
2   * Funkce zapne Bluetooth modul
3   **/
4  public void BluetoothOn() {
5      if (!bltAdapter.isEnabled()) {
6          enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
7          startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
8      }
9  }

```

Obrázek 4.20: Funkce pro zapnutí Bluetooth adaptéru.

Konstanta `REQUEST_ENABLE_BT` je lokálně definovaná číselná hodnota, která musí být větší než 0. Systém tuto konstantu poskytne nazpět v implementaci funkce `onActivityResult()`, která se vyvolá při spuštění dané aktivity. Kromě spouštěcí konstanty obsahuje návratová hodnota funkce i `resultCode` a případná dodatečná data. Pokud uživatel povolí zapnutí Bluetooth adaptéru, obdrží konstantu `RESULT_OK`, v opačném případě `RESULT_CANCELED`. Implementace inicializace a komunikace pomocí Bluetooth se nachází v souboru `MainActivity.java`.

Zobrazení spárovaných zařízení

Pilot si při ovládání aplikace vybere, k jakému Bluetooth zařízení se připojí. Z toho důvodu je vytvořena aktivita, která mu jednotlivá již spárovaná zařízení zobrazí přehledně v seznamu.

Zobrazení zařízení, která jsou s telefonem již spárovaná, je implementováno v souboru `DeviceListActivity.java`. Pro získání seznamu všech spárovaných zařízení slouží funkce `getBondedDevices()` od třídy `BluetoothAdapter`. Funkce poskytne množinu `BluetoothDevice` objektů, které reprezentují jednotlivá zařízení. Tyto objekty obsahují například MAC adresy zařízení, typ, nebo jejich jméno.

```

1 // Ziskani jmen jednotlivych sparovanych Bluetooth zarizeni
2 pairedDevices = bltAdapter.getBondedDevices();
3 if (pairedDevices.size() > 0){
4     for(BluetoothDevice bltDevice : pairedDevices){
5         bltDeviceList.add(bltDevice.getName());
6     }
7 }

```

Obrázek 4.21: Program pro získání jmen spárovaných Bluetooth zařízení.

Seznam dostupných zařízení, u kterých byl proveden párovací proces, se uživateli zobrazí při spuštění aktivity `DeviceListActivity`. Kliknutí na položku seznamu se zvoleným zařízením, dojde k vytvoření nového objektu `Intent`. Jedná se o operaci, která většinou souvisí s výměnou informací mezi aktivitami. Pomocí `Intent` lze spouštět nové aktivity, nebo mezi nimi vyměňovat data. V podstatě se jedná o pasivní datovou strukturu, obsahující abstraktní popis operace, kterou provede. V aplikaci je využita její funkce `putExtra()`, která slouží pro vložení dodatečných dat, při provádění operace. Data musí obsahovat jednoznačný identifikátor, aby byl příjemce schopný data najít a zpracovat. V implementaci je příjemce zprávy hlavní aktivita (`MainActivity`), která obdrží jméno zařízení, které zvolil uživatel, a podle identifikátoru `selectedDevice` data přečte a zpracuje funkcí `getStringExtra()`.

```

1 /**
2  * Prijme zpravu od DeviceListActivity
3  */
4  @Override
5  protected void onActivityResult(int requestCode,int resultCode,Intent ms){
6      if ((requestCode == 100) && (resultCode == RESULT_OK)) {
7          deviceNameText.setText(ms.getStringExtra("selectedDevice"));
8      }
9      super.onActivityResult(requestCode, resultCode, ms);
10 }

```

Obrázek 4.22: Příjem dat ve struktuře `Intent` v hlavní aktivitě.

Vytvoření spojení

V klasických síťových aplikacích se vyžaduje implementace jak strany serveru tak klienta. V práci ovšem jako server funguje Bluetooth modul BTM-112, který má svou část již připravenou, a proto implementace serverové části aplikace není potřeba. Mobilní aplikace tedy plní roli klienta, který se připojuje na server. Tato část je v aplikaci implementována.

Server a klient se považují za připojené, pokud mají oba vytvořený socket na stejném RFCOMM kanálu. Síťový socket je reprezentován třídou `BluetoothSocket`. Server má vytvořený vlastní socket, na kterém naslouchá na příchozí spojení od klienta, který se připojuje na danou MAC adresu serveru. Po úspěšném vytvoření spojení si obě zařízení mohou vyměňovat data pomocí input a output streamů.

Aby se klient úspěšně připojil na zařízení, které přijímá spojení na otevřeném serverovém socketu, musí nejdříve vytvořit `BluetoothDevice` objekt, který představuje vzdálené zařízení. S tímto objektem je vytvořen socket na RFCOMM kanálu. Připojení klienta na server je implementováno ve třídě `ConnectThread`, která dědí od třídy `Thread`, a funguje jako samostatné vlákno.

