



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

LETECKÝ ÚSTAV

INSTITUTE OF AEROSPACE ENGINEERING

SYSTÉM PRO MĚŘENÍ ÚHLU NÁBĚHU

ANGLE OF ATTACK MEASUREMENT SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Denis Koribský

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Pavel Zikmund, Ph.D.

BRNO 2019

Zadání bakalářské práce

Ústav: Letecký ústav
Student: **Denis Koribský**
Studijní program: Strojírenství
Studijní obor: Základy strojního inženýrství
Vedoucí práce: **Ing. Pavel Zikmund, Ph.D.**
Akademický rok: 2018/19

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Systém pro měření úhlu náběhu

Stručná charakteristika problematiky úkolu:

Úhel náběhu je rozhodující veličina, jejíž znalost je klíčová při varování pilota před pádem. Pro měření tohoto úhlu lze použít různé senzory. Pozornost bude věnována systému s korouhvičkou, která pomocí magnetického enkodéru posílá údaj úhlu náběhu do řídicí jednotky. Úkol bude spočívat v dokončení a kalibraci korouhvičky.

Cíle bakalářské práce:

Dokončení rozpracovaného prototypu korouhvičky.
Softwarové zpracování dat (Arduino).
Kalibrace korouhvičky.

Seznam doporučené literatury:

GRACEY, W., Summary of methods of measuring angle of attack on aircraft. 1958.

MITCHELL, J. L., PECK, R. F. An NACA Vane-Type Angle-of-Attack Indicator for Use at Subsonic and Supersonic Speeds. 1955.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2018/19

V Brně, dne

L. S.

doc. Ing. Jaroslav Juračka, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Cílem této práce bylo dokončit systém pro měření úhlu náběhu a úhlu vybočení během letu. Systém je založen na dvou korouhvičkách, jejichž úhel natočení se snímá pomocí rotačních magnetických enkodérů. Bylo provedeno správné zapojení jednotlivých komponent k základní desce Arduino a byl přizpůsoben program v rozhraní Arduino IDE. Naměřené data úhlu natočení korouhviček byla následně ukládána na SD kartu. Po kompletním sestavení systému byla provedena kalibrace s využitím geometrických šablon. Výsledky kalibrace byly vhodně proloženy funkcí, sloužící k výpočtu skutečného úhlu náběhu.

Klíčová slova

Úhel náběhu, úhel vybočení, měření úhlu náběhu, Arduino, rotační magnetický enkodér

ABSTRACT

The aim of the bachelor thesis is to finish a system for measuring angle of attack and angle of sideslip during the flight. The system is based on two wind vanes that measure angle of attack by using a rotary magnetic encoder. The individual components were connected to Arduino and the program was adapted to these components by using software Arduino IDE. Measured data of the vane angle were after that saved on the SD card. After the system was assembled the vanes were calibrated by using the geometric template. Data of calibration were interpolated with trendline to get the function for calculation of real angle of attack.

Key words

Angle of attack, angle of sideslip, angle of attack measurement, Arduino, rotary magnetic encoder

BIBLIOGRAFICKÁ CITACE

KORIBSKÝ, Denis. *Systém pro měření úhlu náběhu* [online]. Brno, 2019 [cit. 2019-02-28]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/117431>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Letecký ústav. Vedoucí práce Pavel Zikmund.

PROHLÁŠENÍ

Prohlašuji, že předloženou bakalářskou práci na téma Systém pro měření úhlu náběhu jsem vypracoval samostatně s použitím uvedené literatury a podkladů, získaných na základě konzultací s vedoucím bakalářské práce

V Brně dne 24. května 2019

Denis Koribský

PODĚKOVÁNÍ

Tímto bych chtěl poděkovat panu Ing. Pavlu Zikmundovi, Ph.D. za jeho cenné rady, příjemné konzultování a poskytnuté připomínky při zpracování této bakalářské práce.

Obsah

1	ÚVOD	15
2	ARDUINO	16
2.1	Hardware	16
2.2	Software	17
2.3	Typy Arduina	18
2.4	Výhody a nevýhody Arduina	18
3	ÚHEL NÁBĚHU	19
3.1	Zavedení souřadnicových soustav	19
3.2	Definice úhlu náběhu a úhlu vybočení	20
3.3	Održení proudu při vysokých úhlech náběhu	20
3.4	Vliv úhlu náběhu na vztlak	21
4	POUŽÍVANÉ SENZORY PRO SNÍMÁNÍ ÚHLU NÁBĚHU	22
4.1	Křídélkové snímače	22
4.2	Tlakové snímače	23
5	SYSTÉM OCHRANY PROTI PÁDU	24
6	ZPŮSOBY SIGNALIZACE PÁDOVÉHO VAROVÁNÍ	24
6.1	Optická signalizace	25
6.2	Zvuková signalizace	25
6.3	Stick Shaker a Stick Pusher	25
7	NÁVRH SYSTÉMU PRO MĚŘENÍ ÚHLU NÁBĚHU	26
7.1	Volba komponent	26
7.1.1	Modul na SD kartu	27
7.1.2	Rotační magnetický enkodér	27
7.2	Schéma zapojení	28
7.2.1	Připojení obou snímačů	29
7.2.2	Připojení tlačítka a LED diody	30
7.2.3	Použití Data Logger Shieldu	31
7.1	Algoritmus řízení	31
7.2	Software snímačů	32
7.3	Vymezení vůle mezi magnetem a ložiskem	33
7.4	Trubice k umístění snímačů	33
7.5	Kalibrace snímačů	34
7.5.1	Geometrická kalibrace v rozmezí 360 stupňů	34
7.5.2	Geometrická kalibrace v rozmezí 50 stupňů	35
7.5.3	Proložení funkce	37
7.6	Další možný vývoj	38
8	ZÁVĚR	39
9	SEZNAM POUŽITÝCH ZDROJŮ	40
10	PŘÍLOHY	42

1 ÚVOD

Letecká doprava je technický obor, vyžadující neustálý výzkum a vývoj. Snaha modernizovat letectví jde neustále kupředu. Firmy pohybující se v oblasti leteckého průmyslu se v dnešní době zaměřují především na ekonomičtější létání a vysokou bezpečnost. Pro zajištění bezpečného pohybu letounu vzduchem, je nezbytné, aby byla posádka nepřetržitě a náležitě informována o aktuálním stavu letounu. Jako vstupy veškerých fyzikálních parametrů a dat o aktuálním stavu letounu se využívají letecké snímače a sondy. Jedním z těchto snímačů, je i snímač úhlu náběhu. Nejčastěji tento snímač využívají dopravní nebo vojenské letouny, u menších letadel je jeho výskyt výjimečný. Úkolem tohoto snímače je piloty informovat o aktuálním úhlu náběhu, který je rozhodující veličinou, při vzniku vztlakových sil. S blížícím se kritickým úhlem náběhu se může letoun dostat do velmi nebezpečných situací. Těm je nutno předcházet.

Tato práce se zabývá dokončením návrhu systému pro měření úhlu náběhu. Systém je sestaven ze dvou korouhviček, které slouží k měření úhlu náběhu a vybočení. Řídící jednotkou byla zvolena platforma Arduino, která je popsána v úvodní části práce. Následující část se zabývá veličinou úhlu náběhu a pojednává o způsobech jejího snímání.

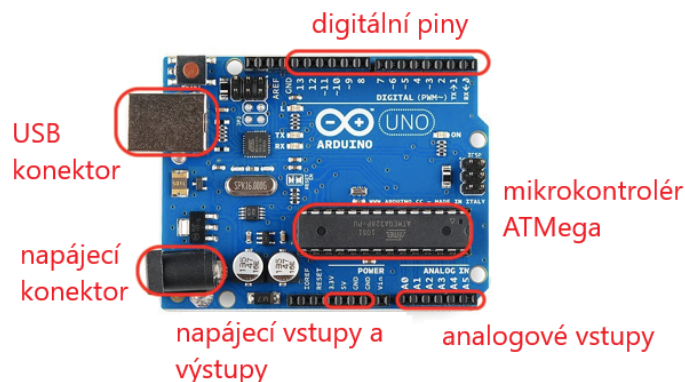
Hlavní pozornost v této práci je věnována konkrétním postupům při realizaci systému. Součástí práce je také schéma zapojení veškerých komponent k Arduino a naprogramovaný software, řídící celý systém. Na kompletně sestavený systém navazuje geometrická kalibrace snímačů, která sleduje závislost hodnoty natočení ze snímače na skutečném vychýlení korouhvičky. Daty získanými kalibrací byla proložena vhodná funkce, která má sloužit k dopočítávání hledané veličiny úhlu náběhu.

2 ARDUINO

Arduino je elektronická platforma, která funguje jako nástroj pro tvorbu interaktivních objektů nebo prostředí. Je založené na jednoduché počítačové desce a softwaru. Jedná se o open-source platformu, což znamená, že veškeré návrhy plošných spojů jsou pod otevřenými licencemi veřejně dostupné. [2] Vznik Arduina začal v roce 2005 v Itálii, kde jej vytvořili studenti jako levné a jednoduché zařízení pro vytváření technických prvků. Zařízení se mezi studenty velmi uchytilo a tvůrci jej nabídli celému světu. V dnešní době je Arduino využíváno mezi designery, inženýry či uživateli, kteří se se zapojováním elektrických obvodů setkávají poprvé. [1]

2.1 Hardware

Nejdůležitější součástí hardwaru Arduina je mikrokontrolér ATMEGA od firmy Atmel, který se vyrábí v několika variantách podle typu desky. Na desce se dále nachází digitální piny, analogové vstupy, USB konektor, napájecí konektor a napájecí vstupy a výstupy Arduina. Digitální piny slouží ke komunikaci mezi Arduinem a dalšími komponentami. Informace z nich mohou buďto vycházet nebo do nich vcházet. Při komunikaci využívají binární soustavu, nesou tedy informaci jedna nebo nula (LOW a HIGH). Analogové vstupy slouží k měření napětí ze senzorů založených na analogové komunikaci. Během komunikace nečtou jednu ze dvou hodnot, ale měří 1024 různých úrovní napětí. K propojení mezi počítačem a Arduinem slouží USB konektor. Pro přívod napětí u komponent, které připojujeme k Arduinu využíváme napájecí vstupy a výstupy. Nejčastěji se setkáme se stejnosměrným napětím 3,3V a 5V.[3]



Obrázek 1: Arduino UNO R3 [18]

2.2 Software

Arduino využívá speciální programovací jazyk, založený na jazyku Wiring a vývojové prostředí Arduino Software (IDE), které je založené na prostředí Processing. Arduino IDE je zdarma ke stažení na internetu a je dostupné pro operační systémy Windows, Linux a Mac OS X. Programovací jazyk může být použitím speciálních knihoven rozšířen o jazyk C++, nicméně samotný jazyk Arduina je velice podobný C++. [2][3]



Obrázek 2: Rozhraní programu Arduino IDE

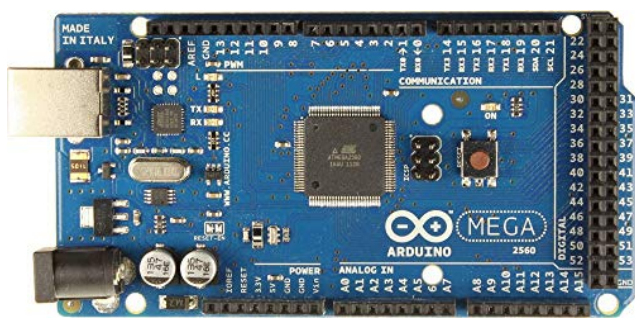
Po spuštění programu se otevře vývojové prostředí (viz obr. 2), ze kterého je Arduino ovládáno. Dříve než se uživatel pustí do programování, musí v záložce *Nástroje > Vývojová deska* zvolit typ Arduina, se kterým bude pracovat. V levém horním rohu [1] jsou ikony sloužící k ověření a nahrání kódu. Kliknutím na ikonu [2], lze otevřít okno *Seriálový monitor*. V tomto okně můžeme pomocí příkazu *Serial.print* vypisovat data. V oblasti [3] se nachází samotný kód, který lze rozdělit do několika oblastí. V první části jsou definovány proměnné, se kterými program pracuje. Pod nimi se nachází funkce *setup()*, ve které proběhne program pouze jednou. V další funkci *loop()* se nachází hlavní část programu, která se opakuje pořád dokola. Ve spodní oblasti [4], můžeme kontrolovat, jak moc je zatížená paměť procesoru Arduina.

