

České vysoké učení technické v Praze
Fakulta elektrotechnická



Bakalářská práce

2015

Adam Půta

České vysoké učení technické v Praze

Fakulta elektrotechnická

katedra mikroelektroniky



Bakalářská práce

Řídicí jednotka modelu solárního letadla

Autor: Adam Půta

Vedoucí práce: Ing. Lubor Jirásek, CSc.

2015

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra mikroelektroniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **P Ů T A Adam**

Studijní program: Komunikace, multimédia a elektronika
Obor: Aplikovaná elektronika

Název tématu: **Řídicí jednotka modelu solárního letadla**

Pokyny pro vypracování:

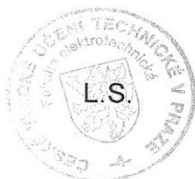
1. Prostudujte dostupnou literaturu týkající se problematiky řízení modelů letadel. Zaměřte se speciálně na řešení řídicího systému, sběr dat ze sensorového systému a jejich vyhodnocení.
2. Na základu 1) navrhnete a realizujete řídicí systém, systém sběru dat ze sensorového systému a způsoby vyhodnocení těchto dat pro model solárního letadla.
3. Navrhnete postup vhodný pro ověření funkce přístroje.
4. Provedte ověřovací měření na zařízení.
5. Zhodnotte dosažené výsledky.
6. Publikování výsledků dosažených v této práci je možné pouze se svolením zadavatele.

Seznam odborné literatury:

- [1] Husák, M.: Mikrosenzory a mikroaktuátory. Academia, Praha 2008.
- [2] Reese, R., B.: Microprocessors from Assembly Language to C. Da Vinci Engineering Press, Hingham Massachusetts 2005.
- [3] Krejčířík, A.: Napájecí zdroje I.-III. Ben, Praha 1996 a pozdější vydání.
- [4] Katalogové listy součástek a aplikační poznámky výrobců.

Vedoucí: **Ing. Lubor Jirásek, CSc.**

Platnost zadání: 31. 8. 2016



prof. Ing. Miroslav Husák, CSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 10. 2. 2015

Čestné prohlášení

Prohlašuji, že jsem zadanou diplomovou (bakalářskou) práci „Řídicí jednotka modelu solárního letadla“ zpracoval sám s přispěním vedoucího práce a používal jsem pouze literaturu uvedenou na konci práce. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 18. 4. 2015.

Adam Půta

Poděkování

Na úvod mé bakalářské práce bych rád poděkoval rodině za projevenou podporu, dále pak vedoucímu práce za cenné připomínky a rady a nakonec také fakultě samotné za možnost používat její laboratoře k vyhotovení práce a měření výstupů z ní vyplývajících.

Abstrakt

Tato bakalářská práce se zabývá problematikou stabilizace dálkově ovládaného modelu solárního letounu a příčinami vzniku nestability během řízení. Dále má za úkol navrhnout vhodný komunikační subsystém pro příjem dat ze sensorové jednotky a rádiového přijímače, vybrat vhodný mikroprocesorový systém, navrhnout a zrealizovat obvod obsahující výše uvedené prvky a nakonec provést kontrolní měření a zhodnocení celého zařízení.

Klíčová slova

letoun, stabilizace, regulátor, sběrnice, mikropočítač

Abstract

This bachelor thesis deals with problematics of stabilization a remote-controlled solar airplane and cause of instability during flight. Also this thesis have to come up with solution for communication with sensory unit and ground control station. Final part brings a design of electronic circuit and results of measuring performed on whole control unit.

Key words

airplane, stabilization, regulator, bus, microcomputer

<i>Abstrakt</i>	7
<i>Klíčová slova</i>	7
letoun, stabilizace, regulátor, sběrnice, mikropočítač.....	7
<i>Abstract</i>	7
1. <i>Úvod</i>	11
2. <i>Nestabilita letu</i>	12
2.1. <i>Příčiny nestability během řízení</i>	12
2.1.1. <i>Vliv větru</i>	13
2.1.2. <i>Vliv momentu vrtule</i>	14
2.2. <i>Regulátory pro stabilizaci letu</i>	14
2.2.1. <i>P složka</i>	17
2.2.2. <i>I složka</i>	18
2.2.3. <i>D složka</i>	19
3. <i>Sběrnicevé systémy pro přenos dat</i>	20
3.1. <i>Požadavky na datové sběrnice</i>	20
3.2. <i>Typy datových sběrnic</i>	20
3.2.3. <i>I²C</i>	26
3.3. <i>Napěťové přizpůsobení komunikujících zařízení</i>	28
3.4. <i>Zhodnocení parametrů a výběr vhodného typu sběrnice</i>	31
4. <i>Realizace systému</i>	32
4.1. <i>Popis hardwaru</i>	32
4.2. <i>Software</i>	35
4.2.1. <i>Algoritmus řízení</i>	36
5. <i>Závěr</i>	37
6. <i>Příloha A</i>	37
7. <i>Příloha B</i>	37
8. <i>Seznam použité literatury</i>	37
9. <i>Obrázky</i>	37
10. <i>Tabulky</i>	37
11. <i>Použité zkratky</i>	37
12. <i>Obsah CD</i>	37

1. Úvod

Je tomu už přes sto let, kdy bratří Wrightové poprvé opustili zemi a vznesli se k nebesům. Od té doby svět letectví neodmyslitelnou součástí naší moderní doby. Létat dnes může opravdu každý, a to nejenom jako pilot skutečného letadla, ale i jako pilot modelář. S rozvojem elektrotechniky, informatiky, strojírenství, atd. je možné stavět stále sofistikovanější stroje schopné takzvaného autonomního letu. Drony, jak jsou tyto stroje nazývány, mají široké uplatnění všude tam, kde je potřeba zvýšit efektivitu a bezpečnost letu. Slouží například v bezpečnostních složkách státu jako je armáda, policie, hasičský a záchranný sbor. Dále mohou být dobrým pracovním nástrojem ve vědecko-technické sféře. Sem patří měření stavu počasí, kontrola venkovního vedení elektrické energie, ale i kontrola stavu takových systémů, které při svém provozu vytvářejí velice nebezpečné podmínky pro člověka, jako je například kontrola jaderných zařízení.

Jak ale takový letoun (dron) pohánět a jaký zdroj energie pro tento účel zvolit? Protože je naším cílem vytvořit dálkově řízený model solárního letounu, je tímto zdrojem **sluneční záření**. Tím pádem je však nutné použít solární články. Ty samy o sobě nejsou schopné transformovat a dodat takové množství energie, aby byl letoun schopen letu jen díky nim. Jsou ale schopné prodloužit let tím, že dobíjejí akumulátorový systém, který v konečné fázi dodá letounu potřebný výkon.

Dále je potřeba vyřešit otázku **řízení a stabilizace** modelu kvůli odstranění vlivu povětrnostních podmínek na letoun a tím zlepšit jeho ovladatelnost.

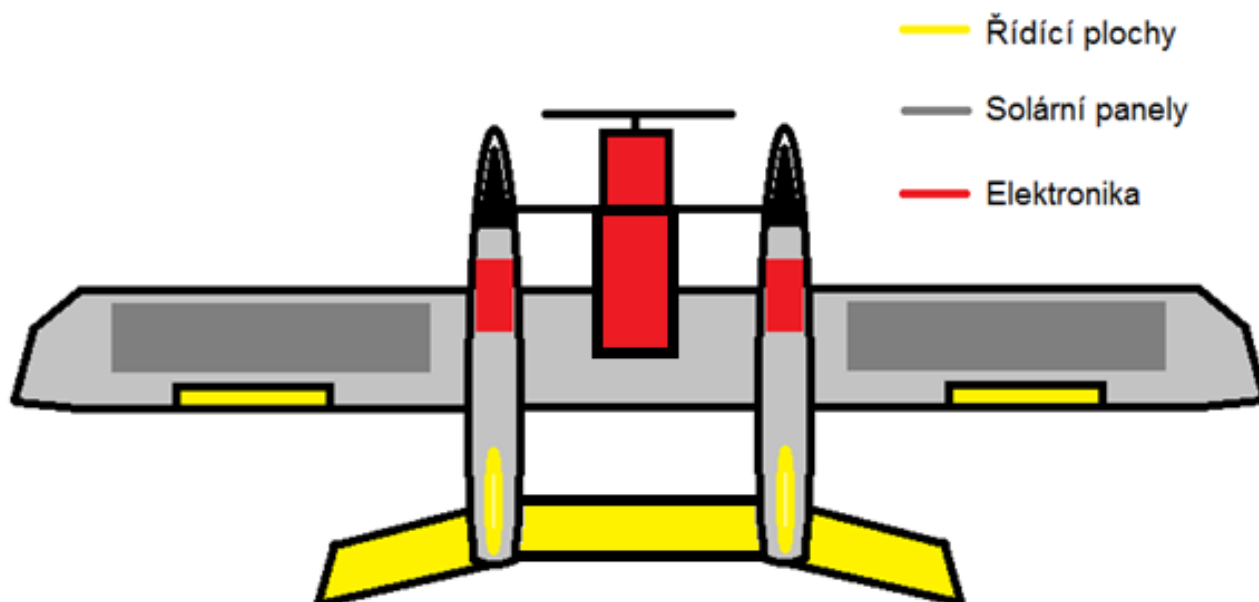
Pokud ovšem chceme provádět stabilizační korekce během letu, je tím pádem nutné získávat informace o aktuálním stavu letounu z hlediska jeho chování v prostoru. K tomuto účelu slouží **senzorická jednotka** podávající aktuální informace o rychlosti stroje, výšce, náklonu atd..

2. Nestabilita letu

2.1. Příčiny nestability během řízení

Už od prvních krůčků lidstva v oblasti letectví si lidé uvědomovali potřebu letoun stabilizovat. Jednak proto, že při letu v turbulenci nebo za větrného počasí, dochází k odklonu letounu od trajektorie letu, kterou chce stroj (pilot) letět, ale i kvůli namáhání konstrukce, které vzniká vibracemi způsobenými obtékáním tělesa vzduchem při určitých rychlostech, též známém jako flutter.

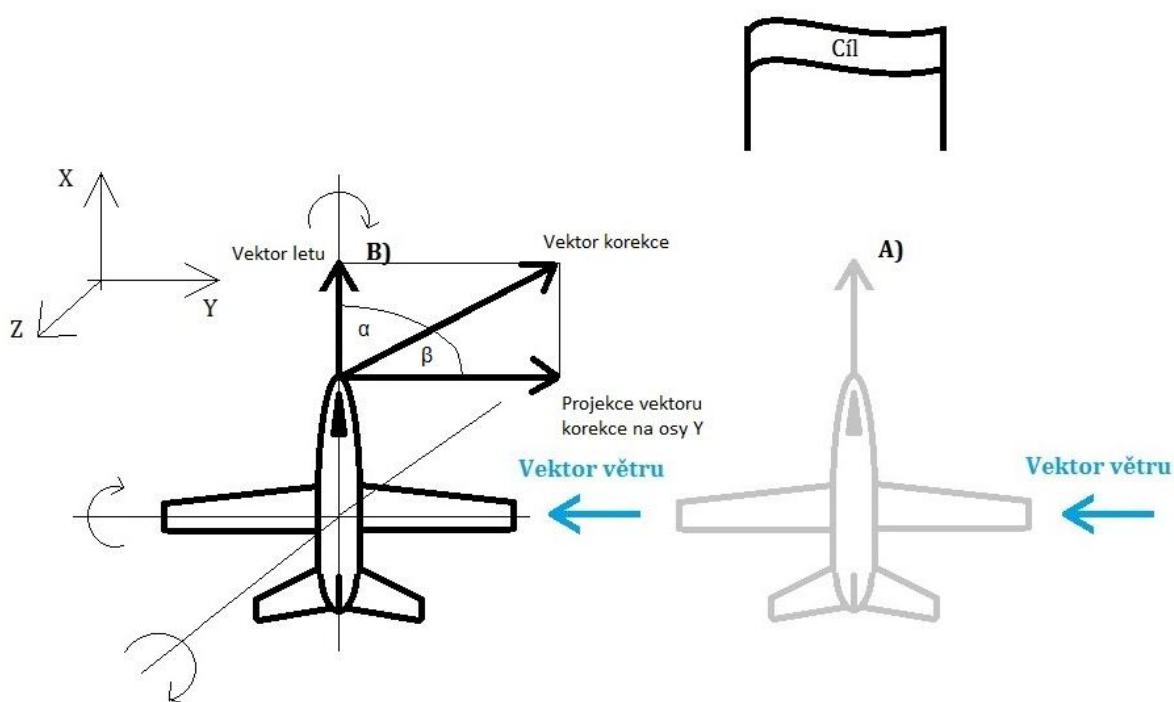
Další problém může nastat i při samotném pilotování, kdy pilot může nedopatřením provést manévř, kterým uvede stroj do nebezpečné situace, jako je například pád způsobený ztrátou rychlosti, nebo při tzv. přetažení letounu a následnou ztrátou vztlaku, ale i manévř o kterém si zprvu myslí, že ho zvládne, ale jehož obtížnost nakonec přesáhne pilotovi schopnosti. Nemaleu míru nestability může u motorového letounu způsobit i reakční moment vrtule, nebo hmotnostní nevyvážení stroje. U modelů či klasických letadel obecně s pohony na chemická paliva (benzín, kerosin, atd..) může k nestabilitě přispět i rozdílné množství paliva v nádržích a přelévání onoho paliva při zatáčení nebo akceleraci (deceleraci). V letectví se lze setkat i s konstrukcemi draků letadel, které schválně zvyšují nestabilitu a tím zvyšují svoji schopnost lépe manévřovat.