Po vytvoření objektu `BluetoothDevice` reprezentující vzdálené serverové zařízení, získáme požadovaný socket voláním funkce `createRfcommSocketToServiceRecord(UUID)`, která inicializuje objekt socketu a umožní klientovi se připojit. Konstanta UUID je 128 bitů dlouhý neměnný univerzální jedinečný identifikátor (Immutable Universally Unique Identifier). Server musí obsahovat stejné UUID, které zasílá klient. Jelikož modul BTM-112 pracuje s SPP profilem, používá obecné SPP UUID `00001101-0000-1000-8000-00805F9B34FB`. Inicializace socketu a RFCOMM kanálu je implementována v konstruktoru třídy.

```
1 public ConnectThread(BluetoothDevice device) {
2
3     BluetoothSocket tmp = null;
4     mmDevice = device;
5
6     try {
7         tmp = mmDevice.createRfcommSocketToServiceRecord(MY_UUID);
8
9     } catch (IOException e) {
10        Log.e("Nepovedlo se vytvorit Socket - ConnectThread");
11    }
12    mmSocket = tmp;
13 }
```

Obrázek 4.23: Vytvoření socketu a RFCOMM kanálu mezi serverem a klientem.

Samotné spojení je navázáno pomocí socket funkce `connect()`. Jakmile klient zavolá tuto funkci, systém prohledá svůj lokální SDP server, zda-li neobsahuje zařízení se stejným UUID. Pokud je vyhledání úspěšné, a server zařízení potvrdí spojení, začnou obě zařízení sdílet komunikační kanál a funkce `connect()` vrátí hodnotu `true`. Jestli dojde k selhání spojení, nebo vyprší timeout pro navázání spojení (zhruba 12 vteřin), funkce vyvolá IO výjimku. Jelikož je volání funkce `connect()` blokující, mělo by se připojení na server provádět v samostatném vlákne, které je oddělené od hlavního UI vlákna. Před navázáním spojení, by se vždy mělo zkontrolovat, jestli mobilní zařízení neprovádí vyhledávání ostatních zařízení v dosahu, což může mít za následek zpomalení, nebo selhání připojení. Připojení na server je implementováno v třídní funkci `run()`. Vytvoření a spuštění třídy `ConnectThread` je provedeno ve funkci `ConnectToDevice()`.

Příjem a zpracování dat

Po úspěšném vytvoření spojení na server, obsahuje každé zařízení socket, kterým lze přijímat nebo odesílat data. Mobilní aplikace nemá žádné informace, které by serveru posílala, tudíž bude implementovat pouze funkce potřebné pro čtení. Třída `ConnectedThread` pro správu připojení je implementována obdobně, jako třída pro vytvoření spojení.

Rozdělení komunikace na příjem a odesílání dat odpovídá použitým funkcím socketu. Pro čtení je potřeba získat vstupní stream (objekt `InputStream`), pro odesílání zase výstupní (objekt `OutputStream`). Vstupní stream je získán funkcí `getInputStream`, a nabízí čtecí funkci `read()`, která čte vstupní data a vrací je v datovém typu `byte`.

```

1 public ConnectedThread(BluetoothSocket socket){
2     mmSocket = socket;
3     InputStream tmpIn = null;
4
5     try {
6         tmpIn = socket.getInputStream();
7     } catch (IOException e) {
8         Log.e("Nepovedlo se vytvorit InputStream - ConnectedThread");
9     }
10    mmInStream = tmpIn;
11 }

```

Obrázek 4.24: Vytvoření vstupního streamu pro čtení dat.

Množství dat, přečtených ze vstupního streamu, záleží na velikosti bufferu, do kterého jsou data ukládána. V aplikaci má tento buffer velikost 1 byte, pro snadnější kontrolu příchozích zpráv a manipulaci s daty. Aplikace přijímá ze snímače dva druhy zpráv, a každá má svůj specifický formát. První zpráva obsahuje údaje o zrychlení letadla ve tvaru "[X:cislo;Y:cislo;Z:cislo]", druhá poskytuje informaci o otáčkách motoru ve formátu "(O:cislo)". Na základě přijatého znaku označujícího začátek jednotlivých zpráv ("["a "("), se rozhodne, do kterých proměnných se bude zbytek zprávy ukládat. Pokud se jedná o data se zrychlením, jsou jednotlivé byty ukládány do proměnné `accMsg` (typ `String`). V případě dat s otáčkami je pouze inkrementována hodnota čítače `rpmCounter`.

Čtecí a zapisovací operace mají své implementační omezení. Obě operace by měly běžet v samostatném vlákne, jelikož jsou obě blokující. Funkce `read()` blokuje provádění dokud jsou ve vstupním streamu k dispozici data. Funkce `write()` obvykle blokující nebývá, ale může taktéž blokovat program v případě, že vzdálené zařízení neprovádí čtecí operaci dostatečně rychle, a dojde k naplnění mezipaměti.