Za zmínku také stojí využívání knihoven v aplikaci. Knihovny totiž tvoří nezbytnou část každé komponenty, kterou chceme k Arduino připojit. Některé knihovny jsou již integrované v rozhraní IDE. K některým komponentům je ale nutné knihovny dodatečně stáhnout z internetu a nainstalovat je. Po úspěšné instalaci, musí být v kódu daná knihovna vyvolána, a to pomocí příkazu `#include <libovolná_knihovna.h>`.

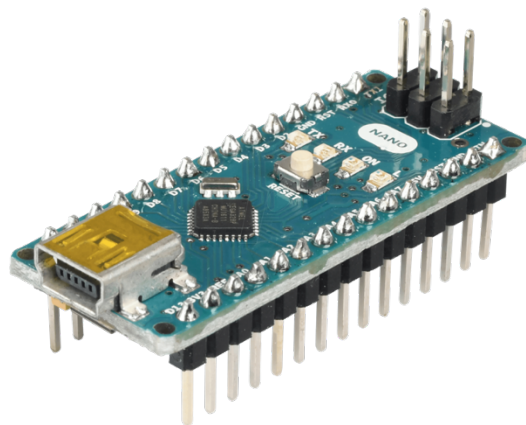
2.3 Typy Arduina

Na trhu se můžeme setkat s různými typy desek. Nejčastěji se od sebe liší velikostí desky, čipem nebo počtem vstupů a výstupů. Většina desek bývá vybavena svým specifickým portem, což je například USB, HDMI, Ethernet či Bluetooth port. Díky otevřenosti Arduina, můžeme na trhu narazit také na klony či alternativy této platformy, které bývají cenově přístupnější ve srovnání s originálními verzemi. Alternativy mají své vlastní vývojové prostředí a s Arduinem nejsou kompatibilní. Jednou z alternativ je například Teensy USB Development Board.

Na obr. 1, kde je popsána struktura desky, můžeme vidět jednu z nejprodávanějších desek Arduino UNO R3. Mezi další desky, které se často objevují na trhu patří Arduino MEGA, které můžeme vidět na obr. 4 a Arduino Nano na obr. 3. [4][2]



Obrázek 4: Arduino MEGA



Obrázek 3: Arduino Nano

2.4 Výhody a nevýhody Arduina

Při tvorbě interaktivního prostředí má uživatel na výběr ze spousty platform, a to nejen od značky Arduino. Většinou ale bývají tyto platformy složitější, co se týče programování a samotné komunikace s mikrokontrolérem. Hlavní výhodou Arduina je tedy jeho jednoduchost, při které je uživateli nabídnuto co nejpřehlednější prostředí. Toto využijí jak úplní začátečníci, tak pokročilejší kutilové. Snadná je také manipulace s komponenty, které chceme propojovat s Arduinem. Pro většinu dostupných komponent, bývá na internetovém obchodě přiložen návod (častěji nazýváno datasheet). Tento datasheet obsahuje schéma zapojení, potřebné knihovny a ukázkou programu. Další výhodou je rozhodně nízká cena. Ta se může lišit u originálních desek či klonů. Výhodou je také velká uživatelská komunita, díky níž je sdílená spousta již zhotovených projektů, návodů či rad. [3]

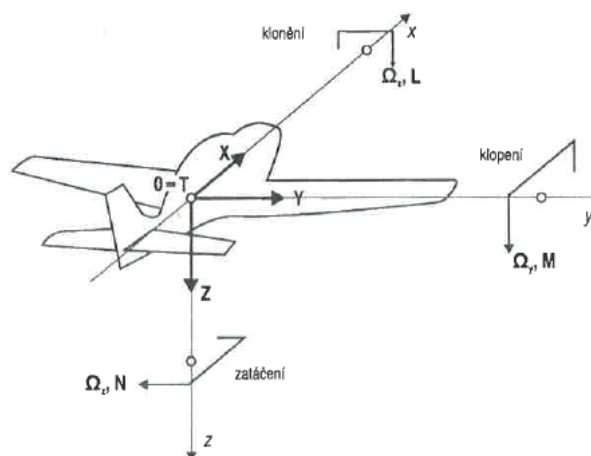
Jednou z nevýhod je například využití Arduina pro reálné komerční účely, které často potřebují certifikaci ke splnění důležitých požadavků. Arduino má krátký životní cyklus, což je způsobeno neustálým příchodem nových modelů desek. Certifikovat tuto desku se tedy nevyplatí. Arduino dále nenabízí bezpečnostní mechanismy či použití šifrovacích protokolů, které jsou v komerční praxi nezbytné. [1]

3 ÚHEL NÁBĚHU

3.1 Zavedení souřadnicových soustav

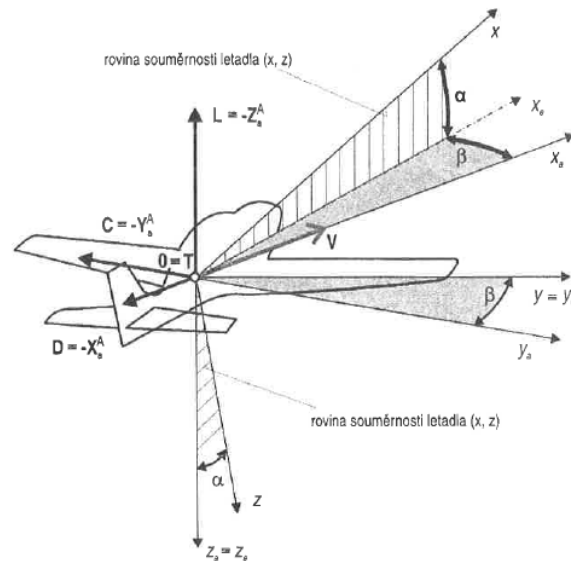
Před definováním veličiny úhlu náběhu je nutné popsat v jakých souřadnicových soustavách se letoun může pohybovat. Tyto souřadnicové soustavy jsou normalizované a jejich volba závisí na konkrétní úloze, která má být řešena. V mechanice letu existuje mnoho těchto soustav. V rámci této práce jsou podstatné pouze tyto dvě – letadlová souřadnicová soustava a aerodynamická souřadnicová soustava.

První je tzv. **letadlová souřadnicová soustava**, která je zobrazena na obr. 5. Tato soustava je pevně spojena s letounem. Její počátek leží v těžišti letounu a osa x , neboli podélná osa, je orientována ve směru osy trupu v kladném smyslu k přídí letadla. Osa z , nazývaná kolmá osa, vede směrem dolů a je kolmá na osu x . Osa x a z tvoří rovinu souměrnosti letadla. Na tuto rovinu je kolmá osa y s kladným smyslem vpravo při pohledu ve směru letu. Osa y se nazývá bočná osa. Rotaci letadla kolem těžiště rozdělujeme na tři rotační pohyby. Rotace kolem osy x se nazývá klonění, kolem osy z zatáčení a kolem osy y klopení. [6][5]



Obrázek 5: Letadlová souřadnicová soustava s vyznačením rotací kolem os

Druhá souřadnicová soustava, zobrazena na obr. 6, je **aerodynamická souřadnicová soustava**. Tato souřadnicová soustava slouží k popisu aerodynamických sil a momentů. Její počátek leží v těžišti letounu a osa x_a je orientována v kladném smyslu ve směru vektoru okamžité rychlosti letounu V . Osa z_a je kolmá na osu x_a a obě tyto osy leží v rovině souměrnosti. Osa y_a je kolmá na osy x_a, z_a a směřuje doprava. Na těchto třech osách leží vektory aerodynamických sil, které jsou kladně orientovány v opačném smyslu vůči osám aerodynamické souřadnicové soustavy. [6][5]



Obrázek 6: Aerodynamická souřadnicová soustava s vyznačenými aerodynamickými silami a úhly ofukování

3.2 Definice úhlu náběhu a úhlu vybočení

Úhel náběhu α a úhel vybočení β , patří do tzv. úhlů ofukování letounu (někdy nazývány jako úhly obtékání). Na těchto úhlech závisí velikosti aerodynamických sil a momentů. Úhly ofukování α a β jsou vyznačeny na obr. 6. [6]

Úhel náběhu (angle of attack – AOA) α pro celý letoun je definován jako úhel, který svírá průmět vektoru rychlosti \vec{V} do roviny souměrnosti xz s podélnou osou x . Jinak řečeno, je to vychýlení mezi aerodynamickou a letadlovou souřadnicovou soustavou v rovině souměrnosti xz . Lze jej také definovat pro samotný profil křídla, jako úhel mezi tětvou profilu křídla a směrem nabíhajícího proudu. Tětvou profilu se myslí spojnice náběžné hrany a odtokové hrany. Úhel náběhu má zásadní vliv na vztlakovou sílu působící na letoun a lze díky němu signalizovat nebezpečí ztráty vztlaku. [7][5]

Úhel vybočení (angle of sideslip – AOS) β je úhel, který svírá podélná osa x a průmětem vektoru rychlosti \vec{V} do roviny xy . Tento úhel nelze stanovit na základě geometrie profilu křídla a nemá žádný vliv na obtékání profilu. Avšak má důležitou roli při vzniku vztlaku a odporu. Tohoto úhlu lze docílit vychýlením směrového kormidla případně se může objevit jako druhotný pohyb následkem klonění. [6][7]

3.3 Odtržení proudu při vysokých úhlech náběhu

Při vysokých úhlech náběhu dojde k takzvanému odtržení mezní vrstvy. Mezní vrstva vzniká na povrchu všech těles, které jsou obtékány proudem tekutiny. Na povrchu obtékaného tělesa, má tekutina vlivem vnitřních třecích sil nulovou rychlost.