Obr. 2.1 Testovací letoun

2.1.1. Vliv větru

Vlivem větru může dojít, jak bylo řečeno k odklonu od požadované trajektorie. Obr. 2.2 ukazuje vliv větru na letící letadlo. Pro úplnost dodejme, že vznik větru je zapříčiněn rozdílnou teplotou v atmosféře, a tím způsobeným rozdílem tlaků v určitých částech atmosféry. Příklad A) je počáteční směr letu k danému cíli. Příklad B) znázorňuje letoun odchýlený od jeho původní trajektorie a nutnost provést korekci kurzu o úhel korekce α , který svírá vektor letu a vektor korekce tak, aby bylo dosaženo cíle.

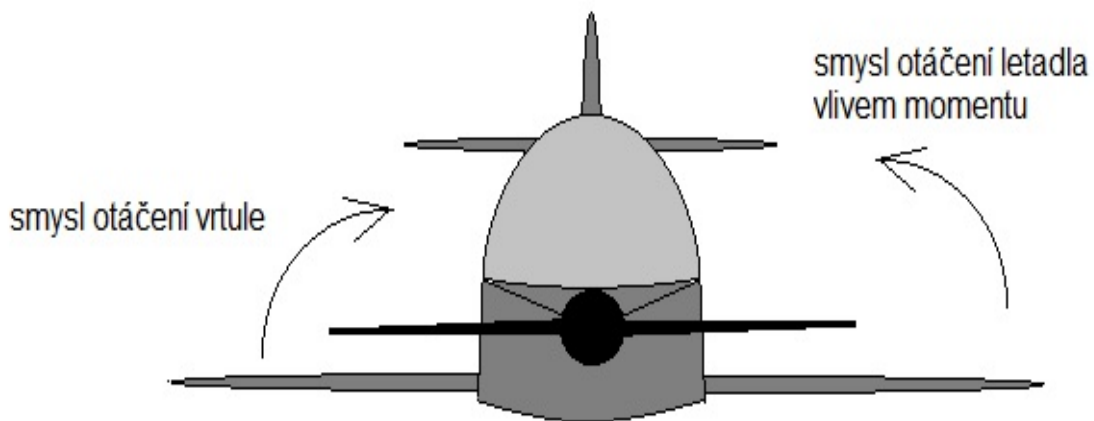


Obr. 2.2 Vliv větru na trajektorii letu

Korekci provede buď pilot osobně nebo autopilot, je-li let v automatickém módu řízení. Úhel korekce α by se dal zjistit například ze souřadnic cíle a aktuálních souřadnic letounu s použitím goniometrických funkcí a Pythagorovy věty. Vítr může mít vliv nejenom na vychýlení z dráhy letu, ale i na změnu výšky a natáčení stroje kolem podélné (X), příčné (Y) a svislé (Z) osy. Stejně účinky může mít vliv turbulentního proudění vzduchu způsobeného neuspořádaným pohybem vzduchu a vlivem termického proudění, které vzniká nad ohřátým terénem, nebo tříštěním masy vzduchu o mechanickou překážku.

2.1.2. Vliv momentu vrtule

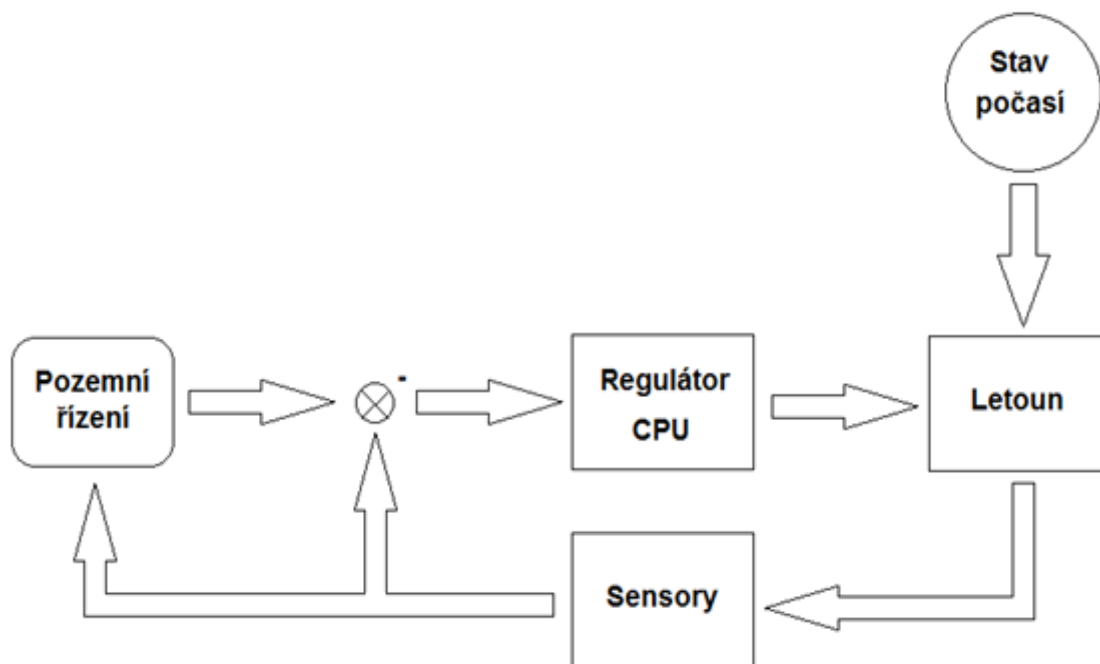
Dalším druhem nestability řízení může být reakční moment vyvolaný otáčející se vrtulí. Ten způsobí, že se letadlo začne otáčet kolem podélné osy na opačnou stranu, než je smysl otáčení vrtule. U dvou motorového stroje je tento moment vyrušen otáčením obou vrtulí proti sobě. Máme – li však k dispozici pouze jednu pohonnou jednotku, je potřeba provést takové konstrukční opatření, aby došlo k jeho vykompenzování (řešením je malá ploška na konci křídla, která je vychýlena tak, že kompenzuje otáčení vlivem momentu), nebo provést takové řídicí zásahy, které povedou ke stabilizaci letadla. Vliv momentu vrtule je patrný na obr. 2.3. Další druh momentu, který způsobuje vrtule, je moment gyroskopický. Ten se projeví při zatáčení stroje (otáčení okolo svislé osy) zatáčením na pravou nebo levou stranu. Opět zde ale můžeme vhodným zásahem do řízení jeho vliv eliminovat.



Obr. 2.3 Vliv momentu

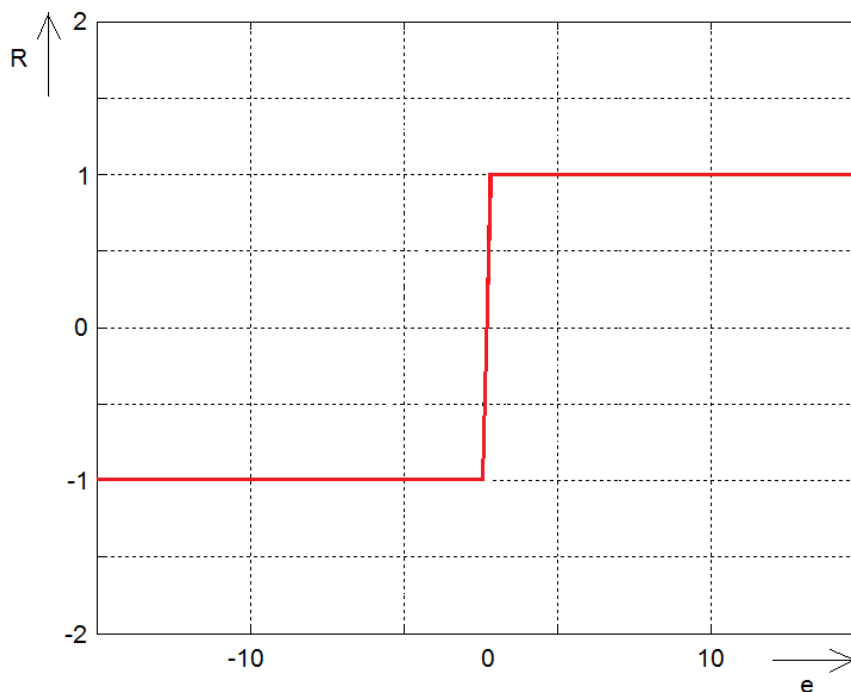
2.2. Regulátory pro stabilizaci letu

Pro stabilizaci, a tím vyrušení nežádoucích vlivů v řízení letounu, je vhodné použít stabilizační regulátor. Na obr. 2.4 je letoun zobrazen jako řízená soustava.

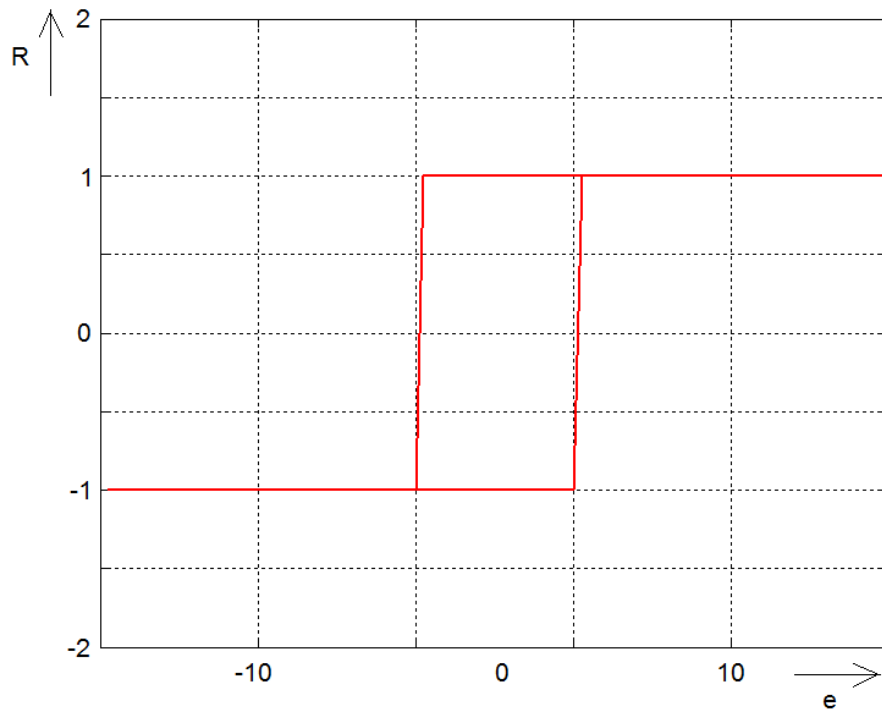


Obr. 2.4 Blokové schéma řízené soustavy

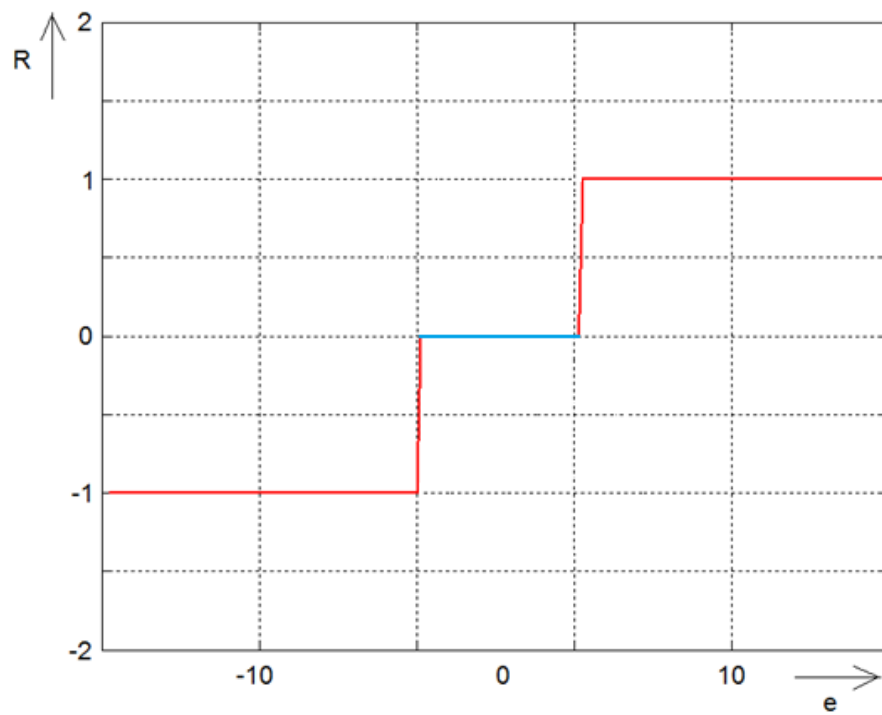
Existuje mnoho možností, jak letoun prostorově stabilizovat. Od jednoduchého „On - Off“ řízení, jehož řídicí zásahy jsou neúměrné odchylce, tj. rozdílu