Třída `ConnectThread` i `ConnectedThread` obsahuje funkci `cancel()`. Tato metoda umožňuje okamžité ukončení Bluetooth spojení zavřením vytvořeného socketu. Funkce `cancel()` by měla být volána, pokud se aplikace rozhodne ukončit Bluetooth spojení.

Komunikace s hlavním UI vláknem

Je-li pro danou aplikaci vytvořen proces, je jeho hlavní vlákno určeno k obsluze fronty zpráv, která spravuje objekty nejvyšší úrovně (např. aktivity), a jakákoliv okna, která mohou vytvořit. V případě, že aplikace obsahuje další samostatná vlákna, nemají tato vlákna přístup k hlavnímu UI a jakýkoliv pokus o manipulaci s grafickými prvky by skončil ukončením aplikace. Komunikaci mezi vedlejšími vlákny a hlavním UI vláknem zajišťuje třída `Handler`.

`Handler` umožňuje odesílat a zpracovávat zprávy a Spouštěcí (`Runnable`) objekty, které jsou spojené s frontou zpráv daného vlákna. Každý `Handler` je spjatý pouze s jedním vláknem. Od chvíle jeho vytvoření bude doručovat zprávy do fronty daného vlákna a vykonávat je v pořadí v jakém přišly. Objekt `Handler` má dvě hlavní využití. První způsob využití je budoucí plánování a zpracování zpráv, druhé využití je vkládání akcí, které budou provedeny v jiném vláknem.

Plánování zpráv se provádí například pomocí příkazu `post()`, `postAtTime()`, `postDelayed()`, `sendMessage()`, `sendEmptyMessage()` a další. Metody s prefixem `post` umožňují do fronty vkládat `Runnable` objekty, zatímco funkce s `send` obsahují balíček dat, která jsou zpracována funkcí `handleMessage()`. Při vkládání pomocí `post` nebo `send` lze dále určit

dobu, kdy budou jednotlivé zprávy zpracovány. Zpracování je možné provést okamžitě, jakmile zpráva dorazí do fronty, nebo zpracování odložit o určenou dobu. K tomu slouží například třídy `Timers`.

Aplikace třídu `Handler` využívá při komunikaci z vlákna `ConnectThread` a `ConnectedThread`. `Handler` je vytvořen v hlavní funkci `onCreate()`, která se nachází v `MainActivity`. Tímto je docíleno zpětného spojení do hlavního UI vlákna, z vláken pro připojení k Bluetooth modulu. Je využito funkcí `sendEmptyMessage()`, v kterých se zasílá kód, spojený s určitou vzniklou situací v daném vlákně. Kód zprávy je reprezentován číselnou konstantou a může mít následující hodnoty:

- `BT_DISCONNECTING` (-1) - Proces odpojování ze sítě Bluetooth
- `BT_CONNECTING` (0) - Proces připojování do sítě Bluetooth
- `BT_CONNECTED` (1) - Zařízení se úspěšně připojilo
- `BT_DISCONNECTED` (2) - Zařízení se úspěšně odpojilo
- `ACC_MSG_FULL` (3) - Přijmutí kompletní zprávy o zrychlení

Zprávy jsou zaslány hlavnímu UI vlákně, zde jsou zpracovány funkcí `handleMessage()`, kde na každou zprávu je volána příslušná funkce. V případě zpráv s kódem -1 až 2 se jedná o akce spojené s GUI prvky aplikace. Zpráva s kódem 3 zpracuje data z akcelerometru.

```
1 // Handler pro získání stavu z Bluetooth vláken
2 mHandler = new Handler(){
3     @Override
4     public void handleMessage(Message msg){
5         byte[] readBuf = (byte[])msg.obj;
6         switch(msg.what){
7             case BT_DISCONNECTING:
8                 SetDisconnecting();
9                 break;
10            case BT_CONNECTING:
11                SetConnecting();
12                break;
13            case BT_CONNECTED:
14                SetConnected();
15                break;
16            case BT_DISCONNECTED:
17                SetDisconnected();
18                break;
19            case ACC_MSG_FULL:
20                GetXYZ();
21                break;
22            default:
23                break;
24        }
25    }
26 };
```

Obrázek 4.25: Zachycení zpráv od připojovací a čtecí třídy v hlavním UI vlákně.

Vyhodnocení stavu letadla

Tarifkace letadla je vyhodnocena na základě doby letu. Ze získaných údajů o otáčkách a zrychlení je třeba odvodit aktuální stav letadla. Při vzletu letadla se zapne měření doby letu, které se po přistání ukončí.