Při zvyšování úhlu náběhu narůstá i podtlaková špička na horní straně profilu. Od náběžného bodu až po tuto špičku rychlost proudu vzduchu roste, někdy i několika násobně vůči rychlosti nerušeného proudu. Za podtlakovou špičkou směrem k odtokové hraně rychlost proudu naopak klesá a podle Bernoulliho rovnice narůstá tlak. [6]

Održení je stav, kdy už mezní vrstva není schopná proudit do příliš velkého protitlaku. Od povrchu tělesa se vrstva oddělí, vytvoří si svou vlastní nulovou proudnici, pod kterou začne vznikat zpětné proudění. K odtržení začne docházet na odtokové hraně a při zvyšování úhlu náběhu se začne posouvat směrem k náběžné hraně. Tento jev způsobí zmenšení součinitele vztlaku a nárůst součinitele odporu. [6][8]

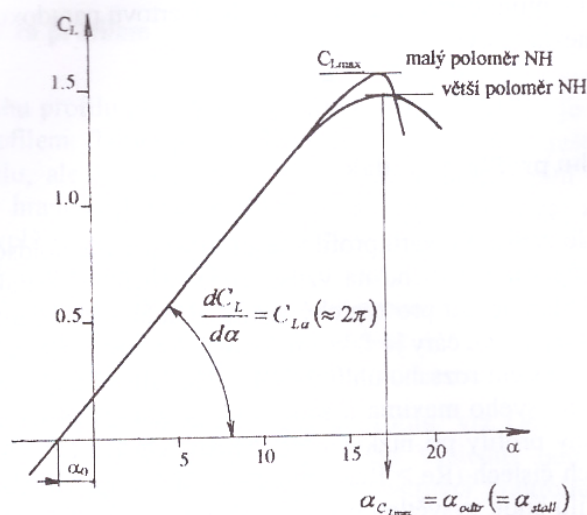
3.4 Vliv úhlu náběhu na vztlak

Úhel náběhu má spolu s tvarem profilu, Reynoldsovým číslem a Machovým číslem vliv na součinitel vztlaku C_L . S narůstajícím úhlem náběhu hodnota C_L lineárně roste. Dosažením určitého úhlu náběhu se začne nárůst C_L zmenšovat až do dosažení svého maxima C_{Lmax} , odkud začnou hodnoty součinitele vztlaku klesat. Úhel náběhu pro C_{Lmax} označujeme α_{odtr} , neboli kritický úhel náběhu, při kterém dochází k tzv. přetažení letounu (neboli *Stall*). [6][10]

Při přetažení letounu dojde k náhlé ztrátě vztlaku v důsledku překročení kritického úhlu náběhu. Letoun začne rychle ztrácet výšku a jeho říditelnost je v tento moment velmi náročná. [9]

Rychlost, kterou se při přetažení pohybuje se nazývá pádová rychlost. Jakou rychlostí dojde ke zmenšení vztlaku lze ovlivnit tvarem profilu křídla či geometrií křídla. Prudký průběh této změny, který je způsoben malým poloměrem náběžné hrany, je v letectví nežádoucí. [6]

Závislost součinitele vztlaku C_L na úhlu náběhu α udává tzv. vztlaková čára profilu, kterou lze vidět na obr. 7. Důležité je zmínit že nulový součinitel vztlaku je při záporném úhlu náběhu. [6]



Obrázek 7: Vztlaková čára profilu [6]

V průběhu letu je stav přetažení nežádoucí a snahou je mu zabránit. Vznik vztlaku je závislý na úhlu náběhu, tvaru profilu a Machově a Reynoldsově čísle. Při vysokých rychlostech stačí letounu menší úhel náběhu, aby dosáhl dané hodnoty vztlaku. Při nízkých rychlostech dosáhne konkrétního vztlaku při vyšším úhlu náběhu, přičemž je nutné dbát na nepřesazení kritického úhlu náběhu. [10]

4 POUŽÍVANÉ SENZORY PRO SNÍMÁNÍ ÚHLU NÁBĚHU

Pro snímání základních aerometrických veličin slouží senzory umístěné po stranách trupu letounu, případně na křídlech. Mezi aerometrické veličiny patří statický a celkový tlak, teplota okolí a úhel náběhu případně vybočení. Z aerometrických veličin se dále vyhodnocují důležité parametry. V rámci této práce bude věnována pozornost pouze snímání úhlů náběhu a vybočení.

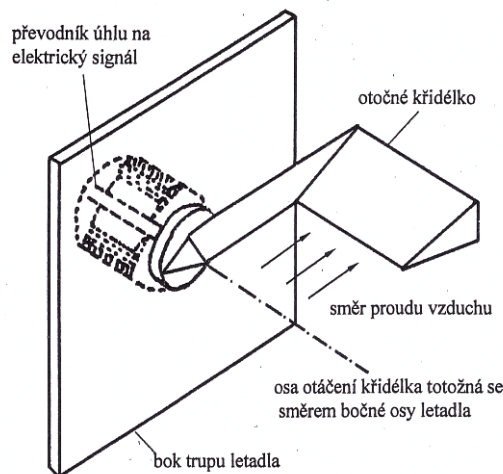
Se snímači úhlu náběhu se nejčastěji setkáme u dopravních či vojenských letadel. Snímače jsou obvykle umístěny na boku trupu v přední části letounu. U menších letadel mohou být upevněny přímo na křídlech. Hlavní snahou je, aby byly co nejméně zasaženy proudnicemi vzduchu od cizích těles.

V současné době existuje více typů snímačů, přičemž všechny fungují na stejném principu – detekují vychýlení podélné osy x vůči směru nabíhajícího proudu. [7]

4.1 Křídélkové snímače

Tento způsob snímání je založen na jednoduchém křídélku, které je volně otočné kolem osy rovnoběžné s podélnou osou letounu. Tento typ snímače je zobrazen na obr. 9.

Princip je založen na vychýlení křídélka ve směru proudícího vzduchu. Tvar křídélka je přizpůsoben tak, aby jej proud vzduchu během letu ustálil na dané poloze. Vychýlení podélné osy letounu vůči směru proudu vzduchu je zaznamenáno elektrickým převodníkem, který převádí rotační pohyb na elektrické signály. [7][11]



Obrázek 9: Křídélkový snímač úhlu náběhu

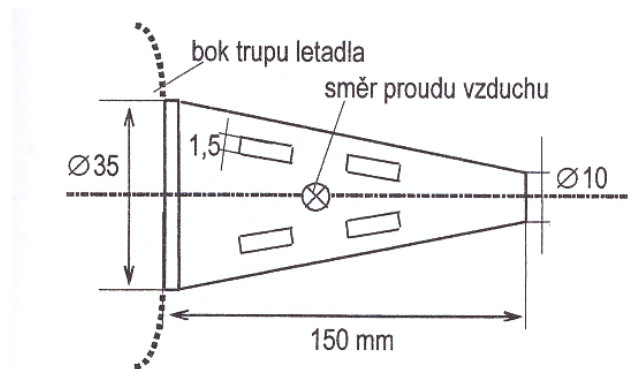


Obrázek 8: Senzor úhlu náběhu na letounu Boeing 737 MAX

4.2 Tlakové snímače

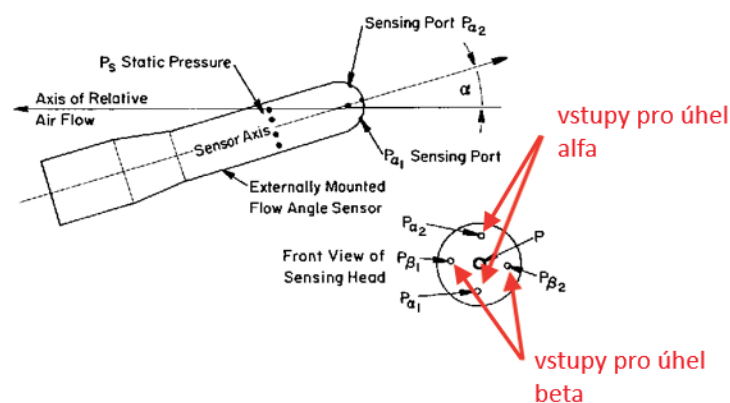
Princip těchto snímačů spočívá ve změně tlaku vzduchu vyvolané nenulovým úhlem náběhu.

První typ je tzv. **null-seeking pressure sensor**. Snímač je kuželovitěho tvaru a je volně otočný kolem osy rovnoběžné s bočnou osou. Vzduch proudí do kužele skrz otvory, umístěné na horní a spodní straně kužele. Při zvýšení úhlu náběhu dojde ke změně tlaků ve spodních a horních otvorech. Tato změna tlaků je vyhodnocena zpětnovazebním systémem, který kužel natočí tak, aby byla diference tlaků nulová. Zjišťovaný úhel náběhu je potom úhel, o který se musí kužel natočit. Schéma snímače můžeme vidět na obr. 10. [7]



Obrázek 10: Null-seeking pressure sensor

Další typ je tzv. **differential pressure tube sensor**. Otvory u tohoto snímače jsou umístěny na přední zakulacené straně trubice, jejíž osa je rovnoběžná s podélnou osou letounu. Dva otvory, které jsou umístěny vertikálně slouží k měření úhlu náběhu α . Další dva horizontálně umístěné otvory měří úhel vybočení β . Prostředním otvorem vstupuje celkový tlak. Princip fungování toho typu je zobrazen na obr. 11. [12]

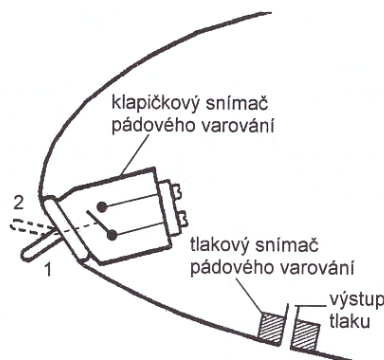


Obrázek 11: Differential pressure sensor

5 SYSTÉM OCHRANY PROTI PÁDU

Systém ochrany proti pádu se používá u letadel, které nejsou vybaveny systémy pro měření úhlu náběhu. Hlavní funkcí systému je varovat pilota při přiblížení se kritickému úhlu náběhu, při kterém může letoun přejít do nekontrolovatelného pádu. Nejčastěji se s tímto systémem potkáme u menších letadel, kde je součástí výstražného systému.

Snímač tohoto systému funguje na principu malé klapičky, která je umístěná těsně pod náběžnou hranou. Při malém úhlu náběhu proud vzduchu tlačí klapičku dolů. Při dosažení kritického úhlu náběhu proud vzduchu zatlačí klapičku do vrchní polohy. Ta následně sepně elektrický obvod, který varuje pilota. [7]



Obrázek 12: Klapičkový a tlakový snímač pádového varování [7]

Vedle klapičkového snímače se používá také tlakový snímač, který je také umístěn pod náběžnou hranou. Jeho úkolem je měřit náporový tlak vzduchu a porovnat ho se statickým tlakem. Dosažením kritického úhlu náběhu vzroste náporový tlak na dovolenou hodnotu, při které dojde k signalizaci pádového varování. Oba typy snímačů jsou zobrazeny na obr. 12. [7][9]

6 ZPŮSOBY SIGNALIZACE PÁDOVÉHO VAROVÁNÍ

Stavu, kdy letoun překročí kritický úhel náběhu a začne ztrácet vztlak, je nutné zabránit. Proto se u letounů používají varovné systémy, které mají posádku o této hrozbě informovat.

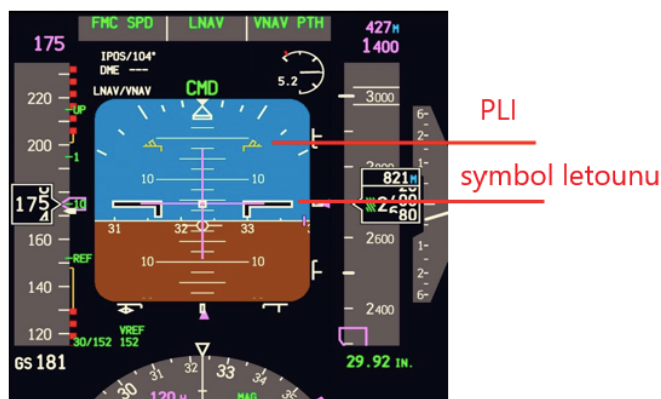
Pokud dojde k signalizaci varování o blížícím se kritickém úhlu náběhu, musí se piloti řídit standartním postupem. Napravením tohoto stavu lze dosáhnout zmenšením úhlu náběhu, tedy poklesem přídě letounu a zvýšením tahu motorů, čímž začne letoun zase zrychlovat.

