**STŘEDNÍ PRŮMYSLOVÁ ŠKOLA STROJNÍ A ELEKTROTECHNICKÁ
A VYŠŠÍ ODBORNÁ ŠKOLA, LIBEREC 1, Masarykova 3**

Masarykova 3, 460 84 Liberec 1, tel. 485 100 113, fax 485 100 063, e-mail sekretariat@pslib.cz, http://www.pslib.cz

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Trikoptéra

Filip Richter
Filip Svoboda

Liberec 2011

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor SOČ: 10. Elektrotechnika, elektronika a telekomunkace

Trikoptéra
Tricopter

Autoři:	Filip Richter Filip Svoboda
Škola:	Střední průmyslová škola strojní a elektrotechnická a Vyšší odborná škola, Liberec 1, Masarykova 3, příspěvková organizace
Konzultant:	Ing. Marek Pospíchal vedoucí práce Bc. Tomáš Kazda, DiS.

Liberec 2011

Prohlašujeme, že jsme svou práci vypracovali samostatně, použili jsme pouze podklady (literaturu, SW atd.) citované v práci a uvedené v přiloženém seznamu a postup při zpracování práce je v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Liberci dne 26. 3. 2011 ....

Filip Richter Filip Svoboda

Liberec 2011

ANOTACE

Tato práce se zabývá projektem „Tricopter“, který má za úkol navrhnout a zrealizovat jednotku pro řízení třívrtulového letounu. Dokumentace je zaměřena na postupy při navrhování jednotky a na praktické zkušenosti, které se během vývoje získaly. Dílčí kroky návrhu jsou patřičně okomentovány a zdůvodněny. Projekt také popisuje rozdíly mezi teorií a praxí.

SUMMARY

This documentation deals with the project called „Tricopter“. In this project we are about to design and realize a control unit for the triple-bladed hovercraft. The documentation is focused on the procedures during designing and the practical experiences gained during developing. Each and every step is appropriately annotated. The project also describes the differences between theory and practice.

PODĚKOVÁNÍ

Nejprve bychom chtěli poděkovat panu Ing. Jaroslavu Semerádovi, díky kterému se projekt vůbec mohl uskutečnit. Další dík patří panu Marku Pospíchalovi za vedení práce a panu Bc. Tomáši Kazdovi, DiS. za jeho pomoc při návrhu a výběru modelářských součástí.

OBSAH

KLÍČOVÁ SLOVA

Klíčové slovo	*Vyjádření zkratky*	*Vysvětlení*
HW	*Hardware*	*Fyzická část zařízení*
SW	*Software*	*Program (data)*
MCU	*Microcontroller Unit*	*Mikrokontrolér*
LED	*Light Emitting Diode*	*Signální dioda*
AD	*Analog Digital*	*Analogově digitální*
AHRS	*Attitude Heading Reference System*	*Referenční systém letové polohy*
USART	*Universal SynchronousAsynchronous Receiver Transmitter*	*Sériová komunikace*
GPS	*Glabal Position System*	*Globální triangulační systém*
DPS		*Deska plošných spojů*
PWM	*Pulse Width Modulation*	*Pulzně šířková modulace*
SMD	*Surface Mount Device*	*Součástky pro povrchovou montáž*
RC	*Radio Control*	*Radiové ovládání*
PC	*Personal Computer*	*Osobní počítač*
AVR		*Označení rodiny mikročipů*
VCC		*Napájecí napětí*
PID		*Druh regulátoru*
Li-Pol		*Lithium-polymerový akumulátor*
CTC	*Clear Timer on Compare Match*	*Režim časovače AVR*
RTC	*Real time clock*	*Blok reálného času*
BLDC	*Brushless DC motor*	*Bezkartáčový motor*
Watchdog		*Blok hlídající činnost*
Bootloader		*Zavaděč programu*
drift		*unášení*
pin		*vývod proceseru*

1 ÚVOD

Cílem této dlouhodobé maturitní práce je vytvořit řídící jednotku letounu „Tricopter“ a to jak po hardwarové tak i softwarové stránce. Úkolem jednotky je komunikace s pilotem, který letoun ovládá a správa letounu.