Měření frekvence otáček motoru letadla je prováděno na základě počtu přijatých pulsů za známý časový interval. Jedná se tedy o přímé měření frekvence, popsané v kapitole 2.2.1. Bluetooth modul na mobilní telefon zasílá zprávu ve formátu popsáném v 4.2.1. Po obdržení zprávy mobilní aplikace inkrementuje hodnotu čítače `rpmCounter`. Při startu aplikace je spuštěno nové vlákno `rpmFreqHandler`, které každé 3 vteřiny zavolá funkci `GetRPM()`. Ve funkci se přečte aktuální hodnota čítače, která se vynásobí dvaceti, aby frekvence otáčení byla v počtu otáček za minutu. Na základě konzultace s vedoucím práce byla stanovena hodnota otáček motoru při vzletu letadla na 4500 ot/min po dobu minimálně desíti vteřin. Pokud frekvence otáček splní tuto podmínku, je nastaven příznak `rpmTakeOff`. Příznak se nastavuje ve funkci `rpmWatcher()`, která je volána každou vteřinu v samostatném vláknu. Implementace funkcí a vláken pro sledování a vyhodnocení frekvence otáčení motoru jsou obsaženy v souboru `MainActivity.java`.

```
1 /**
2  * Funkce pro sledovani poctu otacek motoru, pokud je po dobu 10s
3  * frekvence otacek vice nez 4500 letadlo vzleta
4  **/
5 public void rpmWatcher() {
6     if (Integer.parseInt(rpmTextView.getText().toString()) >= RPM_TRESHOLD
7     ){
8         rpmTimerCounter++;
9
10        if (rpmTimerCounter == RPM_TAKEOFF_TIME){
11            rpmTakeOff = true;
12        }
13    }
14    else{
15        rpmTimerCounter = 0;
16    }
17 }
```

Obrázek 4.26: Funkce pro sledování stavu frekvence otáček a vzletu letadla.

Zrychlení je vyhodnocováno pokaždé, když je přijata zpráva ve formátu popsáném v 4.2.1. Z přijaté zprávy jsou vyextrahována data pro jednotlivé osy ve funkci `GetXYZ()`. Na tato data je použit softwarový vysokofrekvenční filtr, který odfiltruje konstantní gravitační zrychlení. Alternativou by bylo nastavit high-pass bit v konfiguraci akcelerometru. Před vzletem letadlo popojíždí po dráze, a vlivem nízké hmotnosti působí na hodnoty zrychlení nerovnosti terénu. Proto jsou tyto výkyvy ignorovány, a pro vzlet se berou do úvahy až větší rozdíly mezi hodnotami v jednotlivých osách. Pokud hodnota na ose Z prudce vzroste (vzlet), zatímco na ose Y dochází ke změnám (pohyb letadla), nastaví se příznak `accTakeOff` pro signalizaci vzletu letadla. Postup pro přistání je opačný.

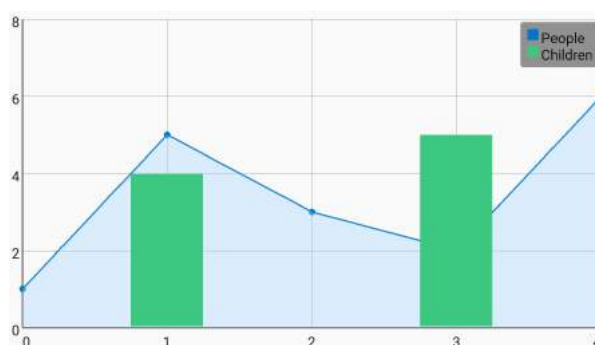
Vlákno `flightWatcherHandler` kontroluje proměnnou `rpmTakeOff` a `accTakeOff`. Pokud mají obě hodnotu `true`, spustí se časomíra na hlavní obrazovce mobilní aplikace, která změří dobu letu do přistání letadla. Při samotném letu nedochází k velkým změnám zrychlení, a proto lze přistání letadla chápat jako změnu z klidových hodnot na záporné. Nastane-

li tento stav, časomíra se zastaví. Tímto je změřena doba letu na základě frekvence otáček motoru letadla a jeho zrychlení.

Vykreslení grafů

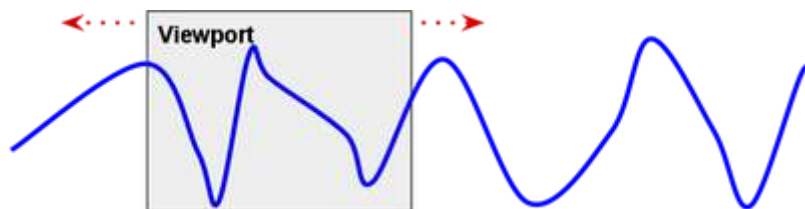
Doplňující funkcí aplikace je aktivita s vykreslováním grafů podle hodnot měřeného zrychlení v jednotlivých osách. Aktivita se spustí kliknutím na tlačítko v sekci Airplane na hlavní obrazovce aplikace. Účel této activity je vizuálně reprezentovat měnící se hodnoty zrychlení v závislosti na pohybu letadla.