V současné době se používá několik způsobů signalizace. U různých typů letounů se používají odlišné způsoby, jakými lze piloty signalizovat. U dopravních letadel se jedná o kombinaci více těchto způsobů signalizace naráz. V následující části budou popsány tyto základní typy signalizace. [13]

6.1 Optická signalizace

Optický způsob signalizace je pro pilota realizován vizuálně na přístrojích. U dopravních letadel bývá nejčastěji součástí primárního letového displeje. Například u letounu Boeing se setkáme se systémem PLI (neboli *Pitch Limit Indicator*), jehož vizualizaci lze vidět na obr. 13. Dá se říct, že když se na umělém horizontu symbol letadla překryje se značkou PLI, dojde k přetažení letounu. [14]

U některých displejů se můžeme také setkat se systémem, který varuje piloty v situaci, jestliže se hodnoty úhlu náběhu ze dvou snímačů neshodují déle jak 10 sekund.



Obrázek 13: Primární letový displej se systémem PLI

6.2 Zvuková signalizace

Tento varovný systém signalizuje pilota před blížícím se přetažením pomocí zvukové výstrahy. Většinou se může jednat o jednoduchý tón, který není lidskému uchu příjemný. U dopravních letadel se v kokpitu rozezná hlasitý alarm střídající se s upozorněním „STALL“. Můžeme se s tímto systémem potkat i u menších letadel, které jsou vybaveny systémem ochrany proti pádu (viz. kapitola 5). Jednoduše dojde k sepnutí elektrického obvodu, který spustí zvukovou signalizaci.

6.3 Stick Shaker a Stick Pusher

Princip tohoto systému funguje na varování, které pilota informuje přes knipl. Nejčastěji se s tímto systémem potkáme u letadel se systémem řízení Fly-By-Wire, kde je přímé mechanické řízení nahrazeno řízením pomocí elektronické jednotky.

Systém **Stick Shaker** je založen na kmitajícím setrvačnicku, který rozvibruje knipl při blížícím se přetažení letounu. U letounů s přímým mechanickým řízením zpravidla vznikají vibrace na kniplu díky turbulenci, která při odtržení na křídle zasáhne vodorovné ocasní plochy a roztřepá výškové kormidlo. U letounů, kde varování skrze řídicí páku chybí, je nutné vibrace vyvolat uměle pomocí elektromotoru, který rozkmitá setrvačník. [9]

Systém **Stick Pusher** už pouze nevaruje, ale přímo zasahuje do řízení. Při blížícím se přetažení, potlačí dva servomotory připojené k řízení, knipl dopředu. Tím vychýlí výškové kormidlo tak, aby se úhel náběhu snížil. Tento systém může být náchylný na chybná data, získané ze snímačů úhlu náběhu. Při jakékoli známce této neshody, musí být systém vypnut. [9]

Podobný systémem, který využívá automatického zásahu do řízení, se používá i u momentálně uzemněného letadla Boeing 737 MAX 8. Ten používá tzv. **MCAS** (neboli *Maneuvering Characteristics Augmentation System*), který před blížícím se přetažením automaticky sníží úhel náběhu vychýlením stabilizátoru. Problém zde nastal v případě chybných údajů získaných ze snímačů úhlu náběhu. V takovém případě je nutné systém MCAS vypnout a ovládat stabilizátor manuálně.

7 NÁVRH SYSTÉMU PRO MĚŘENÍ ÚHLU NÁBĚHU

Cílem tohoto návrhu bylo dokončit již rozpracovaný prototyp snímačů. Na počátku realizace byly snímače již navrhnuty a veškeré další komponenty potřebné k realizaci byly zajištěny. Úkolem bylo tyto komponenty spojit dohromady s Arduinem tak, aby byl celý systém funkční a přizpůsobit tomu již rozpracované softwarové prostředí.

Pro softwarové zpracování dat byla zvolena elektronická platforma Arduino. Hlavní výhodou této platformy je její jednoduchost a poměrně snadná dostupnost komponent, které jsou s Arduinem kompatibilní. Arduino se sice dobře vypořádá se sbíráním dat ze snímačů, avšak pro náročnější výpočty není úplně vhodný, a to hlavně z důvodu jeho omezené výkonnosti. Jeho nepřesnost výpočtů se projevuje například u převádění veličin, které jsou přepočítávány přes knihovny programu Arduino IDE.

Před zahájením realizace bylo nutné se seznámit s principem fungování Arduina. Toho bylo dosaženo cvičením základních úkonů s Arduinem. Snahou bylo se naučit správně zapojovat jednotlivé komponenty do obvodu a umět zacházet s programem.

Jedním z prvních těchto cvičení, bylo rozblikání několika LED diod. O něco složitějším úkonem bylo následně pomocí senzorů měřit teplotu a vlhkost okolí a ukládat naměřená data na SD kartu. Po seznámení se se základy fungování Arduina, bylo možné se přesunout k samotnému zadání.

Při realizaci zadaného systému bylo nutné se zaměřit na hardwarové a softwarové úpravy. Dalším úkolem bylo snímače vhodně přimontovat k trubici. Po vymezení polohy snímačů na trubici, bylo možné začít s jejich kalibrací. V následujících podkapitolách budou popsány jednotlivé kroky při realizaci systému.

7.1 Volba komponent

Pro systém bylo nutné použít několik komponent, které plní různé funkce. Tyto komponenty jsou připojeny k Arduinu a jsou řízeny pomocí softwaru. Mezi komponenty, které plní funkci pro zpracování dat, patří rotační magnetický enkodér a modul na SD kartu. Ke spouštění systému slouží tlačítko. K signalizaci o stavu měření slouží LED dioda. Základní princip fungování těchto součástí je popsán v následujících podkapitolách.

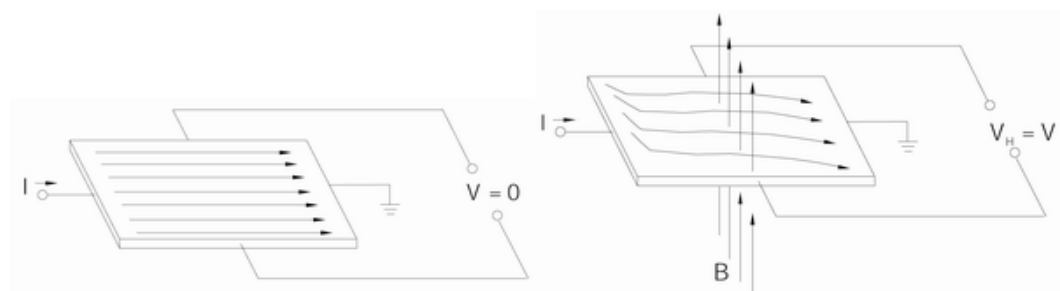
7.1.1 Modul na SD kartu

Modul s SD kartou slouží k ukládání naměřených dat. Při komunikaci s Arduinem využívá tzv. SPI sběrnice (neboli *Serial Peripheral Interface*). Ta je na modulu představena piny MOSI, MISO, SCK a CS. Každý z těchto pinů zaujímá na Arduinu svůj konkrétní pin, který nese číselné označení. Zapojení modulu k Arduinu je následující (první člen vyjadřuje výstup z modulu, druhý člen vstup do Arduina): MOSI–pin11, MISO–pin12, SCK–pin13 a CS–pin10. Tento modul byl následně nahrazen Datta Logger Shieldem, o kterém bude zmínka v kapitole 7.2.3.

7.1.2 Rotační magnetický enkodér

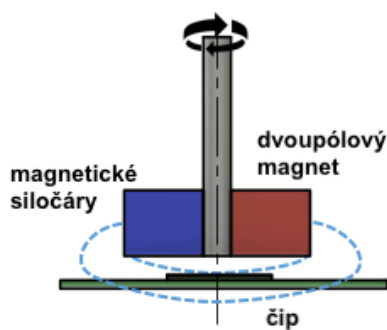
Rotační magnetický enkodér je elektronická součástka, která převádí rotační pohyb na napěťové signály.

Princip tohoto enkodéru je založen na Hallově sondě, která z fyzikálního hlediska využívá *Hallova jevu*. K tomuto jevu dochází, jestliže se tzv. *Hallův element* (vodič), kterým prochází proud elektronů, nachází v magnetickém poli. V takovém okamžiku, se elektrony začnou procházejícím elementem vychylovat do stran (způsobeno tzv. *Lorenzovou silou*). Rozložení nábojů se tak po stranách změní. Pokud by na stranách elementu byly připojeny svorky, generovalo by se na nich *Hallovo napětí*. [16] Tento jev je naznačen na obr. 14.



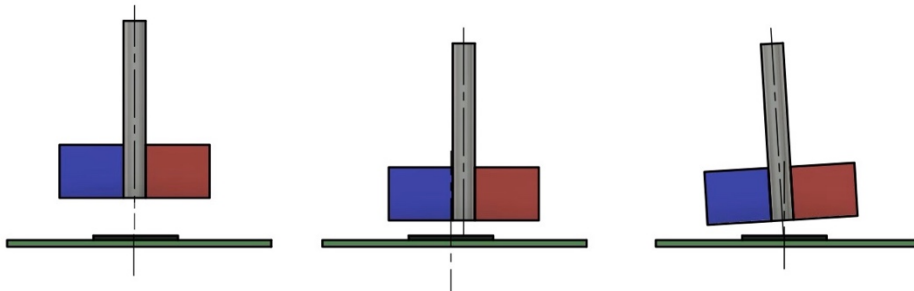
Obrázek 14: Hallův jev– levá strana bez působení magnetického pole, pravá strana s působením magnetického pole

Pro navrhovaný snímač byl použit enkodér AS5048B od výrobce AMS, který využívá sběrnice I^2C . Rozlišení tohoto enkodéru je 14 bitů. Jedné otáčky odpovídá tedy 16385 (vyjádřeno jako 2^{14}) poloh. Jeho provozní napětí se pohybuje v intervalu 4,5 V- 5,5 V. Provozní napájecí proud činí 15 mA. Měří s přesností na 0,0218° nicméně ve skutečnosti je tato hodnota o něco vyšší, a to z důvodu nepřesnosti umístění magnetu nad čipem, případně okolním šumem. Enkodér je u snímače úhlu náběhu představen čipem, nad kterým je umístěn rotující dvoupólový magnet, vychylující magnetické pole (viz obr. 15). Veškeré parametry enkodéru jsou převzaty z katalogu výrobce [17].



Obrázek 15: Schéma dvoupólového magnetického enkodéru

Nesprávné umístění magnetu nad čipem může mít zásadní vliv na přesnost snímání dat. Magnet musí být umístěn ve vhodné vzdálenosti od čipu. Tuto vzdálenost nejčastěji udává samotný výrobce enkodéru. Důležitá je také souosost mezi osou magnetu a osou čipu. K nesprávnému umístění může také dojít vychýlením osy magnetu vůči ose čipu. Způsoby nesprávného umístění magnetu nad čipem lze vidět na obr. 16.