V praxi pak pilot může letoun ovládat například pomocí počítače, který komunikuje s jednotkou. Jednotka na základě přijatých informací ovládá samotný letoun. Také má uživateli zprostředkovat důležitá data potřebná pro let a informace o stavu letounu. Například informace o poloze nebo předpokládané době letu.

Zároveň by měla maximálně usnadnit ovládání letounu a to díky automatické stabilizaci případně jiným autonomním režimům. Při využití automatické stabilizace letoun nesmí rotovat a drift by měl být co nejmenší. Tyto požadavky je možné zajistit pomocí senzorů polohy ve 3D prostoru. Jednotka by měla být flexibilní, aby umožňovala jednoduchou modifikaci a rozšíření například o řízení pomocí GPS.

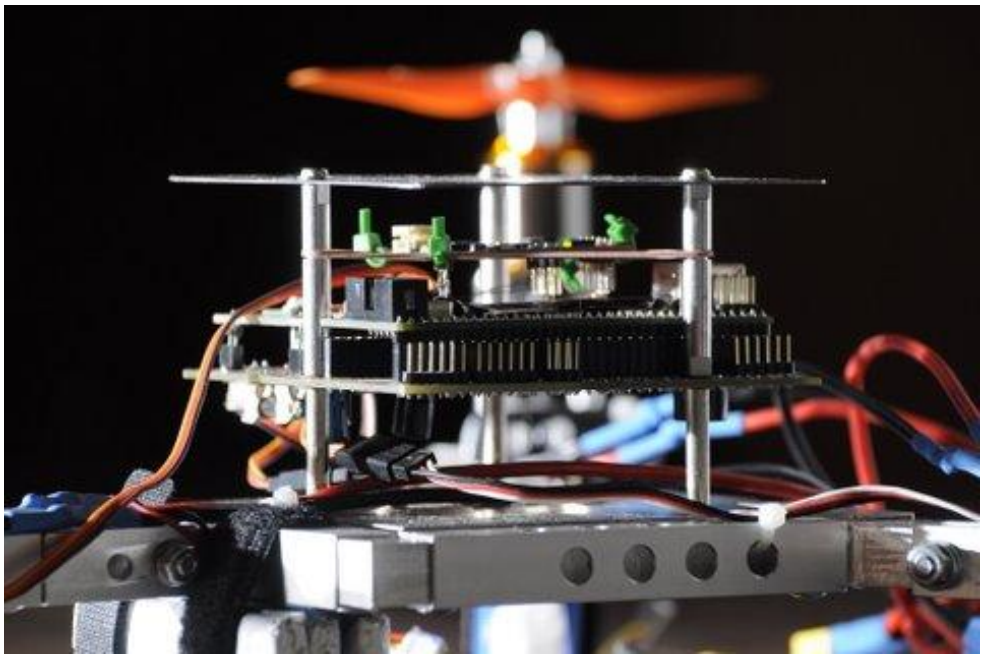
2 SEZNÁMENÍ S MODELEM

2.1 Mechanická konstrukce

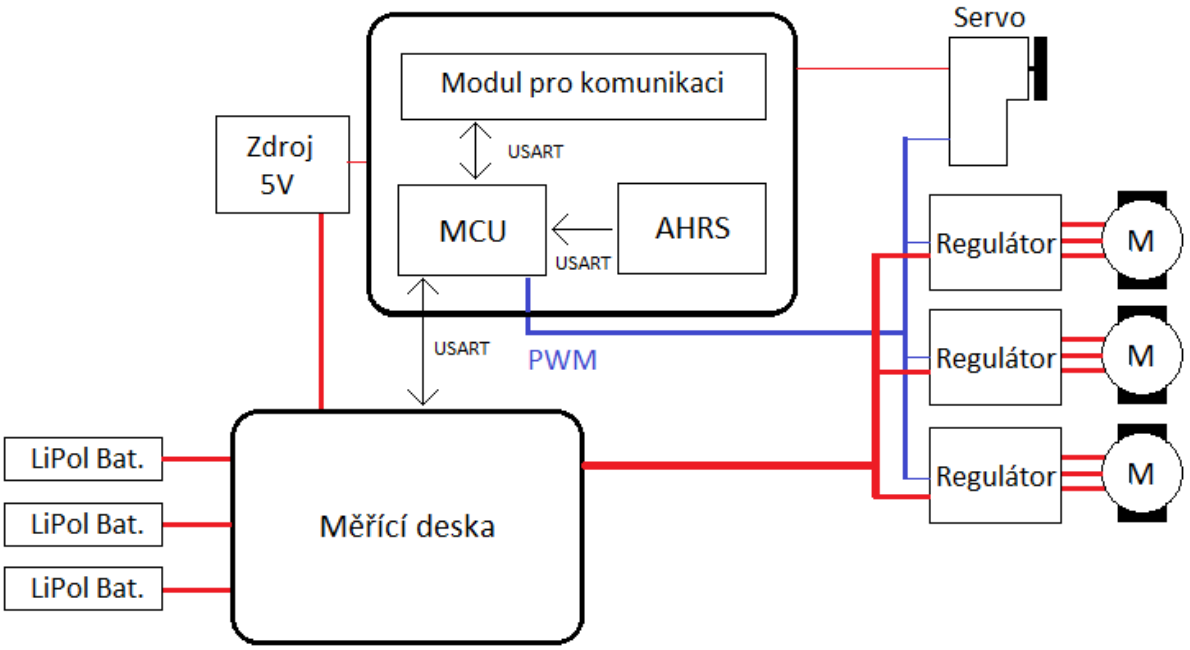
Mechanická konstrukce je vytvořena v rámci Dlouhodobé maturitní práce studentem oboru strojírenství Martinem Štrynclem.

2.2 Elektronika

Řídící elektronika je umístěna ve středu konstrukce na distančních sloupcích, skládá se ze dvou DPS. Deska řídící jednotky obsahuje modul procesoru, modul komunikace a modul senzorů. Druhá je měřící deska, ta obsahuje vše potřebné k měření napětí, proudů a teploty.



Obrázek 1: Osazená řídící jednotka



Obrázek 2: Blokové schéma

Obrázek 2 znázorňuje zjednodušené blokové schéma elektroniky letounu. Jednotlivými částmi se budeme zabývat níže.