K vykreslení grafů, reprezentujících zrychlení v čase, byla použita open-source knihovna `GraphView`. Jedná se o knihovnu pro systém Android, která umožňuje programově vytvářet grafy. Výhodou knihovny je jednoduchost použití a velká míra přizpůsobení. S `GraphView` lze vytvářet spojnicové, sloupcové a bodové grafy. Vytváření grafů pomocí knihovny `GraphView` je implementováno v souboru `ChartActivity.java`.



Obrázek 4.27: Ukázka sloupcového a bodového grafu vykresleného pomocí knihovny `GraphView`.⁵

Jak název knihovny napovídá, `GraphView` dědí vlastnosti od třídy `View`, která je jedna ze základních komponent při tvorbě uživatelského rozhraní. Třída `View` zabírá obdélníkový prostor na obrazovce telefonu, a je zodpovědná za vykreslování. `View` v knihovně představuje tzv. *Viewport*, jedná se o viditelnou část grafu, která je zobrazena na obrazovce (Obrázek 4.28). Jelikož knihovná třída `View` funguje naprosto stejně jako její klasická varianta, je možné vlastnosti a grafické prvky grafu nastavit v patřičném XML souboru, nebo programově v implementaci dané aktivity.



Obrázek 4.28: Příklad *Viewport* při zobrazování delšího grafu.⁶

Ve výchozím nastavení se *Viewport* automaticky synchronizuje s daty, je tedy možné vykreslit všechna data. Vykreslovací okno zvládne změnu měřítka i zvětšení grafu.

⁵Převzato z: www.android-graphview.org/showcase/

⁶Převzato z: www.android-graphview.org/basics-of-graphview/

Důležitým prvkem v knihovně `GraphView` je třída, popisující body grafu. Jedná se o strukturu typu seznam, obsahující souřadnice X a Y všech bodů. V práci jsou použity spojnicové grafy, proto byla použita třída `LineGraphSeries`.

Grafy, vytvořené v aplikaci, zobrazují hodnoty zrychlení, které přichází ze snímače kontinuálně v reálném čase. Je tedy zapotřebí data vykreslovat taktéž v reálném čase. S knihovnou `GraphView` je možné měnit data za běhu aplikace, a tedy grafy v reálném čase vytvářet. Pro tento účel slouží dvě funkce třídy `Series`. První je metoda `resetData()`, která současné data v grafu vymaže a nahradí je novými. Druhá funkce `appendData()` přidá do současné množiny dat novou dvojici souřadnic X a Y, a pokud je povolen příznak `scrollToEnd`, přesune `Viewport` automaticky na nově vykreslená data. V práci je použita druhá metoda, jelikož první způsob neumožňuje pozorovat změny v grafu, právě díky vymazání předchozích hodnot.

Jelikož jsou údaje o zrychlení dostupné v jiné aktivitě (`MainActivity`), než ve které se vykreslují grafy, je nutné tato data předat z hlavní aktivity do aktivity pro vykreslení grafů. Existuje několik způsobů, jak předávat data mezi aktivitami. Jednou z možností je využít funkci třídy `Intent` (viz. 4.2.2), která je schopna zaslat balík dat do jiné aktivity. Nevýhodou této metody je, že po zaslání dat, je nutné spustit aktivitu, které byla data zaslána. Tímto způsobem nelze zasílat kontinuální data, pokud by aktivita příjemce neobsahovala kontrolu příznaku, na základě kterého by se aktivita zobrazila, což je velice neefektivní. Další možností pro výměnu dat je využít jedno z úložišť systému Android. Ten nabízí přístup k internímu, externímu a sdílenému úložišti v telefonu. Interní úložiště je dostupné pouze pro danou aplikaci a není-li stanoveno jinak, může zde číst a zapisovat pouze jedna konkrétní aplikace. Externí úložiště ukládá data veřejně na externí nebo interní paměťové medium v telefonu (např. sd karta). Sdílené úložiště (`Shared Preferences`) je API pro ukládání relativně malých objemů dat ve formátu dvojice klíč-hodnota. Tato data mohou být jak veřejná tak privátní. Při použití `Shared Preferences` je vytvořen XML soubor, o který se programátor kromě čtení a zápisu, nemusí starat. Jedná se o nejjednodušší způsob ukládání dat v systému Android.

Hlavní aktivita, s daty o zrychlení, vytvoří sdílený soubor využitím `Shared Preferences` pomocí funkce `getSharedPreferences()`. Data jsou zapsána funkcí `put()` ve tvaru zrychlení-čas, a v aktivitě s grafy jsou data čtena funkcí `get()`, a následně vykreslena do příčného grafu. Zaslání i příjem dat běží v samostatných vláknech v daných aktivitách.