Obrázek 16: Nesprávné umístění magnetu nad čipem

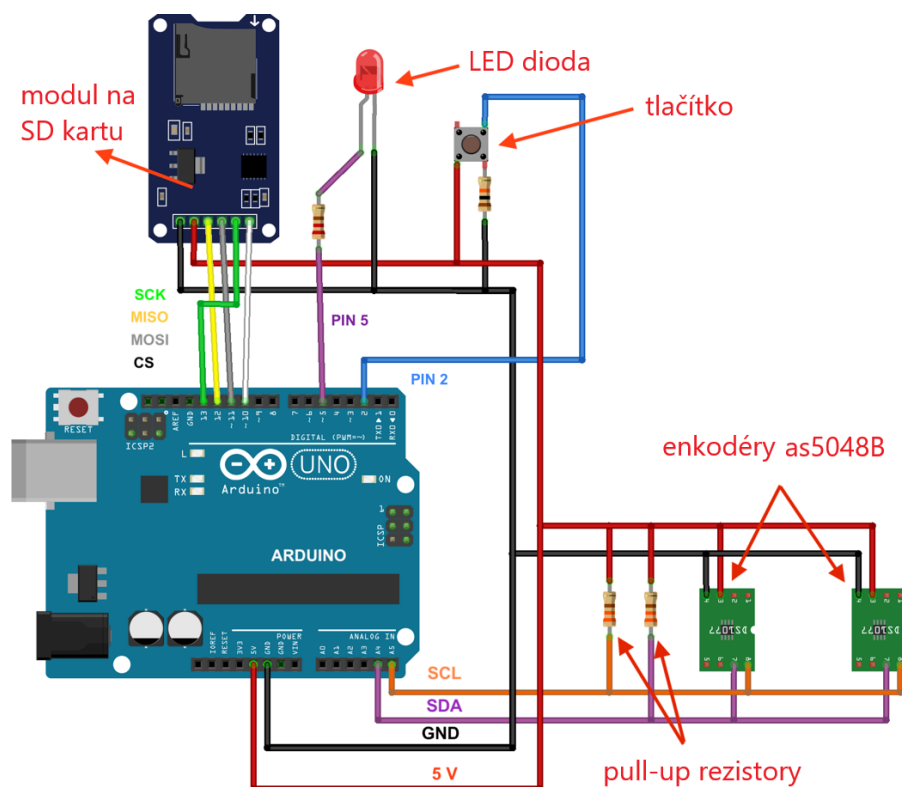
Komunikace přes I^2C sběrnici využívá obousměrný přenos po dvou vodičích – dat SDA (Serial Data) a hodin SCL (Serial Clock). Jedno ze zařízení je v obvodu nastaveno jako *master* (většinou je to mikrokontrolér) a ostatní zařízení jsou *slave* (snímače). Každý *slave* je představen svojí sedmibitovou adresou, díky které jej *master* rozezná a zahájí s ním komunikaci.

7.2 Schéma zapojení

Elektrický obvod je sestaven z několika dílčích částí. Jeho hlavní řídicí jednotkou je Arduino. K Arduinu jsou připojeny dva magnetické enkodéry, modul na SD kartu, tlačítko a LED dioda. Schéma zapojení všech součástí lze vidět na obr. 17. Toto schéma bylo vytvořeno pomocí aplikace Fritzing.

Prvotní verze systému měla určité nedostatky. Jednou z nich byla například funkčnost pouze jednoho snímače, nikoliv dvou. Spouštění snímačů začalo po připojení Arduina k počítači a ukončení nastalo po vypojení z počítače. Nebylo tedy možné systém ovládat na dálku podle potřeb.

Bylo tedy nutné provést několik nezbytných změn, co se týče zapojení komponent v elektrickém obvodu, tak, aby systém fungoval podle daných představ. V následující části budou popsány veškeré kroky, které vedly k hardwarovým úpravám systému.



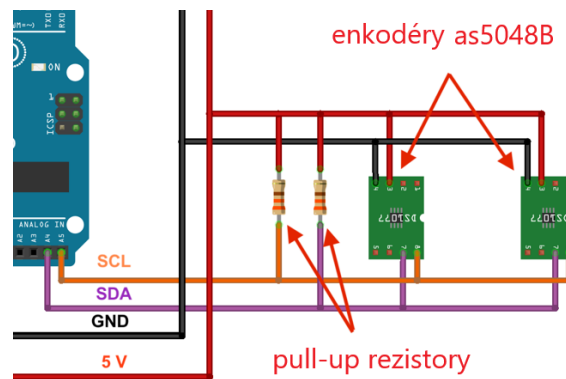
Obrázek 17: Schéma zapojení veškerých komponent k Arduino

7.2.1 Připojení obou snímačů

Prvním úkolem bylo do obvodu přidat oba snímače. Hlavní komponentou těchto snímačů je magnetický rotační enkodér, který převádí rotační pohyb na napěťové signály. S Arduinem tedy komunikuje přes analogové vstupy.

K připojení druhého snímače bylo nutné oba enkodéry mezi sebou propojit sériově. Jejich zapojení s Arduinem bylo následující (první člen vyjadřuje výstup z enkodéru, druhý člen vstup do Arduino): SDA–A4, SCL–A5, 5V–5V, GND–GND. Detailní schéma zapojení lze také vidět na obrázku č. 18.

Jak již bylo zmíněno u sběrnice I^2C mezi sebou komunikuje *master* (v tomto případě Arduino) a *slave* (magnetické enkodéry). Pro správný chod sběrnice bylo nutné připojit dva pull-up rezistory mezi vodiče SDA, SCL a napětí 5 V. Je tomu tak z důvodu udržování kladného napětí na těchto vodičích. V tomto případě byla zvolena hodnota rezistoru 330 Ω . Hodnota rezistorů byla volena na základě plynulosti toku dat ze snímačů, ale také na délce použitých kabelů. Čím delší kabel byl použit, tím byla zvolena menší hodnota rezistoru. Lze narazit na doporučení, že se v takových případech používají rezistory v rozmezí hodnot 1,2 k Ω až 4,7 k Ω . [15]



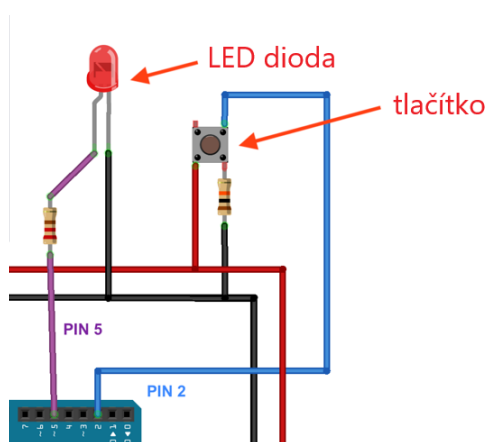
Obrázek 18: Schéma zapojení enkodérů k Arduino

7.2.2 Připojení tlačítka a LED diody

Zavedení tlačítka a LED diody do obvodu sloužilo k usnadnění řízení systému. Tlačítkem bylo možné systém libovolně zapínat či vypínat. Po prvním stisknutí (zapnutí) dostal systém příkaz k vytvoření nového souboru na SD kartě a následně spustil měření obou snímačů. Data ze snímačů se začaly ukládat na SD kartu. Po dalším stisknutí (vypnutí) došlo k uzavření souboru a k ukončení měření snímačů.

Tlačítko bylo na Arduinu připojeno k napětí 5 V, zemi GND a pinu 5. Mezi pin 5 a zem GND bylo nutné přidat pull-down rezistor. Ten při nestisknutém tlačítku, hraje důležitou roli, k dosažení logické 0, tedy nulového napětí. Pokud by pull-down rezistor nebyl použit, tak by i při nestisknutém stavu, mohl tlačítkem protékat nenulový proud. To by vedlo k nahodilému střídání logické 1 a logické 0. Hodnota rezistoru byla zvolena 10 k Ω .

LED dioda sloužila k signalizaci, zda měření probíhá či ne. Zapojená byla k zemi GND a přes rezistor k pinu 2. Rezistor zde slouží hlavně ke snížení proudu protékajícím přes LED diodu. Použití diody pomohlo odstranit problém s chybně ukončeným zápisem, kdy bylo Arduino vypnuto v průběhu zápisu dat na SD kartu.

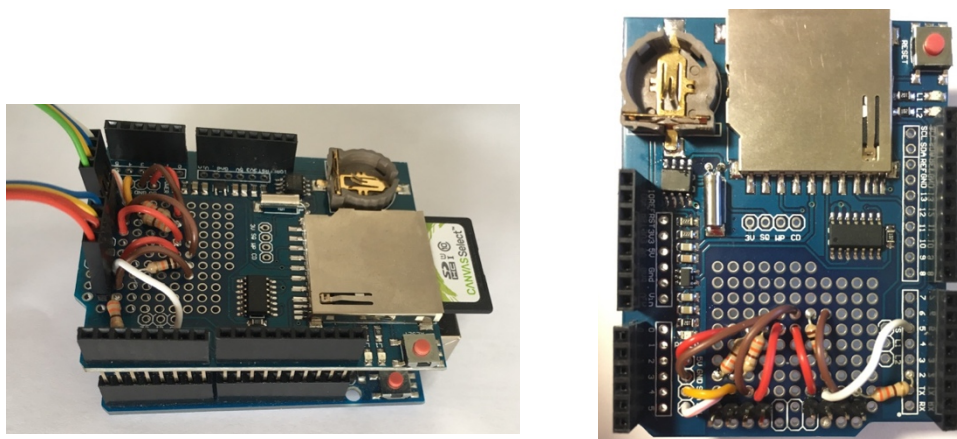


Obrázek 19: Schéma zapojení tlačítka a signalizační LED diody

7.2.3 Použití Data Logger Shieldu

Při sestavování obvodu byly veškeré vodiče vedeny k Arduino přes nepájivé pole neboli tzv. *Breadboard*. Tento způsob vedení nebyl úplně ideální, a to hlavně z důvodu častého vypořádání vodičů z pole při manipulaci s Arduinem.

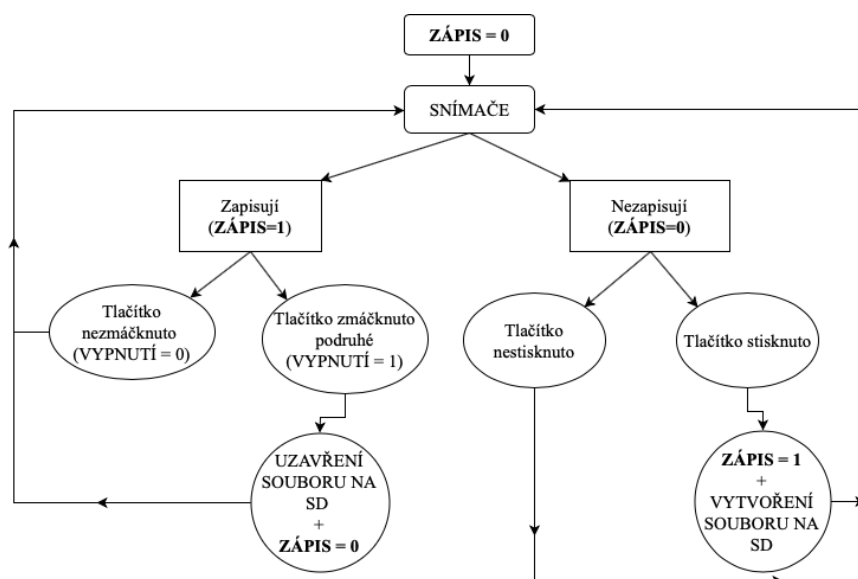
Proto byl použit přídatný modul Data Logger Shield, ve kterém je zabudován modul na SD kartu a obsahuje také pole volných otvorů, ke kterým bylo možné vodiče připojovat. Data Logger Shieldem byl tedy vyřešen problém ndržíčích vodičů a zároveň byl ušetřen prostor, který obsazoval modul na SD kartu a nepájivé pole. Připojení Data Logger Shieldu k Arduino s připájenými vodiči lze vidět na obr. 20.



Obrázek 20: Data Logger Shield připojený k Arduino

7.1 Algoritmus řízení

Sestavení algoritmu celého systému napomohlo při softwarových úpravách. Tyto úpravy vedly k možnosti spouštět či ukončovat měření pomocí tlačítka. Schéma algoritmu bylo vykresleno pomocí aplikace *draw.io*. Algoritmus je zobrazen na obr. 21.



Obrázek 21: Algoritmus řídicí systém pomocí tlačítka

7.2 Software snímačů

Nezbytnou součástí řízení snímačů byl software, naprogramovaný v rozhraní Arduino IDE. Prvotní verze programu byla již rozpracovaná, avšak bylo nutné provést úpravy podle zadaných požadavků. V následujících odstavcích budou popsány nejdůležitější oblasti programu a nezbytné úpravy, které vedly k vylepšení systému. Každá z částí bude odkazovat na kód programu, který je součástí přílohy.