2.3 Řízení modelu

V případě pohybu ovládací páky je vlastně udán úhel, na který se má model dostat. Ten je pak použit, jako žádaná hodnota, pro regulaci.

Ovládání modelu probíhá řízením výkonu jednotlivých motorů, čímž začne model „padat“ směrem k motoru, kterému byl výkon snížen. Tímto řízením lze dosáhnout stability, kdy v okamžiku pádu jedním směrem je proveden „odpovídající“ zásah. V průběhu prací se ukázalo, že nastavení „odpovídajícího“ zásahu je jedna z nejsložitějších úkolů v projektu vůbec.

Další částí je řízení modelářského servomotoru, který ovládá náklon zadního motoru. Úpravou náklonu motor vytváří točivý moment působící proti momentu, který vzniká souhlasným otáčením všech tří motoru. V případě, že by neprobíhala neustálá korekce náklonu, model by se začal otáčet. Točivý moment vznikající souhlasným otáčením totiž není konstantní, ale mění se v závislosti na otáčkách jednotlivých motorů. A jelikož je výkon motoru neustále aktualizován, je třeba korigovat i úhel náklonu motoru. Dalším důvodem pro změnu je povel pro zahájení otáčení z ovládacího systému.

2.4 Napájení

Jednou z dalších důležitých částí modelu je zdroj energie. Stabilizace modelu ve vzduchu je energeticky náročná operace. A tak i přes volbu lehkých a výkonných akumulátorů Li-Pol o celkové kapacitě 73,6 Wh je doba volného visu odhadována přibližně na 12 minut.