Kapitola 5

Testování

Kapitola obsahuje popis testování a ověření funkčnosti komunikace měřícího zařízení mezi procesorem, akcelerometrem, vstupním signálem a Bluetooth modulem. Popis fungování snímače je v kapitole 4.1. Při testování byl použit zdroj napětí o velikosti 12 V. Stejně hodnoty dosahuje i palubní napětí letadla.

5.1 Tvorba testovacího signálu

Pro ověření funkčnosti měření a snímání otáček motoru letadla, je třeba nasimulovat vstupní signál od zapalovacích cívek motoru, který je přiveden na vstup měřícího zařízení. Vytvoření takového signálu, je docíleno pomocí vývojového modulu s procesorem ADuC7024. Jedná se o ARM procesor, patřící do stejné rodiny jako použitý ADuC7020 (viz. 3.2.2).

Signál je vytvořen programově procesorem a odeslán na jeden z jeho výstupních pinů. Z tohoto pinu je přiveden na vstup měřícího zařízení. Implementace tvorby signálu se skládá ze tří kroků. První částí je změna z aktivní logické úrovně 1 na hodnotu 0. Druhým krokem je setrvání v tomto stavu na předem stanovený interval. V posledním kroku je hodnota signálu zvednuta zpět do aktivní úrovně. Tímto je docíleno vytvoření obdélníkového signálu s frekvencí jakou si stanoví programátor, a který bude simulovat frekvenci příchozích pulsů za minutu.

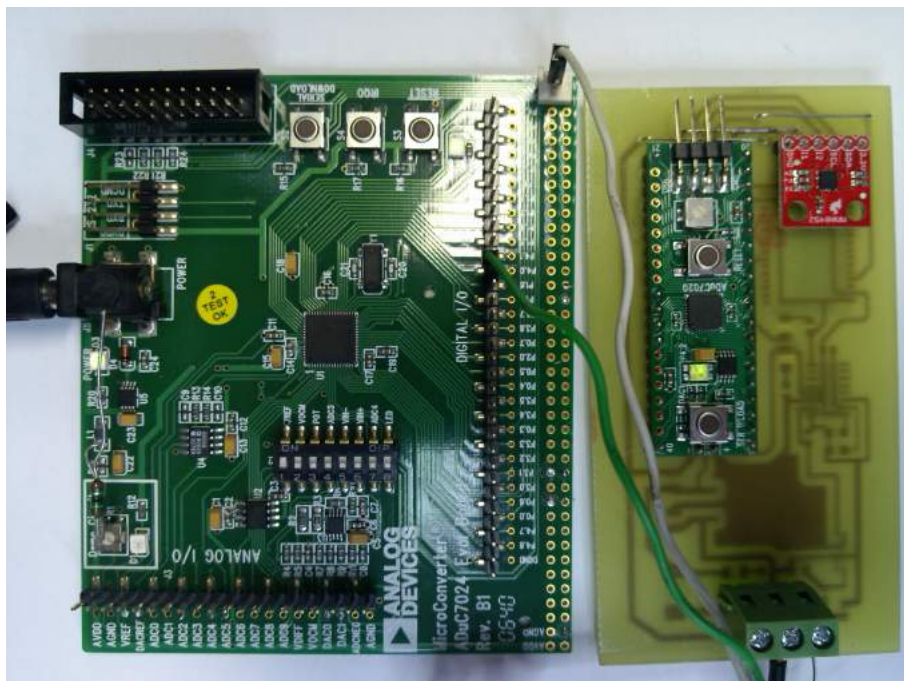
K vytvoření signálu byl použit vnitřní čítač procesoru, který při přetečení vyvolá přerušení. Doba přetečení čítače závisí na nastavené frekvenci procesoru, hodnotě v modulu registru a nastavení předděličky (viz. rovnice 5.1).

$$t_{TOF} = \frac{MOD_{reg} \times Prescaler}{F_{source}} \quad (5.1)$$

Interval přetečení čítače (t_{TOF}) se získá vynásobením modulu registru (MOD_{reg}) hodnotou nastavenou v předděličce a tento výsledek je podělen velikostí frekvence procesoru (F_{source}). Doba přetečení čítače zde znamená dobu, jakou bude signál v úrovni 1 nebo 0. Pokud čítač přeteče dvakrát, je tím vytvořena jedna perioda signálu. V testovacím programu je nastavena hodnota předdělení na 256, frekvence procesoru je 41,78 MHz.

Na začátku programu se nastaví pin P4.0 procesoru na úroveň 1. Čítač je nastaven na velice malou hodnotu, aby co nejrychleji přetekl a vyvolalo se přerušení. V přerušovací rutině je znegována hodnota na pinu P4.0 a čítač modulu registru je nastaven na hodnotu, odvozenou z požadované frekvence pulsů za minutu. Další přerušení opět zneguje hodnotu na pinu a vytvoří zbytek periody signálu.