V úvodu první části programu (viz příloha 7.2.1) byly vloženy potřebné knihovny. Knihovnu enkóderu bylo nutné importovat do softwaru a následně vyvolat příkazem `#include <ams_as5048b.h>`. Dále byly definovány proměnné. Tyto proměnné definovaly vstupní hodnoty pro dané úkony (např. `int j = 0`) nebo určovaly cestu k jednotlivým komponentům, např. příkazem `int tlačítko = 5` je odkazováno na tlačítko připojené k pinu pět. Příkazem `char filename[] = "LOGGER00.txt"` byl stanoven název souboru ukládaného na SD kartu. Příponou `.txt` se data ukládají do textového souboru. V původním programu byla místo přípony `.txt` použita přípona `.csv`, díky níž se data ukládaly přímo do tabulek Excelu, což se může jevit jako výhodné, nicméně docházelo při takto rychlém importování dat k chybám SD karty. Proto byla zvolena přípona `.txt` jako vyhovující. Následně je příkazem `U_DEG 3` zvolena výstupní veličina ze snímačů, tedy v jednotkách stupňů. V posledních řádcích byla stanovena adresa obou enkóderů. Ta byla vyjádřena v hexadecimální (šestnáctkové) soustavě a byla zjištěna pomocí převzatého programu, který zjišťuje adresy I^2C sběrnic připojených k Arduino [15].

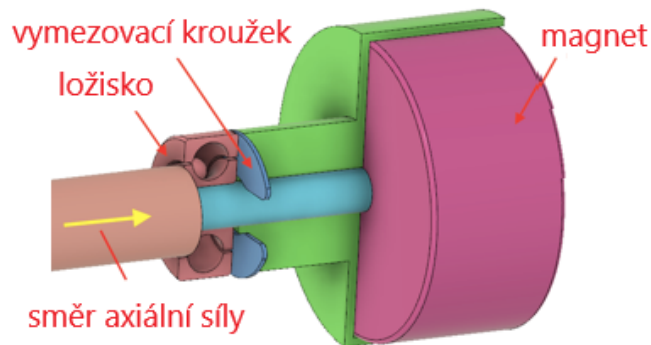
Druhá část programu (viz příloha 7.2.2) začíná smyčkou `void setup()`, ve které proběhlo definování vstupů a výstupů. Smyčkou `void loop()` začíná cyklus programu, který se stále opakuje. Cyklus prochází nejprve přes kontrolu stavu tlačítka. Princip fungování tlačítka je postaven na detekci změny stavu. Znamená to, že při každém stisknutí tlačítka se do aktuálního stavu ukládá určitá proměnná. Pokud je tlačítko zapnuto, program pokračuje ke kontrole přítomnosti SD karty. Tlačítko je v programu také zabezpečeno proti tzv. *debouncing* (neboli *ošetření zákmitů*). Průběh sepnutí tlačítka není vždy ideální a někdy může při sepnutí dojít k zákmitu, tedy rychlému sepnutí a vypnutí. Samotný zákmit trvá jen pár milisekund. Program provede 200 milisekundovou prodlevu (příkazem `long debounce = 200` v první části) od sepnutí a nedá čas tlačítku tento zákmit provést.

Ve třetí části programu (viz příloha 7.2.3) dojde k vytvoření souboru na SD kartě. Maximální počet vytvořených souborů na kartě je 99. Příkazem `pred2 = start` bylo zaručeno, aby se soubor během opakování cyklu vytvořil pouze jednou. Následně je spuštěno měření obou snímačů.

Úvod čtvrté části programu (viz příloha 7.2.4) slouží k ukládání naměřených dat na SD kartu. Pomocí podmínky `if (j>30)`, je docíleno toho, že se data na SD kartu ukládají po 30 naměřených hodnotách. Pokud by se ukládaly data po jednom, měření by velmi zpomaloval zápis na SD kartu. Následující část slouží k ukončování měření, kde dojde příkazem `AoA.close()` k uzavření souboru na SD kartě. Při vypnutém stavu měření je signalizační LED dioda zhasnuta.

7.3 Vymezení vůle mezi magnetem a ložiskem

Při sledování průběhu naměřených hodnot ze snímačů, byla zjištěna vůle mezi magnetem enkodéru a ložiskem. Při axiálním zatížení na ložiska docházelo k posunutí magnetu směrem k čipu. Toto posunutí způsoboval zákřivý v průběhu měření. Vůle byla následně vymezena pomocí kroužku o tloušťce přibližně 1 mm. Tento kroužek byl vytisknut na 3D tiskárně. Detailní pohled je zobrazen na obr. 22.



Obrázek 22: Vymezovací kroužek mezi magnetem a ložiskem

7.4 Trubice k umístění snímačů

Před samotnou kalibrací byly oba snímače připevněny k hliníkové trubici. Snímače byly vůči sobě umístěny v takové vzdálenosti, aby o sebe korouhvičky při natáčení nezavadily. Zároveň bylo nutné snímače vůči sobě natočit o 90 stupňů. Tedy aby jeden snímač snímal úhel náběhu a druhý úhel vybočení. V přední části trubice má být umístěna Pitotova sonda určena k měření dynamického tlaku. Zadní část trubice má sloužit k uchycení ke křídlu nebo trupu letounu. Umístění snímačů a kalibrační šablony na trubici lze vidět na obr. 23.



Obrázek 23: Snímače s kalibrační šablonou umístěné na hliníkové trubici

7.5 Kalibrace snímačů

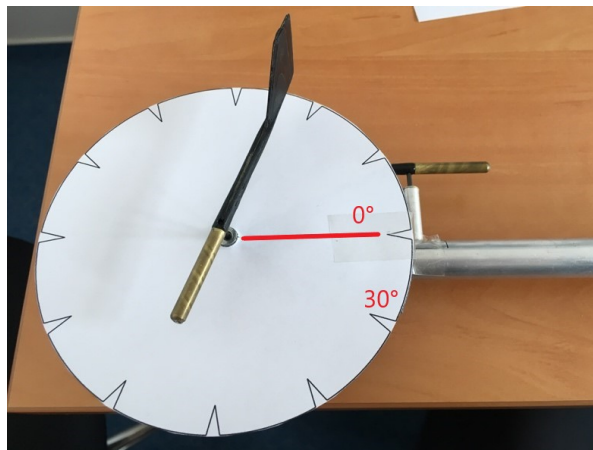
Hlavním úkolem kalibrace snímačů bylo pozorovat vztah mezi hodnotou skutečného vychýlení korouhvičky a naměřenou hodnotou vychýlení ze snímače. Kalibrace byla provedena ve dvou úhlových rozmezích. Nejprve v rozmezí 360 stupňů vždy po 30 stupních. Následně byla provedena detailnější kalibrace v rozmezí 50 stupňů s krokem po pěti stupních. Počáteční nulová hodnota natočení korouhvičky byla zvolena v místě, ve kterém by se korouhvička nacházela při nulovém úhlu náběhu.

7.5.1 Geometrická kalibrace v rozmezí 360 stupňů

Kalibrace v rozmezí 360 stupňů sloužila hlavně k vymezení intervalu, ve kterém se chování snímače zdá být nejstabilnější. Takovýmto ideálním chováním se myslí stav, ve kterém hodnota skutečného vychýlení korouhvičky koresponduje s naměřenou hodnotou vychýlení snímače. Pokud by tato závislost byla vynesena do grafu, její průběh by tedy v ideálním případě měl být lineární.

Před touto kalibrací bylo nutné připravit šablonu pro rozmezí 360 stupňů s vyznačenými ryskami vždy po třiceti stupních. Nejprve byla k trubici se snímači připevněna šablona s rozmezím 50 stupňů, která stanovila počáteční nulovou hodnotu natočení. Poté byla šablona s rozmezím 360 stupňů připevněna pod korouhvičku. Kladný směr otáčení byl volen ve směru hodinových ručiček. Její umístění pod korouhvičkou lze vidět na obr. 24.

Následně byly zaznamenány hodnoty měření. Měření proběhlo nejprve pro snímač umístěný v zadní části trubice a posléze pro snímač v přední části. Naměřené hodnoty byly v Excelu vyneseny do grafů se závislostí hodnoty natočení ze snímače na skutečném natočení korouhvičky.

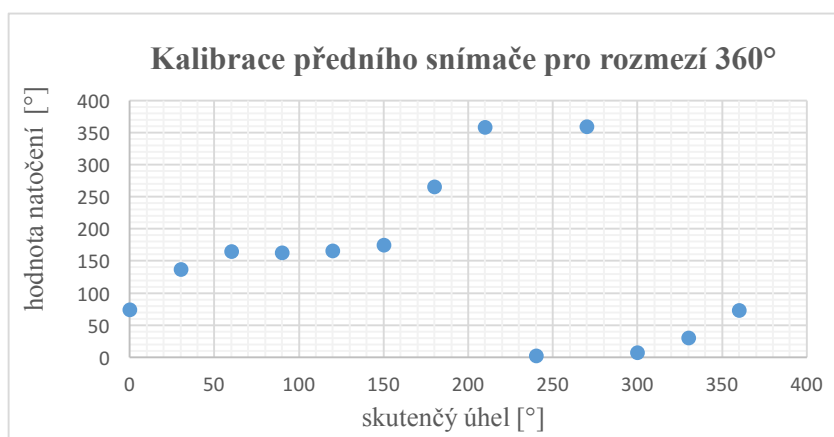


Obrázek 24: Kalibrační šablona pro rozmezí 360 stupňů

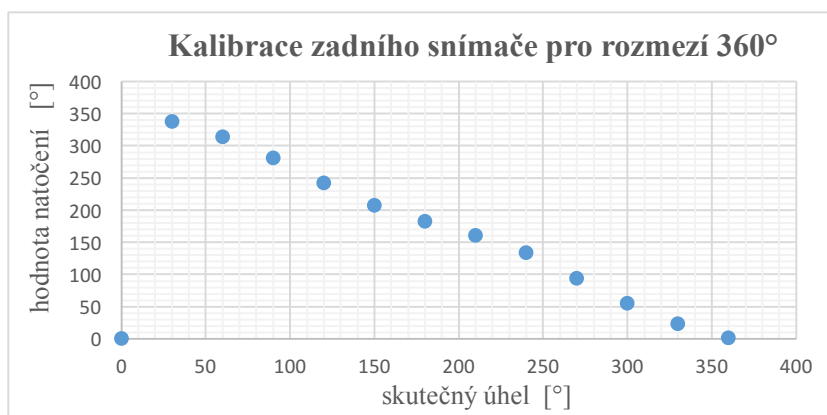
Při kalibraci zadního snímače byly zjištěna jistá anomálie průběhu hodnot natočení. Lze vidět, že v intervalu od 50 do 100 stupňů skutečného natočení, se hodnota ze snímačů téměř nemění. Ke vzniku této nepravidelnosti mohlo dojít z několika důvodů. Jednou z příčin může být nesymetrické umístění magnetu nad čipem. O problematice umístění magnetu vůči čipu je pojednáno detailněji v předchozí kapitole 7.1.2.

O něco stabilnější průběh hodnot u zadního snímače se nachází v intervalu od 330 do 360 stupňů a od 0 do 30 stupňů skutečného natočení. Jelikož přecházela korouhvička v tomto intervalu přes hodnotu nula stupňů skutečného natočení, nebylo nutné upravovat její umístění na snímači. Tento interval byl tedy zvolen jako výchozí stav pro další kalibraci.