2.5 Ovládání modelu

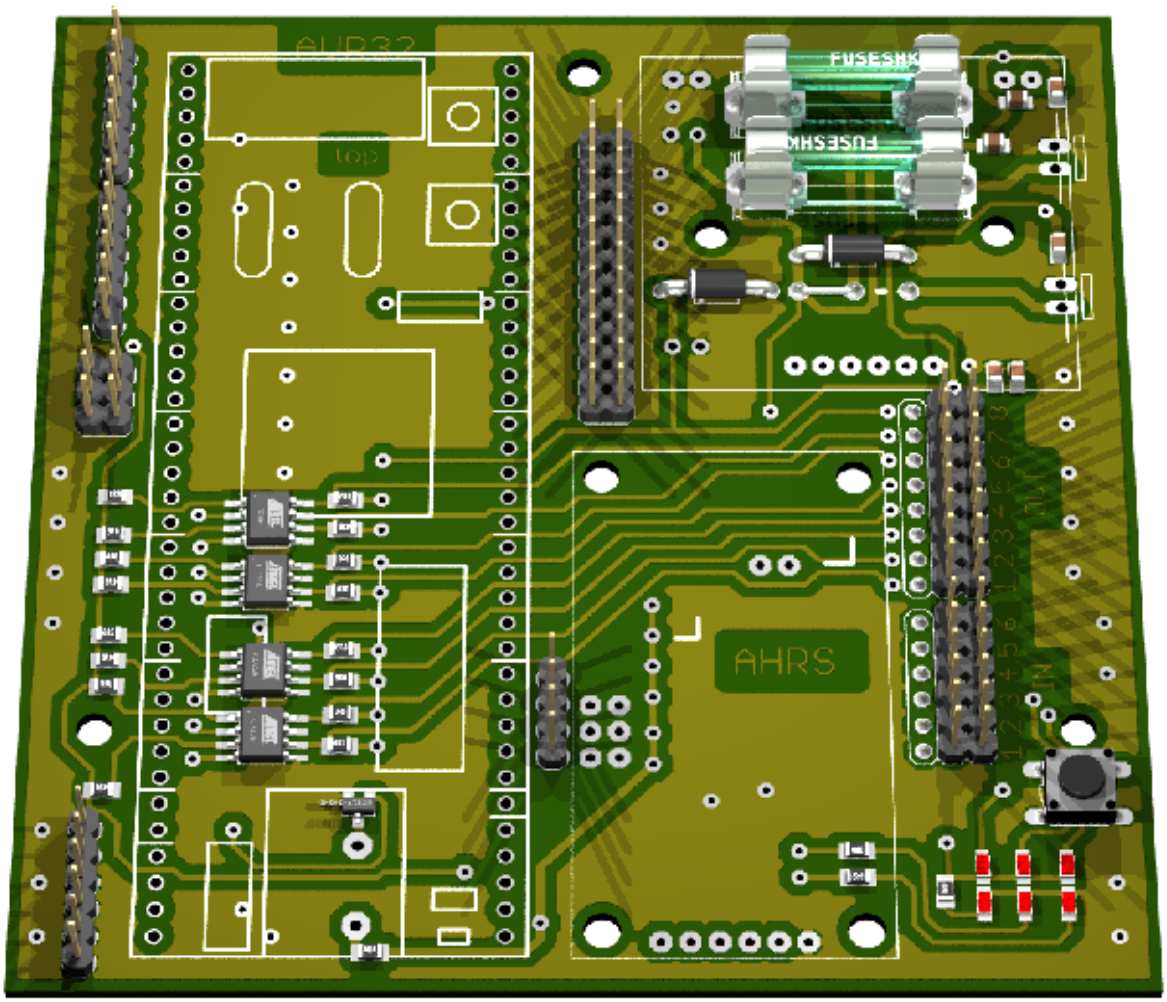
Modely tohoto typu mají obvykle několik režimů ovládání. Podle zvoleného typu si řídící jednotka načte konstanty pro řízení a poté tomu odpovídá chování modelu.

Jeden ze standartních režimů je volný pohyb, kdy je kladen důraz na přesné a citlivé řízení, model se v tomto režimu chová pomaleji. Používá se obvykle pro pohyb ve vnitřních prostorech, transport nákladu nebo focení leteckých fotografii osazeným fotoaparátem.

Dalším režimem bývá akrobacie, kdy je důraz kladen na rychlé, ačkoliv stále přesné, řízení umožňující pak u vyspělých modelů různé akrobatické prvky. Tento režim však není vhodný například pro pohyb ve vnitřních prostorech.