Obr. 2.5 Závislost řídicího zásahu na odchylce „On-Off“ regulátoru ($\text{sign}(e)$)



Obr. 2.6 Hystereze regulátoru „On-Off“



Obr. 2.7 Modře vyznačené pásmo necitlivosti regulátoru

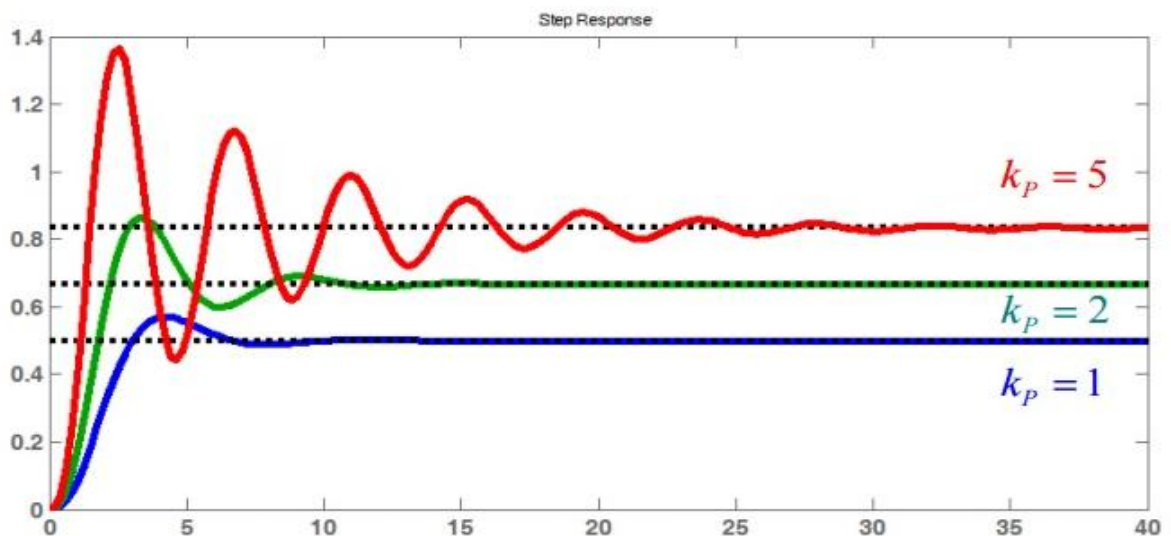
žádané a skutečné hodnoty regulované veličiny, přes PID řízení až po prediktivní systémy, které pracují s modelem řízené soustavy. Podívejme se tedy na další možnost, kterou je PID nebo-li proporcionálně integračně derivačního regulátor. Jak název napovídá, skládá se ze tří složek: P, I a D složky. Každá složka zde plní odlišnou funkci. Od regulačního systému obecně očekáváme malou hysterezi a malé pásmo necitlivosti. Je také možné použít samostatné složky jako celý regulátor.

2.2.1. P složka

Proporcionální složka regulátoru je v podstatě obyčejný zesilovač, který zesílí regulační odchylku e , která vznikne rozdílem mezi požadovanou (referenční, angl. set point) hodnotou w a skutečnou (naměřenou) hodnotou y . Vztah pro výpočet odchylky:

$$e = w - y \quad (2.1)$$

Tato odchylka se násobí tzv. proporcionální konstantou značenou k_p a která udává ono zesílení. Vezměme si třeba autopilota v letadle. Pokud chceme, aby letoun z původních 150 km/h letěl například 300 km/h, nastavíme požadovanou rychlost (setpoint) v palubním počítači, letadlo začne zrychlovat a autopilot začne současně měřit skutečnou rychlost. Za předpokladu, že náš autopilot funguje jako P regulátor, by v ideálním případě vytvořil takový zásah do řízení rychlosti, aby letoun už dále nezrychloval. To ale ve skutečném světě nelze a proto autopilot buď o malou hodnotu nastavenou rychlost překročí, nebo přestane zrychlovat těsně pod její hranici. To zda přestane s akcelerací dříve, či později záleží na výběru hodnoty k_p . Následující obrázek ukazuje vliv hodnoty k_p na regulaci.



Obr. 2.8 Vliv konstanty k_p na regulovanou veličinu [1a]

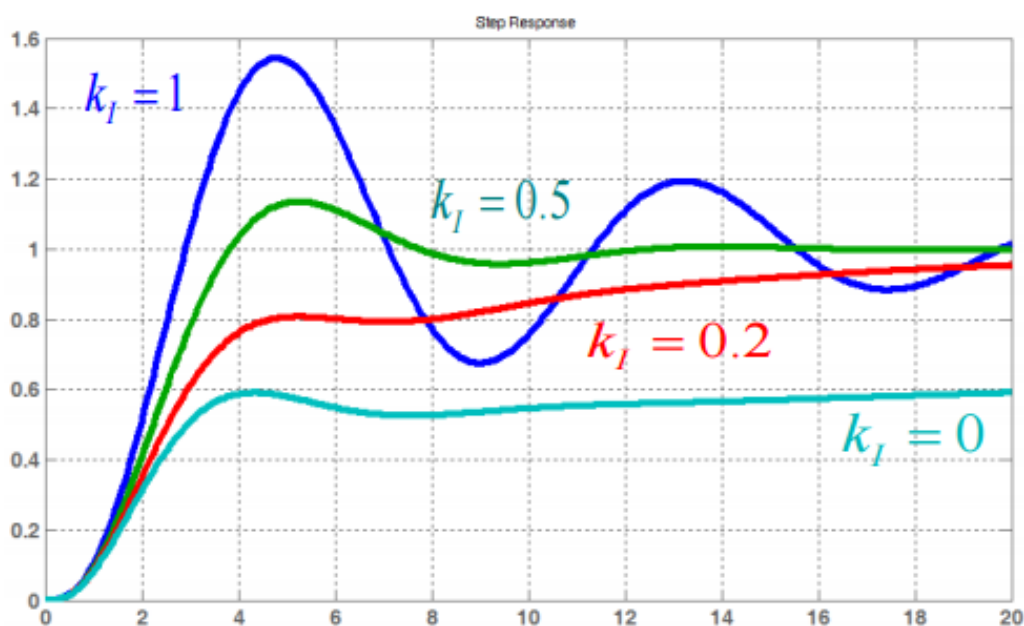
Je-li hodnota k_p příliš malá, dojde k ustálení pod regulovanou veličinou. Při optimálně zvolené hodnotě dojde k ustálení na požadované hodnotě a při příliš vysoké hodnotě dojde k překročení požadované hodnoty. Zvláště u posledního případu je důležité dávat pozor, aby regulovaná veličina nepřekročila požadovanou hodnotu příliš a nedošlo tím k poškození řízeného systému. Záleží i na době působení onoho překročení. Čím kratší je doba překročení, tím lépe. Výpočet P složky je následující:

$$P = k_p * e(t) \quad (2.2)$$

kde P je hodnota regulátoru (P), k_p zesilovací konstanta a $e(t)$ vypočtená odchylka.

2.2.2. I složka

Integrační složka slouží k rychlejšímu přibližování na požadovanou hodnotu. Zesilovací konstanta označená k_I má tu vlastnost, že čím větší má hodnotu, tím rychleji by se měla regulační odchylka blížit nule.



Obr. 2.9 Vliv konstanty k_I na regulovanou veličinu [1b]

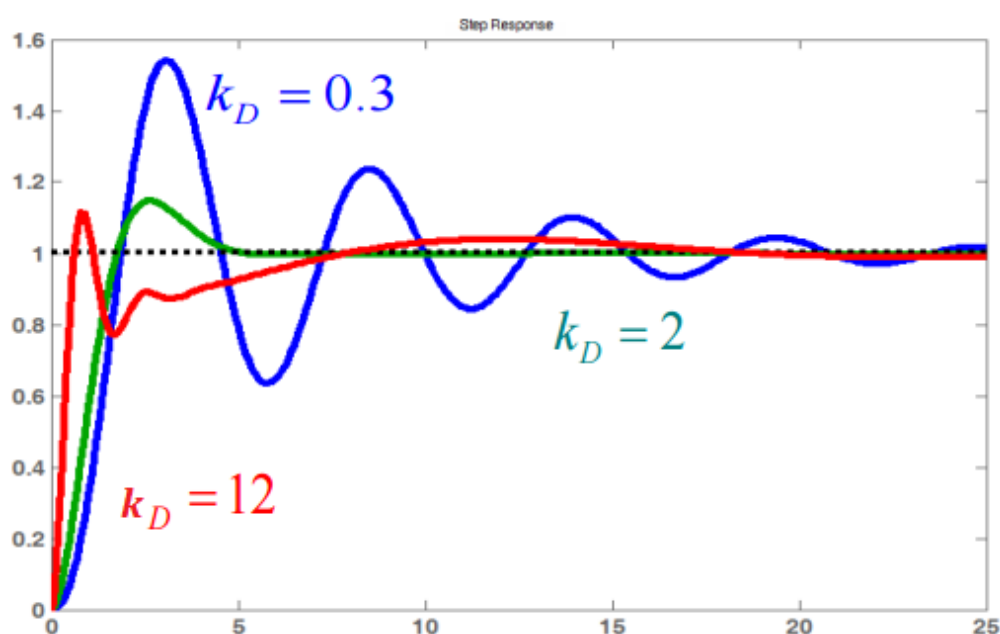
Nevýhodou je, že čím větší je k_I , tím více regulovaná veličina kmitá. Vztah pro integrační složku je následující:

$$I = k_I \int_0^t e(\tau) * d\tau \quad (2.3)$$

Kde I je hodnota regulátoru (I), k_I zesilovací konstanta, $e(t)$ vypočtená odchylka a t doba integrace. Dle vztahu (2.3) je patrné, že integrace probíhá na zvolené délce časového úseku.

2.2.3. D složka

D složka tlumí akční zásah regulátoru, což má za následek to, že s přibližující se požadovanou hodnotou řízené veličiny řídicí veličina neroste tak rychle a to znamená, že zmenší přesah požadované veličiny. Také D složka může být zesílena konstantou k_D .



Obr. 2.10 Vliv konstanty k_D na regulovanou veličinu [1c]

Matematické vyjádření D složky je následující:

$$D = k_D \dot{e} \quad (2.4)$$

přičemž D je hodnota regulátoru (D), k_D zesilovací konstanta a \dot{e} časová derivace odchylky. Konečná podoba rovnice PID regulátoru je tedy:

$$PID = k_p * P + k_i * I + k_d * D \quad (2.5)$$

Vzhledem k neefektivnímu chování systému přepínacího typu „On - Off“ (funguje relativně spolehlivě jen u řízených soustav s velkým zpožděním), složitosti prediktivních systémů a systémů na bázi PID, byl zvolen přístup pomocí P regulátoru. Jeho implementací se budeme zabývat v kapitole 4. *Realizace systému.*

3. Sběrníkové systémy pro přenos dat

3.1. Požadavky na datové sběrnice

Datové sběrnice jsou jedny ze základních stavebních kamenů dnešních moderních „embedded“ systémů. Využívají se všude tam, kde je potřeba rychlý přenos dat, neboli zajištění rychlé komunikace mezi součástmi elektrotechnického zařízení nebo mezi jednotlivými zařízeními obecně.