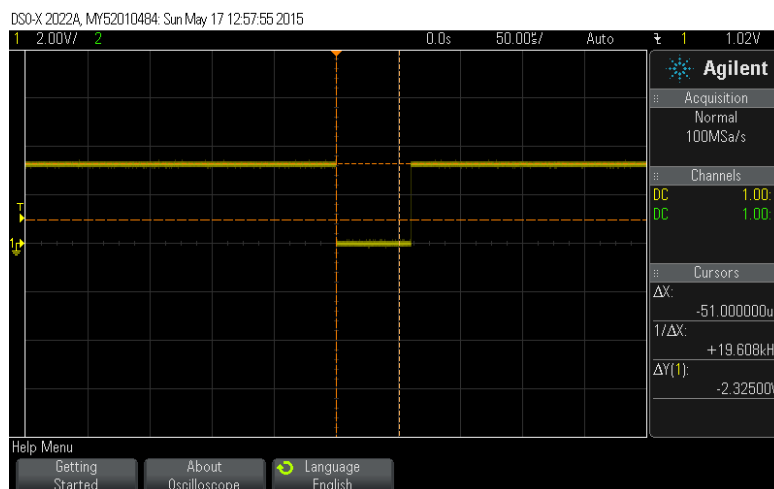
Zapojení vytvořeného měřícího zařízení s vývojovým kitem, bylo realizováno dle následujícího obrázku:



Obrázek 5.1: Zapojení procesoru ADuC7024 se snímcem. Vytvořený signál je z pinu P4.0 přiveden na vstup měřícího zařízení.

5.2 Testování mobilní aplikace

Jak bylo psáno v kapitole 4.1, je signál z cívek přiveden v měřícím zařízení na vstup monostabilního klopného obvodu. Obvod na základě příchozího signálu vygeneruje signál nový, který splňuje formát sériové komunikace pro rozhraní UART (viz. 2.5.1 a 3.2.2). Je tedy nutné zkontrolovat, zda-li se tak stalo. Kontrola se provedla osciloskopem.



Obrázek 5.2: Vygenerovaný signál z monostabilního klopného obvodu. 1 dílek = 50 μ s.

Vygenerovaný signál (obrázek 5.2) splňuje požadavky UART komunikace. Start bit vznikne změnou z úrovně 1 na 0, kde zůstane minimálně 52 μ s. Hodnota ostatních bitů už není důležitá, a proto se může úroveň vrátit zpět na 1.

Další testování probíhalo nastavováním různých frekvencí příchozích pulsů od vývojového modulu a jejich zobrazováním v mobilní aplikaci. Frekvence byla měřena v intervalu 1000 až 10 000 otáček za minutu. Hodnoty byly měřeny pětkrát a zprůměrovány. Mobilní aplikace byla spuštěna na mobilním telefonu Huawei G600.

Huawei G600 - [ot/min]										
Zadaná F	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
Měřená F	996	2000	2988	3990	4988	5988	6984	7992	8984	9978

Tabulka 5.1: Výsledky měření frekvence otáček motoru aplikací v telefonu Huawei G600.

Z tabulky výsledků měření (tabulka 5.1) vychází průměrná chyba měření na 0,2%. Což odpovídá ± 1 odesílanému znaku. Chyba nejspíše vzniká čtením Bluetooth streamů, které jsou bufferované.

Spolu s frekvencí otáček byla taky ověřena správnost údajů z akcelerometru. Měřené hodnoty zrychlení z modulu akcelerometru, byly srovnány s hodnotami, které měří vestavěný akcelerometr v mobilním telefonu. K tomu bylo nutné naprogramovat mobilní aplikaci, která využívá vestavěný akcelerometr.

Jelikož procesor vysílá měřené hodnoty zrychlení a frekvence otáček přes svoje UART rozhraní, je možné využít vyvedené piny procesoru a komunikaci si ověřit ještě zobrazením v počítači. K tomu je zapotřebí počítač se sériovým portem, nebo převodníkem na USB a program pro zobrazení dat příchozích na daný port.

Kapitola 6

Závěr

Cílem této práce bylo vytvořit systém, který provádí tarifkaci ultralehké letadlo na základě provozních parametrů. Výsledný systém je tvořen vestavěným elektronickým zařízením, které měří zrychlení pomocí modulu s akcelerometrem, a frekvenci motoru pomocí signálů od zapalovacích cívek. Systém dále obsahuje mobilní aplikaci, která slouží pro navázání spojení s měřícím zařízením, k vyhodnocení výsledků, a k měření doby letu. Na základě výsledků z testování lze konstatovat, že cíl práce byl splněn.

K vytvoření výsledného řešení jsem nastudoval technologie pro měření zrychlení a frekvence otáčení. Dále bylo nutné nastudovat způsoby komunikace v počítačových periferiích a techniky vytváření aplikací pro mobilní zařízení. Na základě získaných poznatků, jsem navrhl výsledný systém.