3 ŘÍDÍCÍ JEDNOTKA LETOUNU

Z důvodu usnadnění vývoje nového zařízení a možnosti snadné modifikace hardwaru jsme navrhli řídící jednotku modulárně. Základem jednotky je tedy DPS (Obrázek 2), která umožňuje propojení dílčích částí zařízení (Obrázek 3). Řídící jednotku dále tvoří AL-UC3AEB modul (Dipl. Ing. Alexander Dick) s řídícím MCU, AHRS modul (Sparkfun) a modul pro komunikaci (Speziel Electronic). Všechny moduly lze jednoduše vyjmout a tak se dají použít i do dalších verzí hardwaru.



Obrázek 3: DPS Řídící jednotka

3.1 MCU

Jako hlavní procesor byl zvolen 32b mikrokontrolér od firmy Atmel. Konkrétně se jedná o AT32UC3A1512 s jádrem AVR32. Tento MCU byl vybrán pro své výpočetní rychlosti a periferie, které jsou nutné pro naši aplikaci. Další předností je, že se vyrábí i

jako vývojový modul, který obsahuje bootloader, tudíž není nutný programátor (Dipl. Ing. Alexander Dick). Použitím tohoto modulu se také výrazně zjednodušil návrh řídící jednotky a vývoj softwaru.

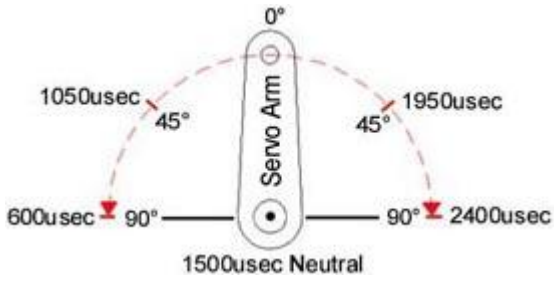
3.2 Periferie jednotky

3.2.1 USART

Řídící MCU disponuje čtyřmi sběrnicemi USART, dva z nich jsou využity interně a probíhá na nich komunikace mezi MCU, AHRS modulem a komunikačním modulem. Další dva slouží ke komunikaci s HW, který je mimo DPS jednotky. Je možné je tedy využít pro připojení měřící desky, GPS modulu atd.

3.2.2 PWM

Díky sedmi kanálovému 16b PWM řadiči doplněného o jeden kanál čítače/časovače, je jednotka schopna ovládat až 8 regulátorů či servomechanizmů. Frekvence PWM je nastavena na 50Hz a střídou každého signálu se volí rychlost motoru, nebo výchylka servomechanismu. Obecně platí, že zařízení pracující s tímto modelářským standardem se ovládají střídou od 600µs do 2400µs (Obrázek 3), (Ježerský). Počáteční velikost střídy signálu se u regulátoru a servomechanismu liší, jelikož je 0° výchylka servomechanismu reprezentována poloviční střídou v daném rozsahu. Počáteční hodnoty pro servomechanismy jsou tudíž nastaveny na 1500µs (nebo dle potřeby počátečního natočení) a pro regulátory na 1100µs (minimum).