U sběrníkových systémů je kladen důraz na jejich bezpečnost provozu a to jak z hlediska bezpečnosti v elektrotechnice tj. dodržení daných norem pro elektrotechnické/elektronické zařízení, tak i z hlediska funkčnosti. U metalických vedení, která jsou dnes stále nejrozšířenější je potřeba brát v úvahu například délku a průřez vodiče (z hlediska odporu a indukčnosti) nebo vzdálenost mezi vodiči (z hlediska kapacity). Tyto a další parametry mají podíl na útlumu a časovém zpoždění. Dále může být špatně navržený komunikační systém citlivý na okolní rušení a tím pádem na ztrátu dat, nebo se také může stát zdrojem rušení. Mezi další moderní technologie můžeme zmínit optická vedení, neboli optické vlnovody, kterými se ale v rámci této práce zabývat nebudeme.

Mezi další bezpečnostní požadavky patří také ochrana přenášených dat před zneužitím jinými osobami nebo systémy. Špatně zabezpečený komunikační kanál může v konečném důsledku znamenat ohrožení života, či majetku.

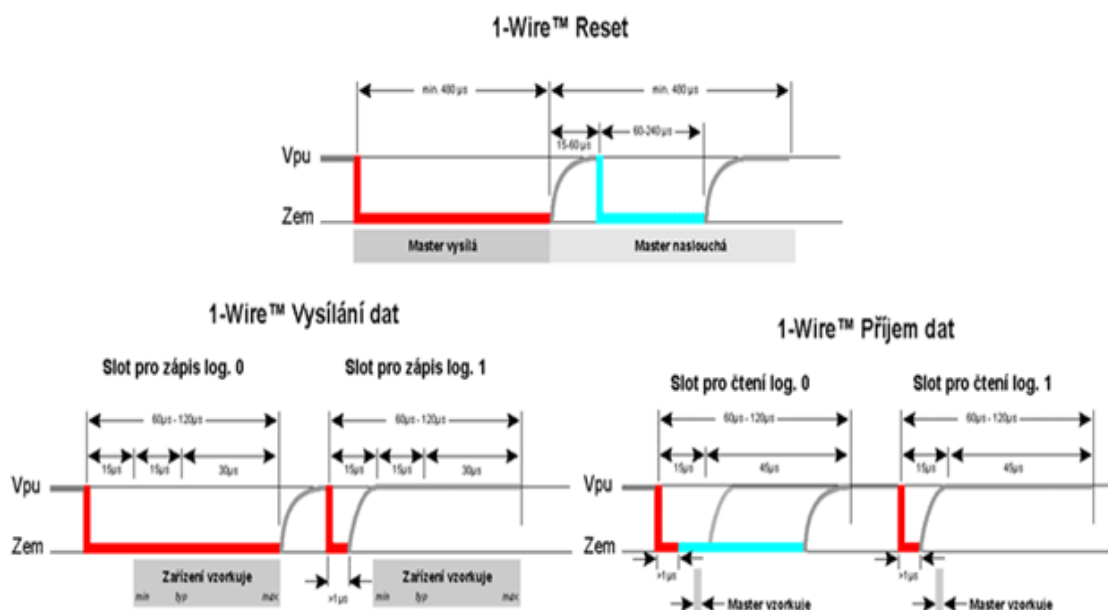
3.2. Typy datových sběrnic

Mezi nejrozšířenější typy datových sběrnic patří sběrnice One-Wire, SPI, I²C, RS 232 a 485, atd. Rozeberme tedy funkci typů I²C, SPI a One-Wire a zhodnotíme, která se pro naši aplikaci hodí nejlépe. Systémy RS 232 a RS 485 jsou vhodné spíše pro průmyslové aplikace a nebudou dále rozebírány, protože jsou náročné z hlediska hardwarové implementace a také kvůli jejich malé pracovní rychlosti.

3.2.1. One - Wire

Typ sběrnice One – Wire je, co se hardwarového zapojení týče, nejméně náročná. Jak název napovídá, k propojení zařízení, které v rámci sběrnice komunikují je potřeba pouze jednoho vodiče. Ze softwarového hlediska se však jedná o velice složitý systém, který si vyžaduje dobrou znalost komunikujícího hardwaru. Obr. 1 ukazuje obvodové zapojení sběrnice. Z elektrického hlediska je samozřejmě nutný společný vodič GND z hlediska společného zemnění. Po aktivaci napájení a uvedení systému do chodu je potřeba, aby zařízení master zjistilo adresy všech obvodů slave na lince a to v případě, že je na sběrnici více zařízení. **Každé z nich má 64bitovou adresu, neboli ID** (unikátní číslo). Postup zjišťování adresy je relativně složitý (princip vyhledávání pomocí binárního stromu) a vyžaduje příkaz „Search ROM“, pomocí kterého se zjistí vždy jeden bit z adresy a jehož hodnota je vstupní hodnotou do již zmíněného algoritmu binárního stromu. Ve chvíli, kdy master zná adresy všech obvodů slave, může je začít adresovat, ale až po odeslání příkazu „Match ROM“, který zajistí, že **na odeslanou adresu odpoví konkrétní zařízení**. Někdy je potřeba, aby se komunikace zúčastnila všechna zařízení najednou, tj. aby všechny obvody slave přečetly data najednou. K tomuto účelu slouží příkaz „Skip ROM“. Jako další příkaz uveďme například „Alarm search“. Na tento příkaz reagují pouze zařízení v poplachovém stavu (např. tlakové čidlo, u kterého tlak překročí povolenou hodnotu).

Komunikace probíhá pomocí tzv. „time slots“ nebo-li časových oken. Master nejprve „stáhne“ datový vodič do úrovně log. nula pomocí tzv. resetovacího pulzu. Na této úrovni je vodič držen v rozmezí od 480 μ s do 640 μ s. Poté master vrátí sběrnici zpět do stavu log. jedna a čeká zda-li se ozve nějaké zařízení slave. Obvod slave detekuje nástupnou hranu (z log. 0 na log. 1) a po zhruba 15 μ s až 60 μ s stáhne sběrnici opět do stavu log. nula na 60 μ s až 240 μ s. Nyní už obvod master může číst nebo zapisovat data na sběrnici a to pomocí již zmíněných time slotů, jejichž délka bývá od 60 μ s do 120 μ s. Jedno časové okno přenáší informaci o velikosti jeden bit. Průběh komunikace je vidět na obr. 3.1.



Legenda:

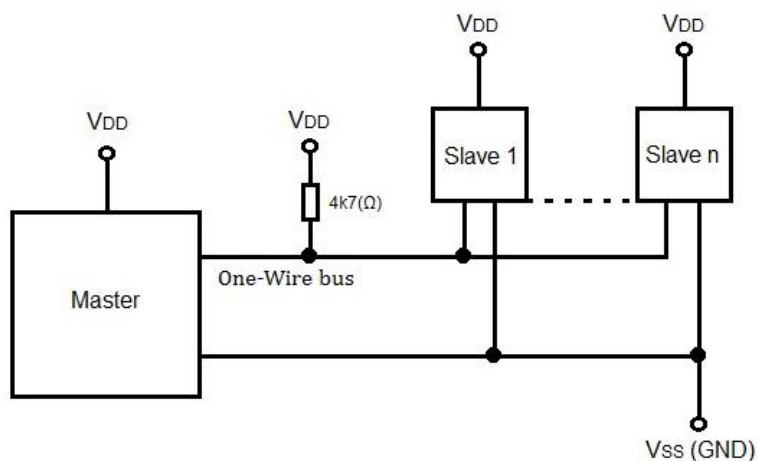
- master stahuje linku na log. 0
- slave stahuje linku na log. 0
- nastavení log. 1 pomocí pull-up rezistoru

Obr. 3.1 Průběh komunikace One-Wire [2]

Mezi časovými okny je potřeba nechat sběrnici minimálně $1(\mu\text{s})$. Existují čtyři druhy těchto time slotů:

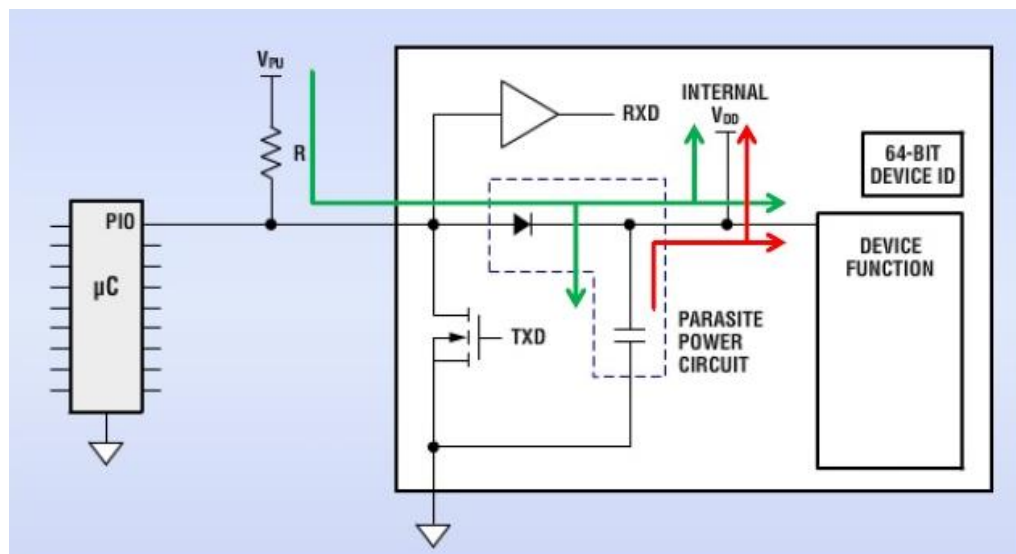
- Zápis log. 1
- Zápis log. 0
- Čtení log. 1
- Čtení log. 0

Každý z těchto druhů je dán svým konkrétním časováním, kdy je slot v logické jedničce a nule.



Obr. 3.2 Zapojení One-Wire sběrnice

One – Wire systém nabízí i možnost napájet komunikující zařízení po datovém vodiči, jak je vidět na obr. 3.3.



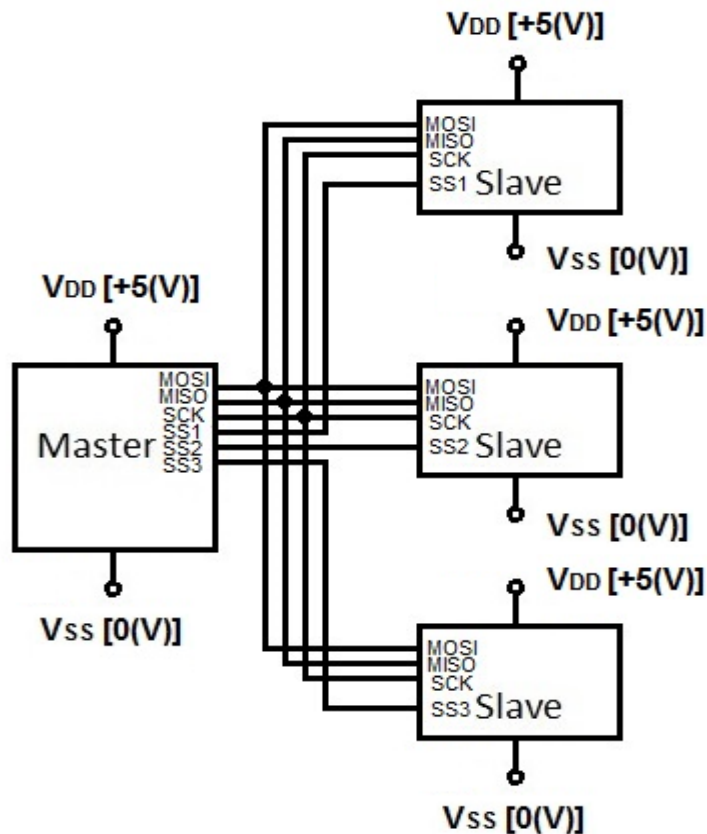
Obr. 3.3 Napájení přes One-Wire [3]

Pro programové nastavení je vhodné použít knihovnu z prostředí mikroC „One_Wire“.

3.2.2. SPI

SPI sběrnice vyvinutá firmou Motorola, je z hlediska počtu potřebných pinů na zařízení nejnáročnější. Kromě dvou datových vodičů (MOSI pro odesílání dat do slave obvodu a MISO pro čtení dat ze slave obvodu) je zde také potřebný hodinový signál SCK. Dále je potřeba takzvaný „výběrový“ vodič SS (slave select, někdy je možno setkat se s označením CS – chip select), který vybere konkrétní slave obvod pro komunikaci. Je-li slave obvodů v systému více, pak počet SS vodičů narůstá lineárně.