Při kalibraci předního snímače, lze vidět o něco stabilnější průběh hodnot, přibližující se lineární závislosti. Při vychýlení korouhvičky z nulové polohy v kladném či záporném směru otáčení docházelo k přeskoku mezi hodnotami natočení 360 a nula stupňů. Tento přeskok mezi hodnotami nebyl příliš žádoucí. Jelikož nebyla korouhvička k snímači uchycena napevno, bylo možné ji na snímači natočit do takové polohy, aby k přeskoku hodnot nedocházelo.



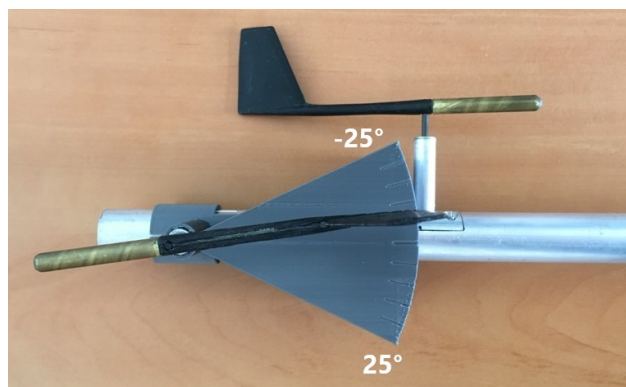
Graf 1: Závislost hodnoty úhlu natočení na skutečném úhlu natočení korouhvičky



Graf 2: Závislost hodnoty úhlu natočení na skutečném úhlu natočení korouhvičky

7.5.2 Geometrická kalibrace v rozmezí 50 stupňů

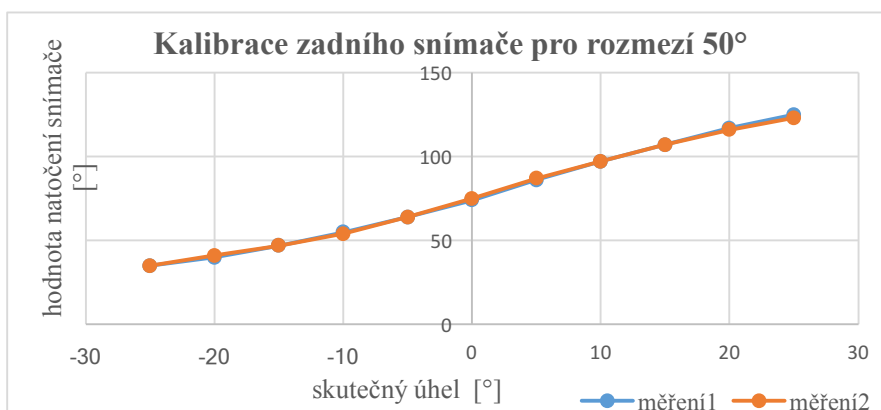
Při této kalibraci bylo voleno rozmezí, ve kterém se mohou korouhvičky pohybovat v průběhu letu. Princip měření je tedy stejný jako u předchozí kalibrace. Pro kalibraci byla s využitím 3D tisku vyrobena šablona, kterou bylo možné připevnit k trubici přímo pod snímač. Průběh natočení byl sledován s vychýlením korouhvičky v kladném směru o 25 stupňů a v záporném směru o -25 stupňů s krokem 5 stupňů. Měření bylo provedeno u každého snímače dvakrát. Umístění kalibračního úhloměru lze vidět na obr. 25.



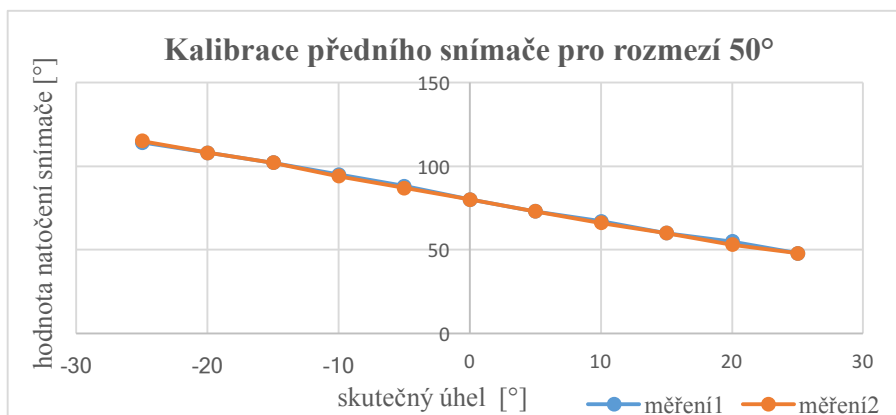
Obrázek 25: Kalibrační šablona pro rozmezí 50 stupňů

Princip kalibrace zadního snímače byl obdobný, jako u předchozí korouhvičky. Průběh naměřených hodnot by se měl opět v ideálním případě přibližovat lineární funkci. Lze vidět, že u zadního snímače se průběh hodnot v určitých rozmezích od lineárnosti odchyluje. Tato odchylka může opět vyházet z již zmíněného důvodu umístění magnetu (viz kapitola 7.5.1). U kalibrace předního snímače lze vidět, že se průběh přibližuje lineární funkci.

Naměřená data z kalibrace mají sloužit k aproximování vhodné funkce. Řídící jednotka by poté tuto funkci používala k výpočtu hledané hodnoty úhlu náběhu.



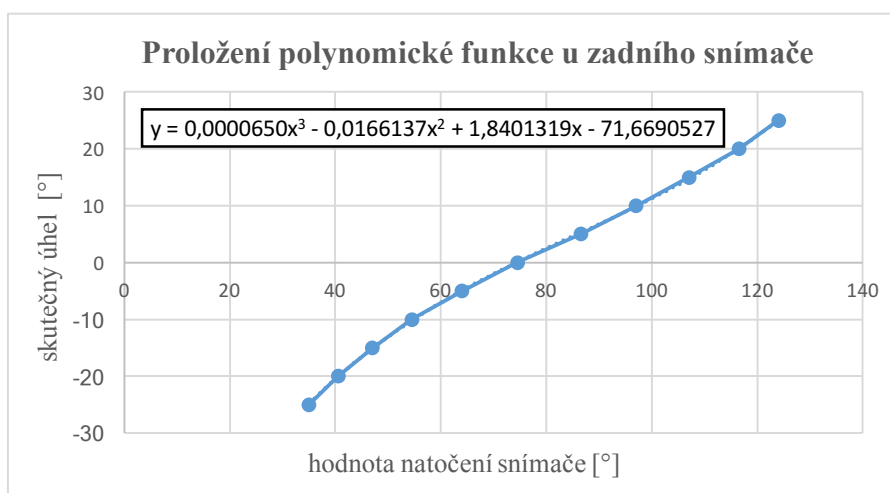
Graf 3: Závislost hodnoty úhlu natočení na skutečném úhlu natočení korouhvičky



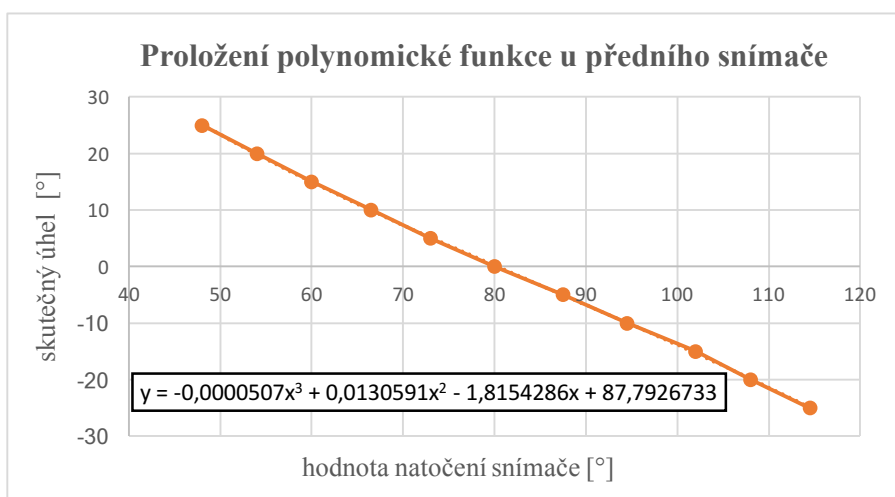
Graf 4: Závislost hodnoty úhlu natočení na skutečném úhlu natočení korouhvičky

7.5.3 Proložení funkce

K proložení funkce bylo nutné grafy z předchozí kalibrace lehce modifikovat. Vznikl tedy graf s průběhem průměrné hodnoty dvou měření, jehož osy, jsou oproti předchozím kalibracím vzájemně prohozeny. Prohození os bylo nezbytné, kvůli funkci, získané proloženou spojnici trendu naměřenými daty. Neznámou jsou totiž hodnoty na ose y , tedy hodnoty skutečného úhlu natočení korouhvičky. Smyslem této funkce $y = f(x)$, je přepočítávat naměřené hodnoty natočení snímače na neznámou veličinu skutečného natočení korouhvičky, tedy na hledaný úhel náběhu.



Graf 5: Získaná funkce, proložená naměřenými daty z kalibrace



Graf 6: Získaná funkce, proložená naměřenými daty z kalibrace

7.6 Další vývoj

Po samotné geometrické kalibraci by měla následovat aerodynamická kalibrace. Provádět by se dala v aerodynamickém tunelu, kde by byla korouhvička obtékána proudem vzduchu a zjišťovaly by se její aerodynamické vlastnosti. Geometrie korouhvičky by měla být v ideálním případě symetrická, tak, aby na ni nevznikala vztlačková síla.

Výstupem kalibrace je proložená polynomická funkce, která přepočítává hodnoty z Arduina na hodnoty hledaného úhlu náběhu. Bude-li nutné dále zpracovávat signál v Arduinu, je třeba polynomickou funkci nahradit tabulkou hodnot nebo nějakou zjednodušenou funkcí, protože Arduino je vhodné zejména pro celočíselné výpočty.

Samotný přepočet úhlu natočení enkodéru z napěťových signálů probíhá přes knihovnu v softwaru Arduina. Přesnost tohoto přepočítaného úhlu je vlivem omezeného výkonu Arduina poměrně nízká. Hodnoty úhlu jsou zaokrouhlovány na celé stupně. Pro přesnější měření by se dal použít výkonnější mikrokontrolér.

Řešit tuto nepřesnost by se dalo, provedením kalibrace pro hodnoty napěťových signálů. Závislost by byla tedy mezi skutečným natočením korouhvičky a polohou z enkodéru. Těchto poloh dosáhne enkodér již zmíněných 16385 za jednu otáčku. Následně by byla proložena vhodná funkce, přes kterou by se zjistila poloha pro daný úhel náběhu. Poloha by byla dále přepočtem převedena na úhel ve stupních. Předěšlo by se tak nepřesné veličině úhlu natočení, která je převáděna knihovnou Arduina.

8 Závěr

Cílem práce bylo dokončit rozpracovaný prototyp systému sestaveného ze dvou korouhviček, které slouží k měření úhlu náběhu a úhlu vybočení.

Úvodní část práce pojednává o platformě Arduina a její hardwarové a softwarové části. Popisuje typy desek Arduina, které se objevují na trhu a dále rozebírá výhody a nevýhody využití této platformy.

V druhé části práce je věnována pozornost teoretickým základům z oblasti aerometrických veličin, konkrétně veličině úhlu náběhu. Důležitou částí je definování samotné veličiny úhlu náběhu a pojednání o jejím vlivu na vztlak. Dále jsou zde uvedeny používané typy snímačů úhlu náběhu a způsoby jakými probíhá signalizace před přibližujícím se přetažením letounu.

Třetí část práce je věnována samotnému dokončení návrhu systému. Nejprve jsou zde popsány důležité komponenty systému a je detailně pojednáno o funkci rotačního magnetického enkodéru.