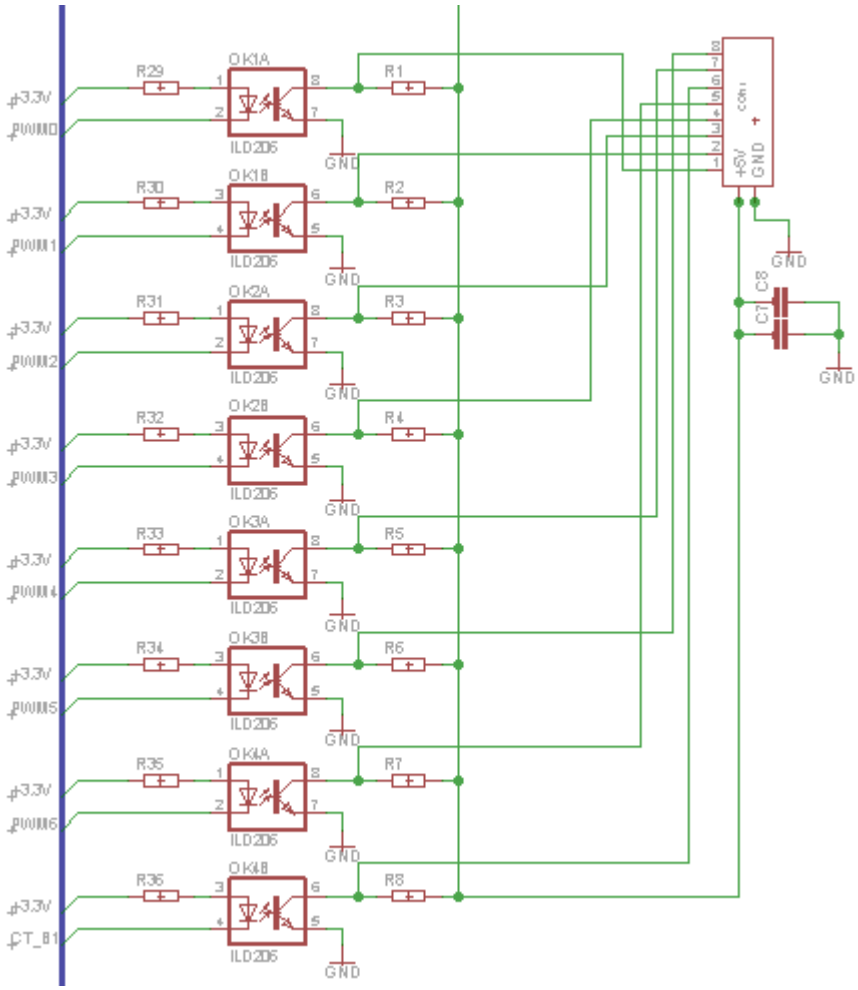


Obrázek 4: Výchylka serva

Z důvodu ochrany řídící logiky a nutnosti amplitudy signálu o velikosti 5V, jsou všechny tyto výstupy galvanicky odděleny pomocí optočlenů. Jelikož piny, které tyto optočleny ovládají, mají maximální zatížitelnost proti VCC -4mA (některé -8mA), musely být i rezistory k optočlenům navrženy s ohledem na tyto proudy.

$$R_{min} = \frac{U_{OUT} - U_{LED}}{I_{max}} = \frac{3{,}3 - 1{,}1}{0{,}004} = 550\Omega$$

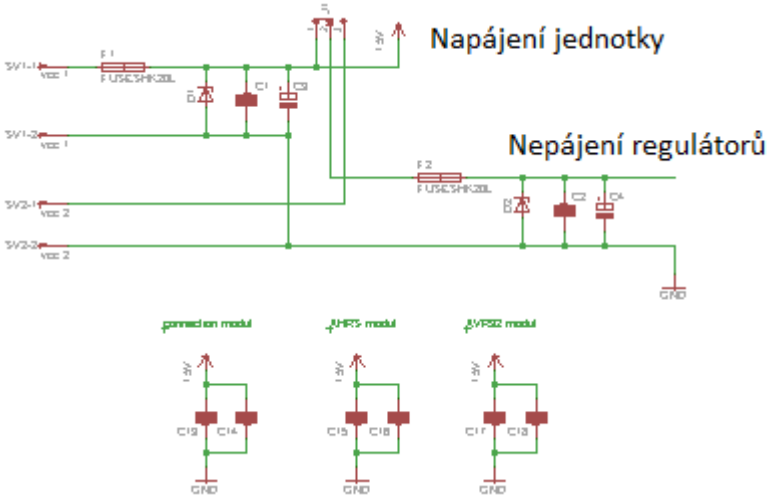
Z předchozího vztahu vyplývá, že minimální odpor rezistoru musí být 550Ω. Námi zvolená hodnota je 680Ω, což vyhovuje této podmínce. Pokud bychom zvolili rezistor s vyšším odporem, pak by proud protékající diodou optočlenu nedokázal dostatečně vybudit LED a tím by tranzistor optočlenu plně nesepnul, to by způsobilo posunutí nuly na výstupu. Optočleny vykazují určité zkreslení hran signálu, které je patrné zejména na vyšších frekvencích, kde může být signál na výstupu zcela utlumen. V našem případě se jedná o frekvenci 50 Hz s proměnnou střídou signálu, na této frekvenci optočleny signál zkreslují jen nepatrně, a jelikož dochází ke zpožděnému sepnutí i rozepnutí, je chyba zanedbatelná a v praxi to znamená minimální posunutí fáze. Schéma galvanického oddělení můžete vidět na obrázku 4.