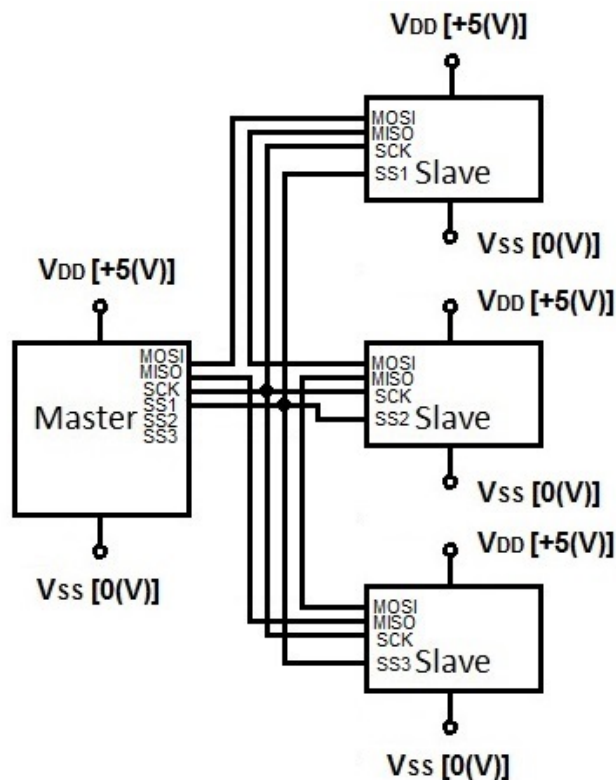
Konkrétní zapojení systému je vidět na Obr. 3.4. Pin MOSI zařízení master je propojen se všemi piny MOSI zařízení slave. To samé platí i pro piny MISO a CLK. Jak již bylo řečeno výše, pin SS vybírá jednotlivá zařízení, které mají v danou chvíli komunikovat s obvodem master a je jich tedy v obvodu master potřeba tolik, kolik je obvodů slave v komunikačním systému.



Obr. 3.4 SPI

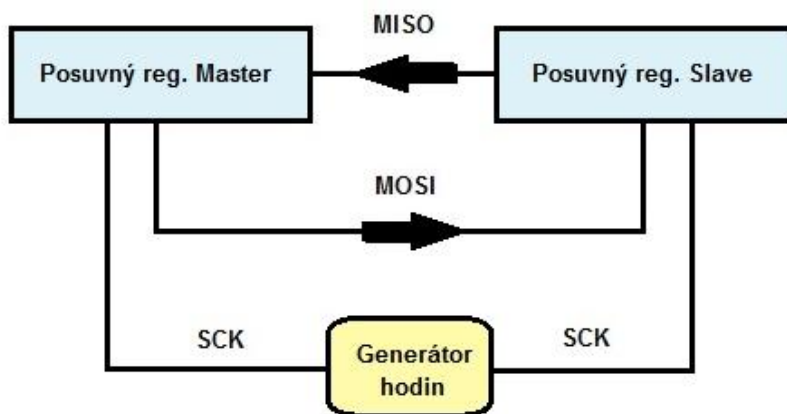
Z toho vyplývá, že každý pin SS zařízení slave je propojen s jedním z několika pinů zařízení master, tak jak je vyobrazeno na Obr. 3.4.

Komunikační rozhraní je též možné provozovat v zapojení nazvaném „Daisy Chain“, u kterého je pin MOSI obvodu master zapojen na pin MOSI pouze do prvního zařízení slave. Z tohoto obvodu slave je pak vyveden výstup MISO do dalšího zařízení slave a to konkrétně do vývodu MOSI. Takto se pokračuje dál podle toho kolik ke sběrnici připojeno zařízení slave. Z posledního zařízení slave je vývod MISO zapojen do vývodu MISO obvodu master. Další rozdíl je v zapojení vodiče SS, který vede z masteru do všech obvodů slave. Obr. 3.5 názorně popisuje dané zapojení.

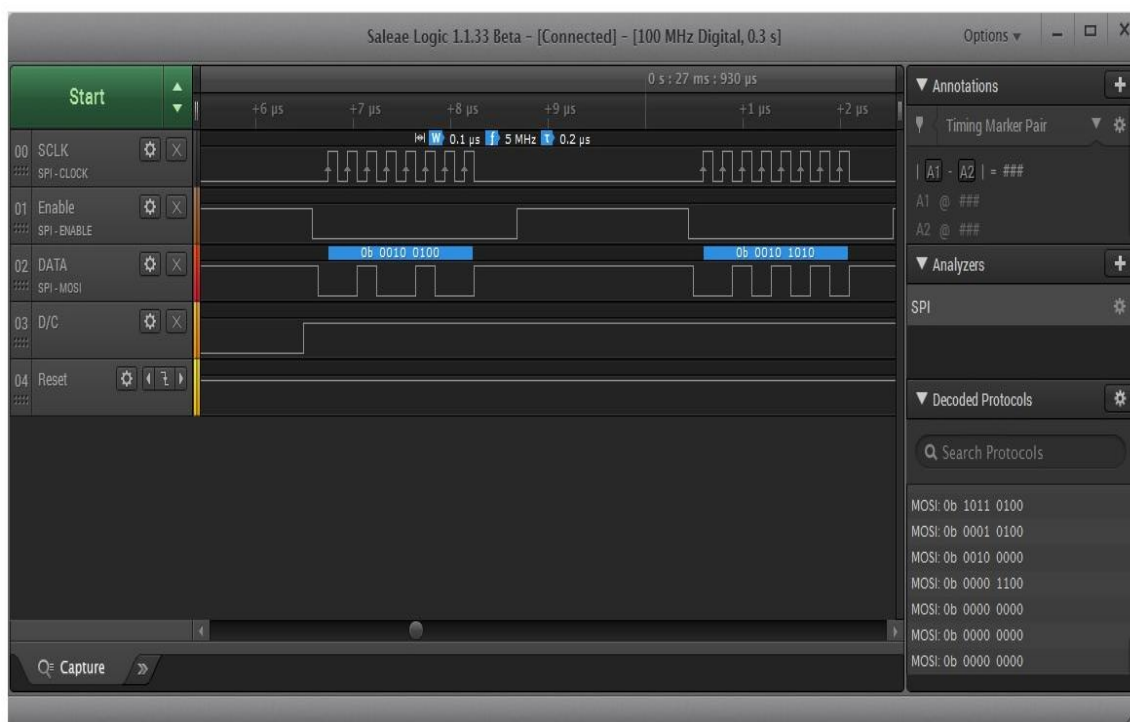


Obr. 3.5 Daisy Chain

Komunikace se uskutečňuje vždy mezi masterem a příslušným slavem, který je **vybrán vodičem SS**. Samotný přesun dat je realizován díky posouvání v posuvném registru, které jsou obsaženy jak v obvodu master tak v obvodu slave. Zjednodušené schéma průběhu posuvu dat je vidět na obr. 3.6. Obr. 3.7 ukazuje konkrétní průběhy signálů při datové komunikaci. Měření bylo provedeno USB logickým analyzátozem od firmy Sealea.



Obr. 3.6 Datový tok v SPI



Obr. 3.7 Měření SPI

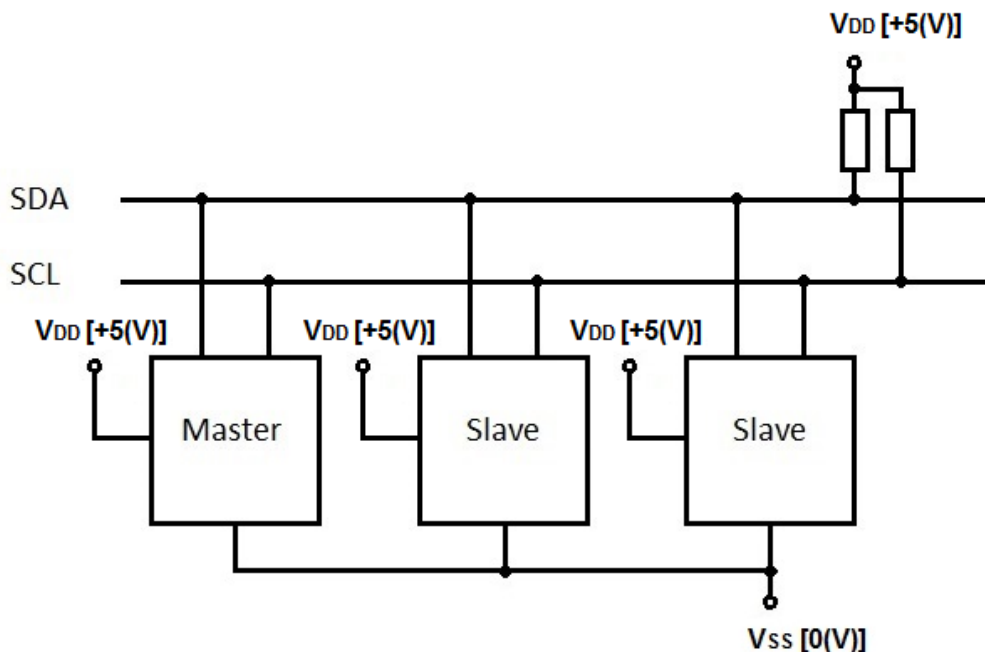
Stejně jako u One-Wire komunikace, tak u SPI platí, že ze softwarového hlediska je lepší použít implementované knihovny pro práci s SPI rozhraním ve vývojovém prostředí a to konkrétně knihovnu „SPI“ nebo „Software_SPI“. Parametry knihovny „Software_SPI“ jsou popsány zde:

- režim master
- frekvence hodinového signálu F_{CLK} 20 kHz
- vzorkování dat uprostřed intervalu
- klidový stav hodin je v log. 0

Knihovna „SPI“ se liší v nastavení frekvence hodin F_{CLK} . Konkrétně se jedná o to, že frekvence hodin je dána vztahem: $F_{CLK} = F_{OSC}/4$. Z tohoto vztahu je patrná závislost na taktovací frekvenci řídicího obvodu master a z toho vyplývající možnost nastavení daného kmitočtu.

3.2.3. I²C

I²C je sběrnice vyvinutá firmou Philips, která pro svojí funkci potřebuje dva vodiče. Jeden vodič pro přenos/příjem dat a druhý pro hodinový signál, který je důležitý z hlediska synchronizace přenosu. Každý vodič sběrnice je nutné připojit přes pull-up rezistor na napětí V_{DD} . Komunikující zařízení se připojí ke sběrnici tak, že se **připojí zvláště datový a hodinový vodič na příslušný pin zařízení**. Konkrétní zapojení ukazuje obr. 3.8. Dále je samozřejmě nutné propojit V_{SS} všech zařízení.



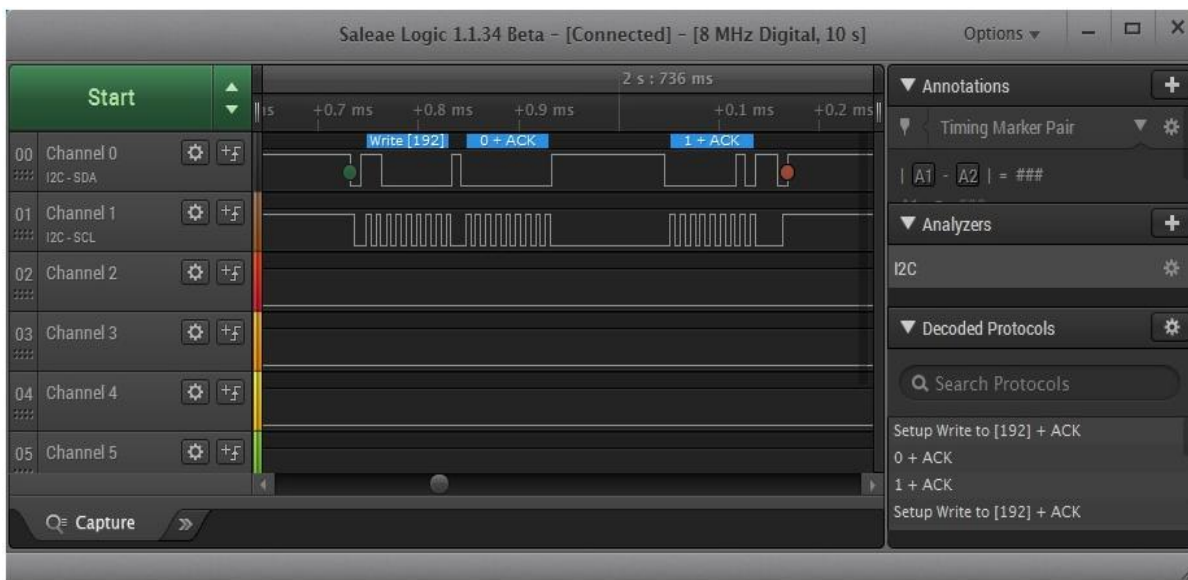
Obr. 3.8 I²C zapojení

Na sběrnici je možné připojit omezené množství zařízení. Důvodem je omezení v podobě kapacity na vedení, která nesmí přesáhnout 400 pF. Pracovní kmitočet signálu SCL je dle standardu 100 kHz nebo 400 kHz. Je možno však použít i frekvence vyšší (např. 1,7 MHz). Zajímavostí je i to, že se v prostředí mikroC můžeme setkat s frekvencí 20kHz a to konkrétně v knihovně „Software_ I²C“.