V prvním kroku při realizaci systému byly všechny komponenty zapojeny k Arduinu. Následně byl pro systém naprogramován software v aplikaci Arduino IDE. Výsledkem této části byl funkční systém, který snímá úhel natočení korouhviček a zapisuje naměřená data na SD kartu. Tuto část lze tedy považovat za úspěšnou.

V druhém kroku proběhla geometrická kalibrace snímačů s využitím úhlových šablon. Výsledky kalibrace ukázaly, že jeden ze snímačů vykazoval v určitých intervalech odchylky od předpokládaného průběhu. Řešením bylo zvolit interval, ve kterém k těmto odchýlením nedocházelo. Naměřenými daty z kalibrace byla proložena funkce, sloužící k přepočtu hodnoty ze snímače na hodnotu natočení korouhvičky, tedy na hledaný úhel náběhu. Geometrickou kalibraci lze považovat za splněnou. Pokračováním v realizaci systému by bylo vhodné ještě ověřit výsledky geometrické kalibrace kalibrací aerodynamickou.

9 Seznam použitých zdrojů

Publikace

- [1] SELECKÝ, Matúš a Martin HERODEK. Arduino: uživatelská příručka. Brno: Computer Press, 2016, 344 stran : ilustrace ; 23 cm. ISBN 978-80-251-4840-2.
- [5] BROŽ, Václav, Vladimír DANĚK, Karol FILÁKOVSKÝ a Jaromír TEICHL. Základy aerodynamiky: studijní modul 8. Brno: Akademické nakladatelství CERM s.r.o, 2003, 163 stran : ilustrace. ISBN 80-7204-316-1.
- [6] DANĚK, Vladimír a Karol FILÁKOVSKÝ. Základy letu (081 00): [učební texty dle předpisu JAR-FCL 1]. Brno: Akademické nakladatelství CERM, s.r.o, 2006, 314 stran : ilustrace. ISBN 80-7204-449-4.
- [7] ČORBA, Stanislav. Aerodynamika, konstrukce a systémy letadel: studijní modul 13. Brno: Akademické nakladatelství CERM, 2004, 450 s. : il. ; 30 cm. ISBN 978-80-7204-395-8.
- [9] GLINZ, J. Létání při nízkých rychlostech. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2012. 32 s. Vedoucí bakalářské práce Ing. Róbert Šošovička, Ph.D.
- [11] KOZELSKÝ, M. Metody zpracování dat z letových zkoušek. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2008. 30 s. Vedoucí bakalářské práce Ing. Ivo Jebáek, Ph.D.
- [12] POPOWSKI, Stanisław a Witold DĄBROWSKI. Measurement and estimation of the angle of attack and the angle of sideslip. Aviation [online]. Taylor & Francis, 2015, 19(1), 19-24 [cit. 2019-04-21]. DOI: 10.3846/16487788.2015.1015293. ISSN 1648-7788.
- [13] EVANS, Julien. Jak létají dopravní letadla. Ramsbury: Crowood, 2011, 128 s. ISBN 978-80-247-8169-3.

Elektronické zdroje

- [2] CO JE TO ARDUINO?, Arduino.cz. [online]. [cit. 2019-04-23]. Dostupné z: <https://arduino.cz/co-je-to-arduino/>
- [3] BŘESKÝ, Jan. Mini průvodce světem Arduino. [online]. [cit. 2019-04-22]. Dostupné z: https://mujweb.fmk.utb.cz/~j_bresky/arduino_interactive_2.pdf
- [4] 2. díl o Arduinu – typy desek. PHGame.cz. [online]. [cit. 2019-04-22]. Dostupné z: https://phgame.cz/PHGame_serialy/serialy/zaciname-s-arduinem/2-dil-o-arduinu-typy-desek
- [8] Vznik mezní vrstvy tekutiny. Encyklopedie fyziky. [online]. [cit. 2019-04-22]. Dostupné z: <http://fyzika.jreichl.com/main.article/view/554-vznik-mezni-vrstvy-tekutiny>

- [10] Základní pojmy: Přetažení. Opentx.cz. [online]. [aktualizace: 4. 3. 2019 20:13]. [cit. 2019-04-22]. Dostupné z: https://www.opentx.cz/index.php/Základn%C3%AD_pojmy:_Přetažen%C3%AD
- [14] Angle of attack. Aero 12. [online]. [cit. 2019-04-22]. Dostupné z: http://www.boeing.com/commercial/aeromagazine/aero_12/attack_story.html
- [15] Krátký průvodce I2C (TWI, wire) sběrnici. Arduino návody. [online]. [cit. 2019-04-22] Dostupné z: <https://navody.arduino-shop.cz/technikuv-blog/zjisteni-adresy-i2c-twi-wire-zarizeni.html>
- [16] Magnetické senzory s Hallovým efektem. Automatizace.hw.cz. [online]. [cit. 2019-05-14] Dostupné z: <https://automatizace.hw.cz/magneticke-senzory-s-hallovym-efektem-1-princip>
- [17] AS5048A/AS5048B. Mouser Electronics. [online] [cit. 2019-05-17] Dostupné z: https://cz.mouser.com/datasheet/2/588/AS5048_DS000298_4-00-1100510.pdf
- [18] Arduin Uno R3. Robotech shop. [online] [cit. 2019-05-20] Dostupné z: <http://robotechshop.com/shop/arduino/arduino-board/arduino-uno-r3-china/?v=928568b84963>

10 Přílohy

Příloha 7.2.1 První část programu

```
KOROUHEV_TLACITKO_FINAL §
1 #include <SPI.h>
2 #include <SD.h>           // vloží knihovnu SD karty
3 #include <ams_as5048b.h> // vloží knihovnu enkóderů
4
5 int tlacitko = 5;         // vstup (pin5) pro tlacitko
6 int zeleny = 2;          // výstup (pin2) pro zelenou LEDku
7 //int cerveny = 6;
8
9 int j = 0;
10 //int zapis = 0;
11 const int CSpin = 10;    //propojeni CS s SD kartou
12 File AoA;
13 //String dataString = "";
14 //boolean closefile = true;
15 char filename[] = "LOGGER00.txt"; // název souboru ukládaný na SD kartu
16 unsigned long startTime = 0;      // promenna startTime = 0
17 unsigned long stopTime = 0;       // promenna stopTime = 0
18 String buff = "";
19 unsigned long val1 = 0;            // promenna val1 = 0
20 unsigned long val2 = 0;            // promenna val2 = 0
21
22 int start = 1;                    // promenné sloužící k spouštění systému tlacitkem
23 int konec = 1;
24 int pred1 = 0;
25 int pred2 = 0;
26
27 int stav = HIGH;                 // proměnné k funkci switch-tlačítka
28 int cteni;
29 int predchozi = LOW;
30 int korouhev;
31
32 long time = 0;
33 long debounce = 200;            // čas sloužící ke kontrole záskmitů tlačítka
34
35 #define U_DEG 3                  // definování veličiny úhlu
36 AMS_AS5048B mysensor1(0x40);    // stanovení adres dvou enkóderů
37 AMS_AS5048B mysensor2(0x41);
```

Příloha 7.2.2 Druhá část programu

```
38
39 void setup() {                               // smyčka setup(), která proběhne jednou
40   Serial.begin(9600);
41   pinMode(tlacitko, INPUT);                  // tlačítka jako vstup
42   pinMode(zeleny, OUTPUT);                  // LED jako výstup
43   //pinMode(cerveny, OUTPUT);
44 }
45
46 void loop() {                                // stále se opakující smyčka loop()
47   // SPOUŠTĚNÍ MĚŘENÍ ZMAČKNUTÍM TLAČÍTKA
48   cteni = digitalRead(tlacitko);            //zjistí stav tlačítka a uloží jako proměnnou
49
50   if (cteni == HIGH && predchozi == LOW && millis() - time > debounce) { // kontrola zámků tlačítka
51     if (stav == HIGH){
52       stav = LOW;
53       pred1 = 0;
54     }
55     else{
56       stav = HIGH;
57       pred2 = 0;
58     }
59     time = millis();
60   }
61   // KONTROLA PŘÍTOMNOSTI SD KARTY
62   if(stav == LOW){
63     if(start == 1 && pred2 == 0){
64       Serial.print("\nInitializing SD card..."); // rozpoznává SD kartu
65       pinMode(CSpin, OUTPUT);                  // CS pin jako výstup
66       if (!SD.begin(CSpin)){                  // pokud ji nerozpozná napíše chybu
67         Serial.println("Card failed, or not present");
68         return;
69       }
70       Serial.println("card initialized.");      // napíše rozpoznání SD karty
71   }
```

Příloha 7.2.3 Třetí část programu

```
71
72 // VYTVÁŘENÍ SOUBORŮ NA SD KARTĚ
73   for (uint8_t i = 0; i < 100; i++) {          // vytváření nového souboru, max 99 souborů
74     filename[6] = i/10 + '0';                 // očísluje soubor
75     filename[7] = i%10 + '0';
76     if (! SD.exists(filename)) {             // pokud na SD kartě soubor není, otevře ji a
77       AoA = SD.open(filename, FILE_WRITE);    // otevře SD a vloží soubor
78       break;
79     }
80   }
81
82   if (! AoA) { // pokud nebude moct vytvořit nový soubor (např. přesažení paměti)- hodí chybu
83     Serial.println("couldnt create file");
84   }
85
86   AoA.close(); // zavření složky
87   pred2 = start; // slouží k tomu, aby vytvoření souboru proběhlo během cyklení pouze jednou
88 }
89 // SPOUŠTĚNÍ ENKÓDERŮ
90 mysensor1.begin();                          // spustí měření senzorů
91 mysensor2.begin();
92 for (int i=0;i<10;i++){
93   val1=mysensor1.angleR(U_DEG, true); // zjistí hodnotu "úhlu natočení" snímače
94   val2=mysensor2.angleR(U_DEG, true);
95   j++; // počítadlo hodnoty "j", díky kterému nedochází k přetěžování SD karty
96 }
```

Příloha 7.2.4 Čtvrtá část programu

```
97 // ČÁST UKLÁDÁNÍ NA SD KARTU
98 //Serial.print(String(millis())+" "); // vypíše "casu" na monitoru
99 Serial.print(val1); // vypíše "úhel" snímače 1 na monitoru
100 Serial.print(" "); // uděla mezeru
101 Serial.println(val2); // vypíše "úhel" snímače 2 na monitoru
102 buff = String(buff)+"\n" + String(millis())+";" +String(val1)+";" +String(val2);
103 // do proměnné "buff" uloží "cas" a "úhly natočení" snímaču 1 a 2
104 if (j>30) { // aby se na SD kartu neukládala hodnota po hodnotě, zjistí 30 hodnot a ty následně uloží
105     AoA = SD.open(filename, FILE_WRITE); // otevře soubor na SD kartě
106     AoA.print(buff); // vloží do souboru hodnoty
107     buff = "";
108     AoA.close(); // uzavře soubor
109     j = 0; // začne znovu, dokud j=30
110 }
111 }
112 // UKONČENÍ MĚŘENÍ ZMÁČKNUTÍM TLAČÍTKA
113 if(stav == HIGH){ // slouží pro vypnutí měření
114     if(konec == 1 && pred1 == 0){
115         Serial.println("Uzavření souboru"); // napíše "uzavření souboru" na monitoru
116         AoA.close(); // uzavře soubor kam se ukládaly data
117         pred1 = konec;
118     }
119 }
120 // INFORMUJE O STAVU SNÍMAČŮ ( MĚŘÍ/NEMĚŘÍ)
121 digitalWrite(zeleny, !stav); // rozsvítí zelenou LEDku- PROBÍHÁ MĚŘENÍ
122 digitalWrite(cerveny, stav); // rozsvítí červenou LEDku- NEPROBÍHÁ MĚŘENÍ
123 predchozi = cteni;
124 }
```