Obrázek 5: Oddělení výstupů

Abychom předešli případnému rušení napětí od regulátorů frekvencemi, se kterými regulátory pracují, vytvořili jsme dvě napájecí svorky, které je možné paralelně

propojit nebo samostatně napájet a jistit logiku jednotky i signál do regulátorů (nebo servomechanizmů). Je tedy možné použít jeden nebo dva zdroje napětí. Oba napájecí vstupy jsou jištěny proti přepólování zapojením Zenerovy diody a pojistky, v případě přepólování se bude dioda chovat jako zkrat a následný velký proud přepálí pojistku, tím se zabrání poškození elektroniky zapojené na těchto svorkách. K přepálení pojistky by došlo, i pokud by se vyskytl zkrat za napájecími svorkami.



Obrázek 6: Rozdělení napájení

3.2.3 Vstup pro RC

Pro zajištění kompatibility s modelářským rádiem je jednotka vybavena šesti galvanicky oddělenými vstupy, připojenými k externím přerušením mikrokontroléru. Tím pádem je možné letoun ovládat i modelářskou vysílačkou a to šesti kanály.

3.2.4 Rozšiřující slot

Vzhledem k flexibilitě letounu z hlediska využití a vybavení, jsme použili 12 pinů MCU k vytvoření univerzálního konektoru, pro připojení externích zařízení. Z tabulky 1 je patrné, o které piny se jedná a k čemu se dají využít.

číslo pinu	možnosti využití
PA7	SPI0-NPCS[3]/PM-GCLK[0]/USART1-CLK
PA8	EIM-EXTINT[7]/SPI0-NPCS[1]/USART1-RTS
PA9	MACB-WOL/SPI0-NPCS[2]/USART1-CTS
PA11	USB-USB_ID/SPI0-MISO
PA12	USB-USB_VBOF/SPI0-MOSI
PA13	SPI0-SCK
PA14	EBI-NCS[0]/SPI1-NPCS[0]/SSC-TX_FRAME_SYNC
PA15	EBI-ADDR[20]/SPI1-SCK/SSC-TX_CLOCK
PA16	EBI-ADDR[21]/SPI1-MOSI/SSC-TX_DATA
PA17	EBI-ADDR[22]/SPI1-MISO/SSC-RX_DATA
PA18	MACB-WOL/SPI1-NPCS[1]/SSC-RX_CLOCK
PA19	SPI1-NPCS[2]/SSC-RX_FRAME_SYNC

Tabulka 1:Rozšiřující slot

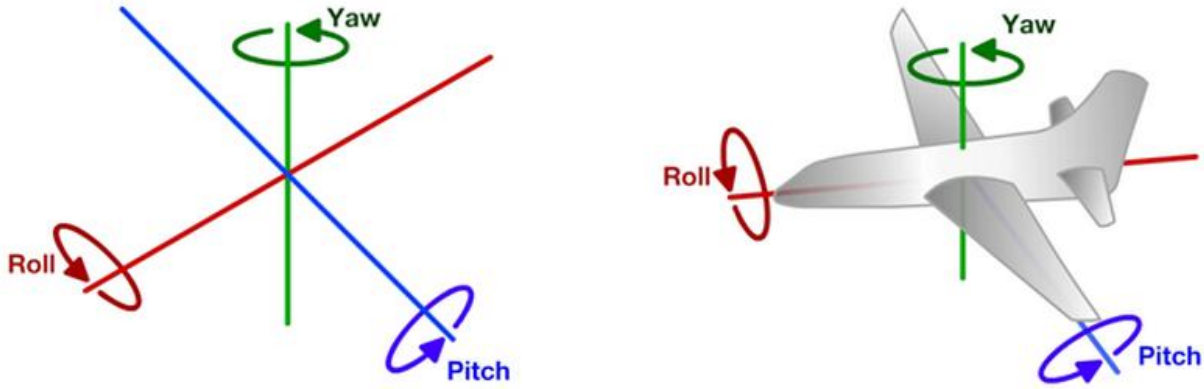
Dostupné jsou také čtyři kanály 10b ADC, které byly také vyvedeny.