I²C sběrnice je typu multimaster a to znamená, že zde může být připojeno více obvodů master, nebo-li obvodů, které mají oprávnění řídit datovou komunikaci. Komunikace se řídí následujícím protokolem.

Oba vodiče jsou nejprve ve stavu H vlivem připojení přes pull-up rezistor na napětí V_{DD}. Dále následuje vyslání takzvané START podmínky od obvodu master. Tím dojde ke stažení SDA vodiče do úrovně L, zatím co SCL vodič je stále v úrovni H, ale pouze do chvíle, kdy zahájeno odesílání/přijímání dat. Dále je po SDA lince vyslána sedmi nebo desetibitová adresa, která slouží k výběru obvodu slave, s nímž má být zahájena komunikace. V praxi se setkáme s termínem arbitrace. Dodejme, že **logická úroveň na lince SDA se mění pouze, je-li SCL v rovní L**. Za adresou následuje R/W bit sloužící k definici toho, zda má dojít ke čtení, nebo zápisu dat (úroveň H je pro čtení dat ze slave, úroveň L pro zápis na slave). Jakmile dojde k přijetí adresy konkrétním zařízením, vyšle ono zařízení potvrzovací nebo-li ACK (acknowledge) bit. Master po detekci ACK bitu začne posílat data. Po odeslání všech osmi bitů dat vyšle slave další ACK bit. Pak může obvod master buďto posílat další data, nebo pošle další adresu a tím naváže komunikaci s jiným obvodem slave. V případě čtení dat ze slavu pošle ACK bit obvod master.

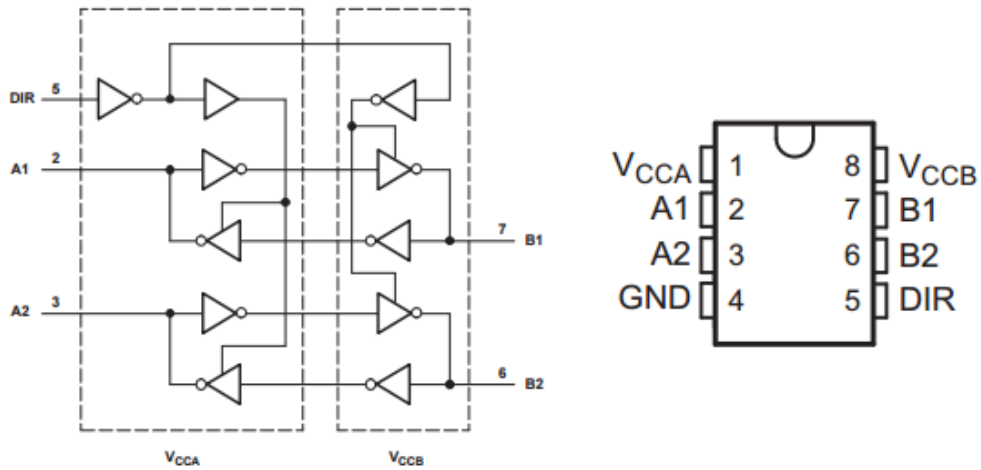
Pro ukončení se vyšle STOP podmínka. Od této chvíle jsou opět oba vodiče SDA a SCL v logické úrovni H a systém je tím pádem připraven na START podmínku, která opět zahájí komunikaci. Obr. 3.9 ukazuje průběh komunikace dvěma mikrokontroléry PIC.



Obr. 3.9 I²C měření

3.3. *Napětové přizpůsobení komunikujících zařízení*

Problém může nastat, je-li několik zařízení napájeno rozdílným napětím V_{DD} . Například je-li zařízení master napájeno napětím +3,3 V (např. FPGA obvody nebo některé mikrokontroléry) a slave zařízení je typu TTL, vlivem neslučitelnosti napětových úrovní není možná komunikace. Je tedy potřeba provést **konverzi napětí** tak, aby byla komunikace možná a to obousměrně. Jeden ze způsobů je použití obvodu SN74LVC2T45. Vnitřní zapojení a pouzdro s vývody je na obr. 1. Datový proud probíhá mezi příslušnými piny PORTU A ($A_1 - A_2$) a B ($B_1 - B_2$), přičemž směr toku dat se přepíná vstupem DIR. Je-li vstup DIR v úrovni H, pak je směr datového přenosu od PORTU A do PORTU B, v úrovni L je směr obrácený. V_{CCA} slouží jako napájecí vstup PORTU A, V_{CCB} plní stejnou funkci pro PORT B. Je-li V_{CCA} například +5 V a V_{CCB} +3,3 V, pak budeme-li uvažovat směr přenosu od A do B, bude při buzení obdélníkovým signálem (budíme např. vstup A_1) s amplitudou +5 V na výstupu B_1 jeho amplituda v úrovni H pouze +3,3 V. Při přepnutí směru z B do A bude situace obrácená. Budeme-li tedy budit např. vstup B_1 obdélníkovým signálem o amplitudě +3,3 V, bude na výstupu A_1 amplituda signálu v úrovni H +5 V.

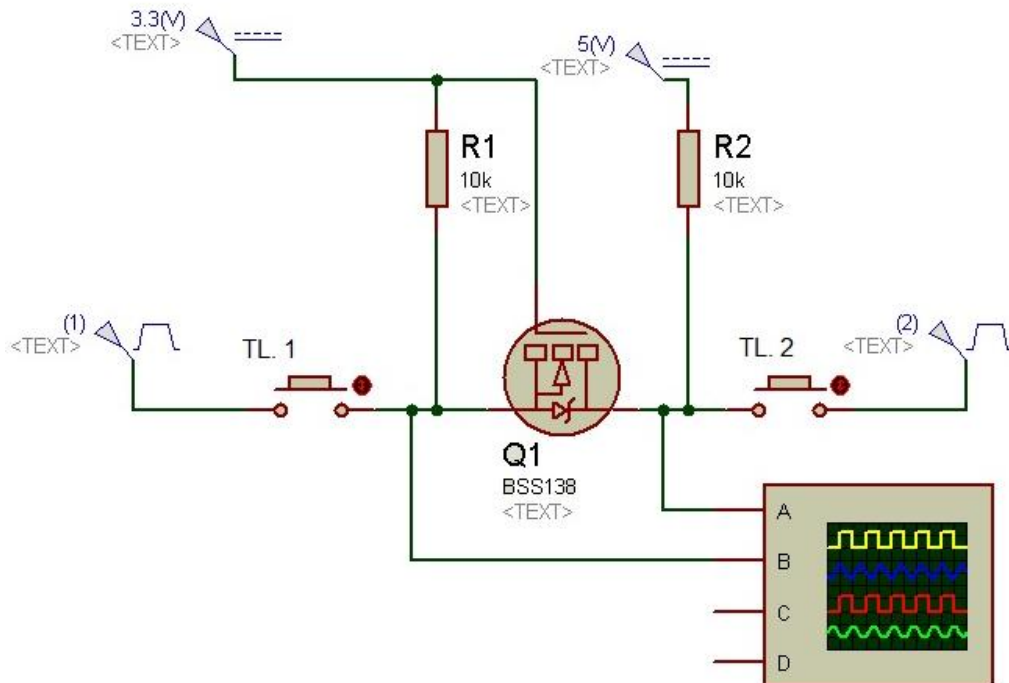


Obr. 3.10 Obvod napětového přizpůsobení

Jen pro úplnost dodejme, že **napětová úroveň L je na obou stranách stejná.**

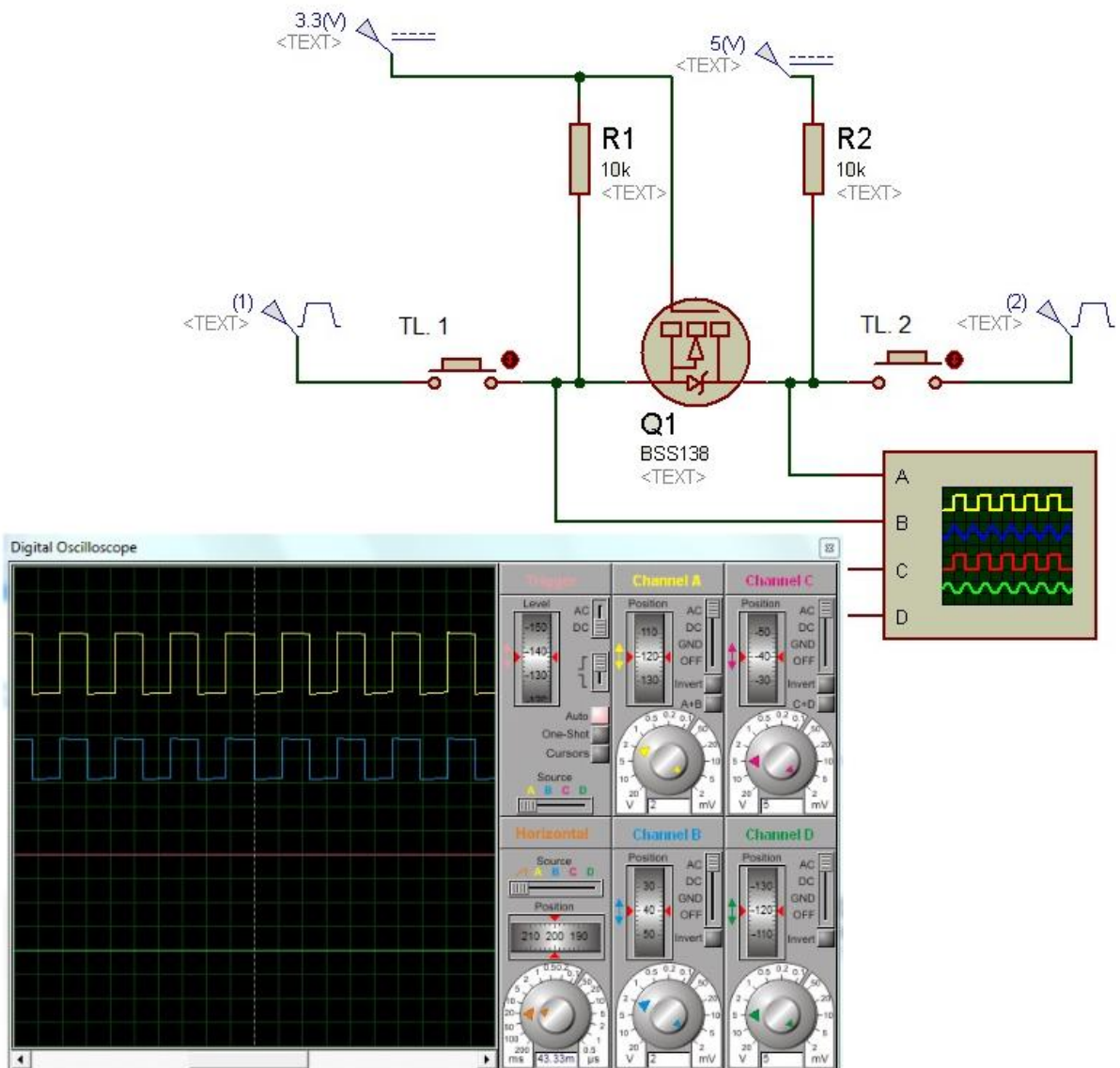
Nevýhoda tohoto řešení je cena obvodu a fakt, že při požadavku na napětovou konverzi více než dvou kanálů ($A_1 \leftrightarrow B_1, A_2 \leftrightarrow B_2$) je potřeba více než jednoho čipu.

Existuje však mnohem levnější řešení, které využívá jednoho tranzistoru MOSFET a dvou rezistorů. Obr. 3.11 ukazuje dané schéma, které bylo zároveň otestováno v simulačním programu Proteus ISIS. Toto řešení je aplikováno v této práci.