Prostudoval jsem dostupnou literaturu z oboru provozu ultralehkých letadel, která obsahovala informace o možném využití vestavěných systémů letadla, například některých servisních konektorů.

Na základě nastudovaných informací jsem navrhl způsob tarifkace letadla podle doby letu. Doba letu je určena z hodnot zrychlení a frekvence otáčení motoru. Tyto údaje rozhodují o vzletu a přistání letadla, a díky tomu je možné interval mezi těmito stavy změřit.

Provozní parametry (zrychlení a frekvence motoru) jsou snímány elektronickým zařízením, které je na palubě letadla. Pro měření frekvence otáčení motoru slouží výstupní konektor od zapalovacích svíček motoru. Zrychlení je měřeno vybraným modulem s akcelerometrem, který je vestavěn do měřícího zařízení. Vestavěný systém řídí mikrokontrolér. Zařízení dále obsahuje Bluetooth modul, který měřené údaje bezdrátově odešle do mobilního telefonu. Vytvořená mobilní aplikace tato data přijme, vyhodnotí a změří dobu letu. Měřící zařízení musí brát do úvahy vlastnosti ultralehkých letadel, a jeho instalace musí být nenáročná.

Z výsledků testování systému plyne určitá nepřesnost měření frekvence motoru. Tuto odchylku lze redukovat průměrováním výsledné hodnoty z hodnot minulých, prodloužením intervalu pro snímání impulsů, nebo použitím výkonnějšího mobilního zařízení pro měření.

Rozšířením stávající práce může být přidání vytvořeného systému do komplexnějšího zařízení, které vyhodnocuje další provozní parametry letadla, jako trajektorii, výšku atd. Na základě těchto údajů lze vytvořit rozsáhlý letový záznam, který by se ukládal do vytvořené databáze. Díky použití akcelerometru lze dále vyhodnocovat vliv letu na konstrukci letadla.

Literatura

- [1] *Android activity life*. [Online; navštíveno 1.5.2017].
URL <https://developer.android.com/reference/android/app/Activity.html>
- [2] *Android platform architecture*. [Online; navštíveno 1.5.2017].
URL <https://developer.android.com/guide/platform/index.html>
- [3] *BTM-112 Datasheet*. [Online; navštíveno 20.4.2017].
URL https://www.sparkfun.com/datasheets/Wireless/Bluetooth/BTM112_wATcommands.pdf
- [4] *I2C Specification*. [Online; navštíveno 20.4.2017].
URL http://www.nxp.com/documents/user_manual/UM10204.pdf
- [5] *Material design guidelines*. [Online; navštíveno 1.5.2017].
URL <https://material.io/guidelines/>
- [6] *MMA8452Q Datasheet*. [Online; navštíveno 20.4.2017].
URL https://www.sparkfun.com/datasheets/Wireless/Bluetooth/BTM112_wATcommands.pdf
- [7] Bejček, L.; Čejka, M.; Rez, J.; aj.: *Měření v elektronice*. [Online; navštíveno 7.1.2017].
URL <http://docplayer.cz/16949561-Mereni-v-elektrotechnice.html>
- [8] Fraden, J.: *Handbook of modern sensors*. Springer, 2010, ISBN 978-1-4419-6466-3.
- [9] Haasz, V.; Sedláček, M.: *Elektrická měření - Přístroje a metody*. Vydavatelství ČVUT, 1998, ISBN 80-01-01717-6.
- [10] Hušek, M.: *Měření otáček*. [Online; navštíveno 7.1.2017].
URL <http://www.qtest.cz/mereni-otacek/mereni-otacek.htm#otackomery>
- [11] Martínek, R.: *Senzory v průmyslové praxi*. BEN, 2004, ISBN 978-80-7300-114-4.
- [12] Valášek, P.: *Monolitické mikroprocesory a mikropočítače*. SNTL, 1989, ISBN 80-03-00056-4.
- [13] Vlach, J.; Vlachová, V.: *Počítačová rozhraní*. BEN, 2000, ISBN 80-7300-010-5.
- [14] Vojáček, A.: *Principy akcelerometrů*. [Online; navštíveno 7.1.2017].
URL <http://automatizace.hw.cz/clanek/2007011401>
- [15] Šulc, J.; kol.: *Technologická a strojnická měření*. SNTL, 1982, ISBN 04-214-82.

Přílohy

Seznam příloh

A Obsah přiloženého CD

76

Příloha A

Obsah přiloženého CD

- Mobilní aplikace - zdrojové kódy a instalační balíček pro mobilní zařízení se systémem Android
- Mikrokontrolér - zdrojové kódy pro procesor ADuC7020
- Text práce - zdrojový text práce ve formátu .tex a .pdf
- Schémata - Schémata zapojení elektrického obvodu měřicího systému a desky plošných spojů