Obr. 3.11 Simulace konverze napětí

Konkrétně se jedná o tranzistor BSS138 a dva rezistory o hodnotě 10 k Ω . Na simulovaném schématu je navíc pár tlačítek TL. 1 a TL. 2, které přivádí testovací obdélníkový signál z generátorů funkcí 1 a 2. Na konec je zde osciloskop, na kterém lze provést srovnávací měření a ověřit funkci zapojení. Na obr. 3 je zobrazen výsledek simulace. Při stisknutí tlačítka TL. 1 se z funkčního generátoru 1 přivede obdélníkový signál o amplitudě +3,3 V na kanál B osciloskopu a zároveň přes tranzistor na kanál A osciloskopu. Na obr. 3 je jasně vidět průběh signálu a jeho amplituda, která je na kanálu A větší než na kanálu B. Pokud rozepneme TL. 1 a sepneme TL. 2, pak přivedeme signál z generátoru 2 o maximální hodnotě +5 V a opět uvidíme na osciloskopu obdélníkový průběh a opět vyšší amplitudu na kanálu A oproti kanálu B.



Obr. 3.12 Měření

Princip konverze napětí je velice jednoduchý. Je-li například na levé straně datové linky, tj. na funkčním generátoru 1 logická úroveň H, pak je napětí této úrovně +3,3 V. Toto napětí je tedy nejen na vývodu S (source), ale je i na elektrodě G (gate) a tím pádem je rozdíl napětí mezi G a S skoro rovno nule (vlivem kapacit přechodů je zde menší napěťový offset). Tranzistor je tedy uzavřen a chová se jako bariéra, která odděluje úroveň H (na levé straně +3,3 V) a na pravé straně +5 V. Při změně z úrovně H na L na levé straně datového vodiče už nebude napětí mezi G a S téměř nulové, ale zvýší se na hodnotu zhruba 3,3 V. Tranzistor se otevře a dojde ke stažení úrovně datové linky na nulu také na druhé straně.

Budeme-li postupovat z druhé strany (strana +5 V), při nastavení linky na úroveň L dojde díky diodě mezi S a D (drain) opět ke zvětšení potenciálu U_{GS} a tranzistor se otevře. Tím pádem se objeví log. 0 také na levé straně datové linky.

3.4. Zhodnocení parametrů a výběr vhodného typu sběrnice

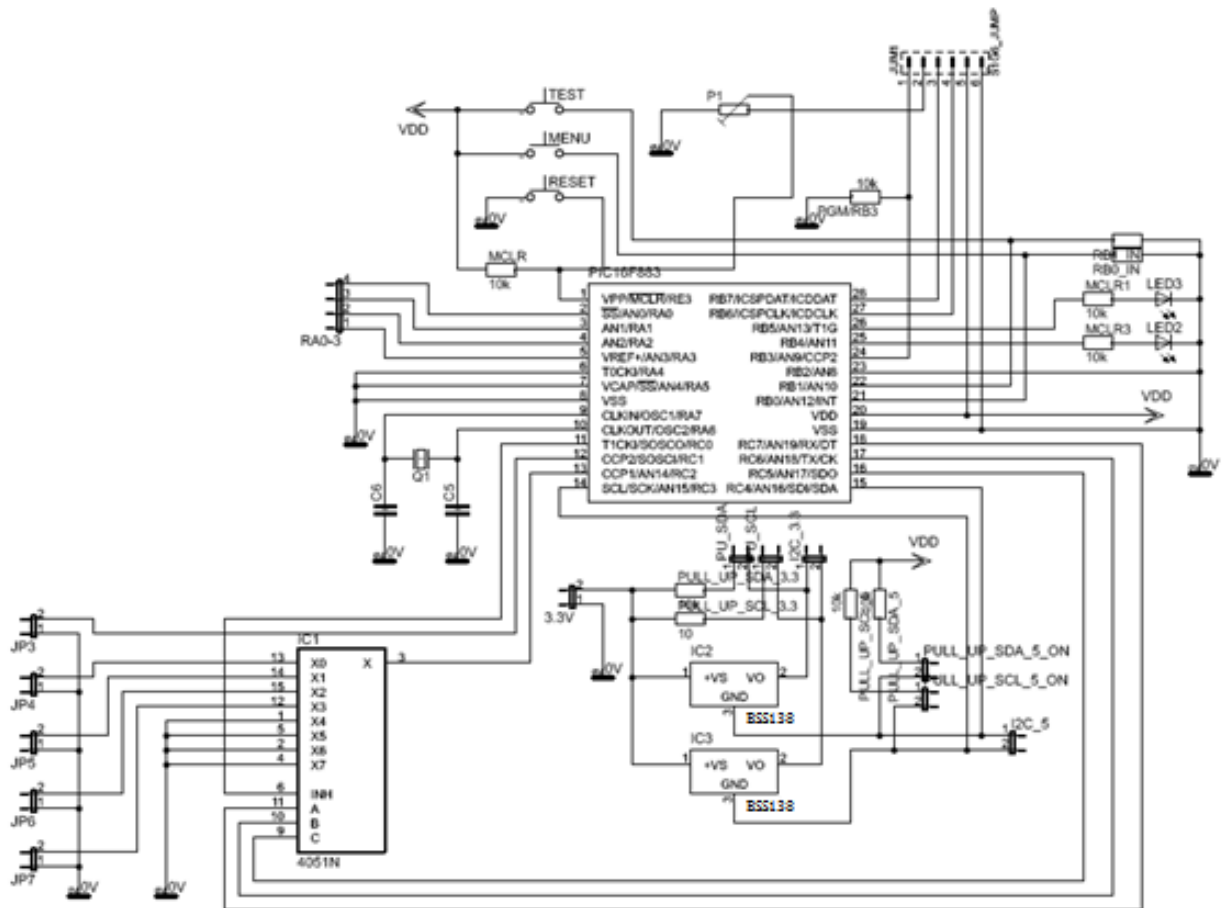
Proveďme nyní srovnání všech popsaných druhů datové komunikace. Následující tabulka shrnuje všechny pozitivní i negativní vlastnosti sběrnic One – Wire, SPI a I²C.

Sběrnice					
One - Wire		SPI		I ² C	
+	-	+	-	+	-
1 datový vodič úspora místa	složitě softwarové nastavení	vysoká provozní rychlost (až 10 MHz)	minimálně 3(+1) vodiče s dalším zařízením roste počet lineárně	relativně dobrá provozní rychlost (100 až 400 kHz)	počet zařízení omezen součtem jejich kapacity
možnost napájení přes datovou linku	nízký počet zařízení podporujících danou technologii	Relativně jednoduchá softwarová obslužnost	menší počet podporovaných zařízení	pouze 2 vodiče pro komunikaci	
			1 obvod master	velký počet podporovaných zařízení	
				více obvodů master	
				jednoduchá softwarová obslužnost	

Tab. 3.1 Porovnání typů sběrnic

Z tabulky je patrné, že pro tento projekt je vhodná implementace systému I²C.

4. Realizace systému



Obr. 4.1 Schéma řídicí části

4.1. Popis hardwaru

Srdcem stabilizačního systému je mikrokontrolér PIC16F883. Jedná se o obvod od firmy Microchip založený na tzv. Harvardské architektuře, který je vybaven řadou periferních systémů.

Mezi ně patří například A/D převodníky, komparátory, obvody časovačů, moduly pro generování PWM signálu a další. Mezi další vlastnosti můžeme zmínit interní generátor hodinového signálu s možností výběru z přednastavených frekvencí. Obvod je také možné taktovat připojením externího zdroje hodinového signálu. Sem patří

například RC oscilátor (tato varianta je levná, ale nehodí se pro aplikace, kde je potřeba pracovat s precizním zdrojem hodin), krystalový oscilátor (bývá zpravidla tvořen krystalem a dvěma kondenzátory), můžeme také signál přivést z jiného mikrokontroléru a nebo z obvodů, které jsou speciálně určeny pro tento úkol. Na plošném spoji je realizována varianta s krystalovým oscilátorem. Vstupem do řídicí jednotky jsou tři tlačítka: **menu** pro výběr typu regulátoru a jeho nastavení (pro budoucí vylepšení, v současné době neaktivní), **test** (testování kormidel, při stisku se všechny řídicí plochy pohnou od jedné krajní polohy do druhé) a tlačítko **reset**, které slouží při zamrznutí systému během testování. Dodejme, že resetovací tlačítko je zapojeno na pin MCLR-, zatím co tlačítko menu a test jsou zapojeny na PORTB (RB0 a RB1). Kromě těchto digitálních vstupů je zde také možnost přivést analogový signál a to na PORTA (RA0-RA3) pro testování polohových senzorů.

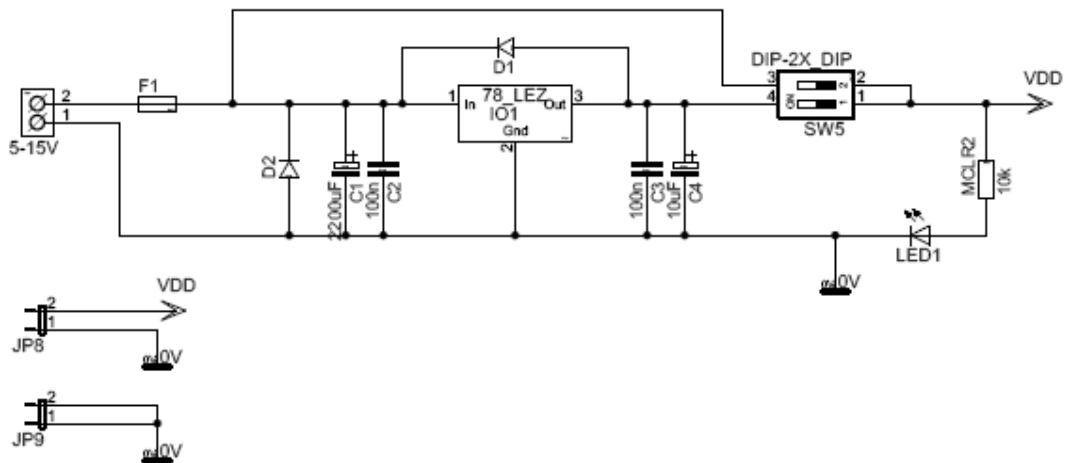
PIC16F883 základní elektrické parametry:	
pracovní teplota	40 až 125 °C
napájecí napětí	3,3 až 5,5 V
max. F_{osc} (s krystalem)	20 MHz
max. proud jedním pinem	25 mA
max. napětí na MCLR	0 až 12,5 V

Tab. 4.1 Základní parametry mikropočítače

Dalším prvkem je integrovaný obvod 4051N. Jedná se o multiplexor, který umožňuje přivést jeden ze dvou PWM signálů na jednotlivé ovládací prvky (serva řídicích ploch letounu). Druhý PWM signál slouží k ovládání pohonné jednotky stroje. Oba PWM signály jsou generovány interním modulem CCP1 a CCP2 na portu RC (viz. katalogový list výrobce [12]).

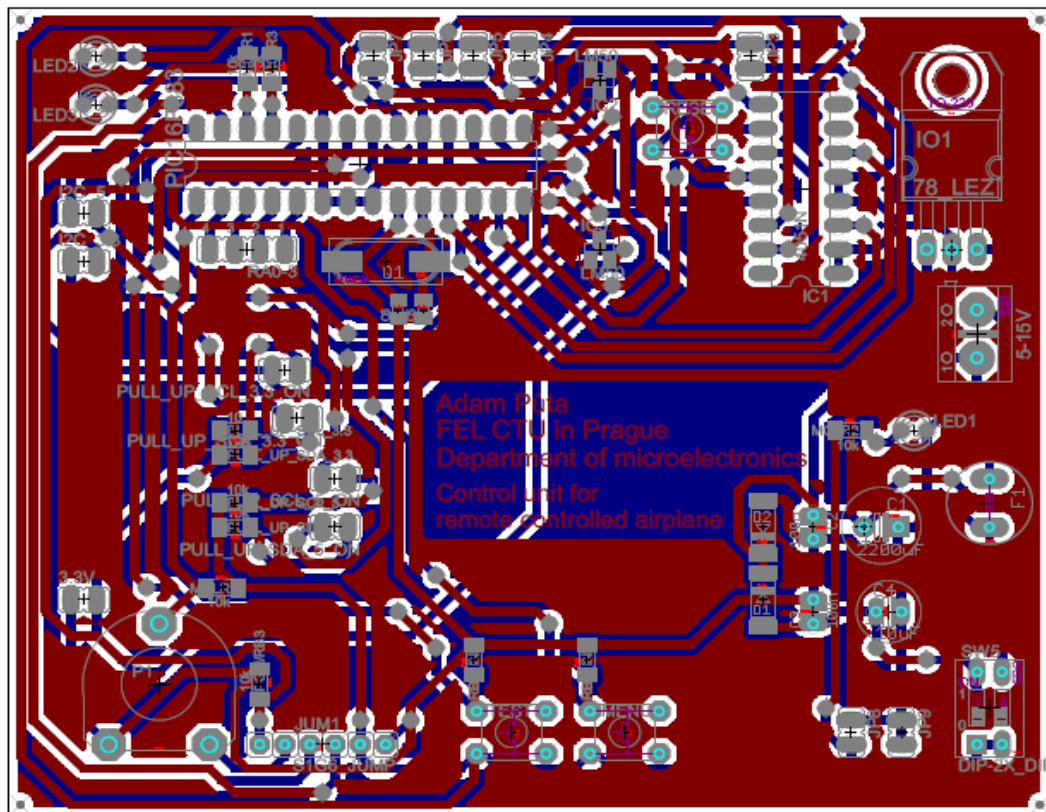
Tranzistory MOSFET BSS138, jak bylo řečeno v odstavci 3.3. *Napětové přizpůsobení komunikujících zařízení*, slouží k napětové konverzi na sběrnici I²C v případě, že je sběrnice provozována v režimu +3,3 V. Sběrnice je na obou stranách připojena k příslušnému V_{DD} pomocí pull – up rezistorů o hodnotě 10 k Ω .

Poslední z integrovaných obvodů je stabilizátor napětí 7805, který slouží jako zdroj v případě, kdy není k dispozici stabilní zdroj +5 V pro řídicí elektroniku. V případě, že je k dispozici zdroj +5 V, vybereme přepínačem (ve schématu SW5) toto napětí jako hlavní zdroj.



Obr. 4.2 Obvod vstupního napájení

Na vstupu obvodu je vratná proudová pojistka, která chrání elektroniku proti překročení maximálního povoleného proudu. Z důvodu vyhlazení a odfiltrování napájecího napětí jsou v obvodu elektrolytické a keramické kondenzátory. Hodnoty jsou dané katalogovým listem výrobce stabilizátoru 7805. Dioda D1 chrání stabilizátor proti náhlému poklesu napětí na vstupní straně. LED1 slouží jako indikace stavu napájení.



Obr. 4.3 Deska plošného spoje

Potenciometr P1 má funkci napěťového děliče. Jeho úkol bude vysvětlen v následující podkapitole. Deska byla vytvořena v programu EAGLE.

4.2. Software

Řídicí software byl napsán ve vývojovém prostředí *mikroC*. Toto prostředí nabízí širokou nabídku knihoven. Od práce se sběrnicemi, přes nastavování A/D převodníků, až po PWM nastavení.

Po naprogramování kódu převede software tento kód na soubor s koncovkou hex (hexadecimální soubor) a ten je poté nahrán do softwaru programátoru, který fyzicky zapíše program do mikrokontroléru. Jako programátor byl zvolen systém presto od firmy ASIX.

K nahrání programu je potřeba připojit piny VPP, DATA, CLOCK, VDD a GND. Pin VPP se připojuje k mikrokontroléru na pin MCLR-, DATA na pin PGD (zpravidla RB7) a CLOCK na PGC (zpravidla RB6). U pinu VPP je ale problém s velikostí dodávaného napětí během tzv. HVP (High Voltage Programming), protože různé typy mikropočítačů mají různé maximální povolené hodnoty.

Příklad: pin VPP dodává +13 V, ale náš typ mikrokontroléru (PIC16F883) má povoleno jen 12. Příručka radí použít klasický napěťový dělič.

$$P = \frac{R_2}{R_2 + R_1} \quad (2.4)$$



Obr. 4.4 Programátor Presto [6]

Výhodnější je však použití potenciometru. Ten slouží jako proměnný napěťový dělič a my tím pádem nemusíme řešit maximální hodnotu na pinu MCLR-, protože si nastavíme dělič tak, abychom dostaly žádanou maximální přípustnou hodnotu.

4.2.1. Algoritmus řízení

Algoritmus řízení obstarává data ze sensorické jednotky a přijímače dálkového ovládání a z těchto informací vygenerovat takové hodnoty akčních veličin, aby došlo ke stabilizaci letu v co možná nejkratší časové době. Ve světě dálkově ovládaných modelů existují dva druhy stabilizace:

- a) 2D stabilizace, která má za úkol „pamatovat si“ polohu horizontu a v případě, že si jí pilot vyžádá, vyrovnat letadlo tak, aby se z téměř libovolné polohy dostalo samo do horizontálního letu.
- b) 3D stabilizace, která má za úkol porovnávat řídicí zásahy pilota (to znamená signál z dálkového ovládání) a údaje ze sensorického subsystému (gyroskopy, akcelerometry) a pokud se skutečný pohyb letadla neshoduje s povely ze země, to znamená došlo vlivem povětrnostních podmínek k nechtěné změně letu, provést kompenzaci tak aby se průběh letu shodoval co nejvíce s průběhem řídicího signálu z ovladače.

5. Závěr

Tento projekt měl za úkol dokázat, že budoucnost může počítat s bezpilotními prostředky, a to s různorodými zdroji energie pro let, jako jsou v tomto případě solární články (z úvodu víme, že primární zdroj energie je akumulátorový systém, který je panely dobíjen). Dále měla tato práce vytvořit podmínky pro spolupráci s různými obory, jako jsou: řídicí technika, sensorová technika, komunikační technika a energetika. Moje část se týkala návrhu a realizace řídicího subsystému pro prostorovou stabilizaci. Protože moji kolegové, kteří měli na starosti dodávat informace do mého zařízení, nepředložili dosud žádné výsledky, bylo nutno testovat stabilizační jednotku za použitím mikropočítače, dvou modelářských servopohonů a potenciometru jako senzoru, který simuloval vstupní parametry, jako je například náklon letounu a podobně.

Testování tedy probíhalo formou improvizovaného systému (vliv rušení prostředí, nedokonalost kontaktního pole atd.), a tím pádem nebylo možné jednotku řízení prohlásit za plně kompatibilní s jinými druhy systémů k tomu určených. I tak ale navržené algoritmy řízení vykazují výsledky například v podobě rozsvícení příslušné LED, nebo pohybu servomotoru na žádanou stranu.

6. Příloha A

Následující část kódu ukazuje funkci čtení ze sensorického subsystému pomocí I²C:

```
void sens(){  
    I2C1_Init(100000);  
    I2C1_Start();        // issue I2C start signal  
    I2C1_Wr(0b11000000); // send byte via I2C bus (device address + W)  
    I2C1_Wr(0b00000100); // data write to I2C bus  
    I2C1_Stop();        // I2C stop signal  
    Delay_ms(20);  
    I2C1_Init(100000);  
    I2C1_Start();        // issue I2C start signal
```

```

I2C1_Wr(0b11000001); // send byte via I2C (device address + R)
reg_data = I2C1_Rd(0); // Read the data (NO acknowledge)
Delay_ms(20);
I2C1_Stop(); // issue I2C stop signal
}

```

Generování PWM pro jeden akční člen (křídélko):

```

void P_reg(){
    PORTC = 0b11100000; //switch multiplexer for left_ailerons
    correction = ground_data - reg_data; // create erro
    P_con = kp*correction; // P controller
    U_a = U_a - P_con;
    PWM1_start(); //start PWM
    PWM1_Set_Duty(Ua); //pwm settings
}

```

Hlavní část programu:

```

void main(){

    Init();
    while(1){
        if(RB1_bit = 1){ //

            loop:
                sens_sensoric_unit(); // call function sens to get data
                Delay_ms(10);

```

```

sens_ground_unit();
P_reg();      // P regulator function
goto loop;
}
}
}

```

Jako náhrada za měřící jednotku byl zvolen mikrokontrolér PIC16F88 v režimu slave:

```

const Address = 0b11000000;      // set I2C device address

unsigned short j;
unsigned short rxbuffer;      //
unsigned short tx_data;      //

void Init(){
    ANSEL = 0;
    TRISA = 1;      // Set PORTA as input
    TRISB = 0xFF;      // Set PORTB as input
    SSPADD = Address;      // Get address (7bit). Lsb is read/write flag
    SSPCON = 0x36;      // Set to I2C slave with 7-bit address
    PIE1.SSPIF = 1;      // enable SSP interrupts
    INTCON = 0xC0;      // enable INTCON.GIE
}

void interrupt(){
    if (PIR1.SSPIF == 1){
        PIR1.SSPIF = 0;
    }
}

```

```

//transmit data to master
if (SSPSTAT.R_W == 1){
    SSPBUF = tx_data;
    SSPCON.CKP = 1;
    j = SSPBUF;
    return;
}
if (SSPSTAT.BF == 0){
    j = SSPBUF;
    return;
}

//recieve data from master
if (SSPSTAT.D_A == 1){
    rxbuffer = SSPBUF;
    j = SSPBUF;
    return;
}
}
j = SSPBUF;
}

void main(){
    Init();
    while(1){

```



```
tx_data = 5;  
Delay_ms(250);
```

```
}
```

```
}
```

7. Příloha B

Katalogové listy

SN74LVC2T45

<http://www.ti.com/lit/ds/symlink/sn74lvc2t45.pdf>

multiplexor 4051

<http://www.ges.cz/sheets/c/cd4051.pdf>

PIC16F88

<http://ww1.microchip.com/downloads/en/DeviceDoc/30487D.pdf>

PIC16F883

<http://ww1.microchip.com/downloads/en/DeviceDoc/41291B.pdf>

EAGLE library

<http://www.cadsoftusa.com/download-eagle/>

8. Seznam použité literatury

[1a] Šebek M. ČVUT FEL katedra řízení

<https://moodle.dce.fel.cvut.cz/course/view.php?id=68>

[1b] Šebek M. ČVUT FEL katedra řízení

<https://moodle.dce.fel.cvut.cz/course/view.php?id=68>

[1c] Šebek M. ČVUT FEL

<https://moodle.dce.fel.cvut.cz/course/view.php?id=68>

[2] Malý M. HW

<http://www.hw.cz/navrh-obvodu/rozhrani/sbernice-1-wiretm.html>

[3] Teplý T. MIK FEL ČVUT katedra mikroelektroniky One-Wire

<http://datasheets.maximintegrated.com/en/ds/DS18S20.pdf>

9. Obrázky

2.1	Testovací letoun.....	12
2.2	Vliv větru na trajektorii letu.....	13
2.3	Vliv momentu.....	14
2.4	Blokové schéma řízené soustavy.....	15
2.5	Závislost řídicího zásahu na odchylce „On-Off“ regulátoru (sign(e)).....	15
2.6	Hystereze regulátoru „On-Off“	16
2.7	Modře vyznačené pásmo necitlivosti regulátoru.....	16
2.8	Vliv konstanty k_p na regulovanou veličinu [1a].....	17
2.9	Vliv konstanty k_i na regulovanou veličinu [1b].....	18
2.10	Vliv konstanty k_i na regulovanou veličinu [1b].....	19
3.1	Průběh komunikace One-Wire [2].....	21
3.2	Zapojení One-Wire sběrnice.....	22
3.3	Napájení přes One-Wire [3].....	22
3.4	SPI.....	23
3.5	Daisy Chain.....	24
3.6	Datový tok v SPI.....	25
3.7	Měření SPI.....	25
3.8	I ² C zapojení.....	26
3.9	I ² C měření.....	27
3.10	Obvod napěťového přizpůsobení.....	28
3.11	Simulace konverze napětí.....	29
3.12	Měření.....	30
4.1	Schéma řídicí části.....	32
4.2	Obvod vstupního napájení.....	34

4.3	Deska plošného spoje.....	34
4.4	Programátor Presto [6].....	35

10. Tabulky

3.1	Porovnání typů sběrnic.....	31
4.1	Základní parametry mikropočítače.....	33

11. Použité zkratky

F_{osc}	frekvence oscilátoru mikropočítače
F_{CLK}	frekvence hodinového signálu
MISO	vstupní pin obvodu master (výstup obvodu slave)
MOSI	výstupní pin obvodu master (vstup obvodu slave)
CLK	hodinový signál SPI
SDA	datový vodič I ² C
SCL	hodinový signál I ² C
V_{DD}	kladné napájecí napětí
V_{EE}	záporné napájecí napětí (zem)
V_{SS}	záporné napájecí napětí (zem)
V_{CC}	kladné napájecí napětí
GND	zem

12. Obsah CD

CD obsahuje PDF soubor této bakalářské práce, seznam použitých součástí a soubory sch. a brd. vygenerované programem EAGLE