3.2.5 Signalizace a ovládání

O stavu jednotky informují dvě trojice SMD LED umístěné v přední části DPS. Jedna trojice LED je vyhrazena pro zobrazení stavu letounu a druhá trojice informuje o připojení jednotky k počítači a přenosu dat. Vzhledem k charakteru aplikace byla jednotka opatřena buzzerem, který zvukově informuje uživatele, bez nutnosti vizuálního kontaktu s jednotkou.

3.3 AHRS modul

Tedy „Attitude snd Heading Reference Systems“ slouží k orientaci letounu v 3D prostoru. Výstupem tohoto systému, je informace o náklonu v osách X, Y a rotace kolem osy Z.



Obrázek 7: Názvy os

rozpoznání pohybu ve třech osách byly u tohoto modulu (Sparkfun) využity:

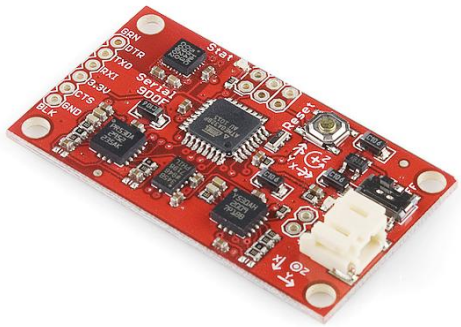
Jednoosé gyro *LY530ALH - 300°/s, s analogovým výstupem*

Dvouosé gyro LPR530ALH - 300°/s, s analogovým výstupem

Tříosý akcelerometr ADXL345 - 13-bit, ±16g, s digitálním výstupem

Tříosý magnetometr HMC5843 s digitálním výstupem

Informace ze senzorů zpracovává MCU od firmy Atmel, ATmega328 a odesílá je po sběrnici USART do řídícího MCU.



Obrázek 8: AHRS modul

3.3.1 Výhody AHRS modulu

Výhodou zmiňovaného modulu je centralizace jednotlivých senzorů potřebných pro výpočet polohy. Tímto řešením se nám usnadnil řídící HW a díky samostatnému MCU, který DPS modulu obsahuje je SW pro řízení letounu logicky decentralizovaný, což zase usnadňuje správu programu a snadné hledání chyb při poruše. Obvod je také doplněn o svůj vlastní stabilizátor napětí, tudíž je celé zařízení částečně chráněno proti externí poruše napájení. Další výhodou je podpora ze strany výrobce, který k modulu dodává i software.

3.3.2 Software AHRS modulu

Pro naši aplikaci jsme vyzkoušeli dva SW, které výrobce umožňuje k modulu stáhnout. První z nich, pouze měřil hodnoty ze senzorů a bez jakéhokoli přepočtu nebo filtrace je cyklicky odesílal frekvencí 50Hz (data aktualizovaná za každé 0,02 s). Tudíž řídící MCU musel obsahovat algoritmy pro přepočet zrychlení a změnu úhlu za čas na informaci o rotaci v dané ose. Praktické testování ukázalo, že tato informace není použitelná pro další zpracování (PID regulaci). Konstrukce letounu při rotaci motorů začne vibrovat, což způsobí vysoké zrychlení ve všech osách, které výrazně ovlivňuje výslednou informaci o náklonu.

Graf 1: Bez filtrace

Graf 1 znázorňuje chybu úhlu natočení letounu v závislosti na rostoucích otáčkách motorů, kdy už při rychlosti otáčení, která není dostatečná pro let, je chyba více jak 20°. Z tohoto důvodu byl v programu AHRS modulu nastaven rozsah z ±2g na ±16g (Obrázek 9). Nastavení rozsahu bylo provedeno v inicializaci akcelerometru, pomocí prvních dvou bitů v registru 0x31 (Tabulka 3).

Table 18. *g* Range Setting

Setting		
D1	D0	*g* Range
0	0	±2 *g*
0	1	±4 *g*
1	0	±8 *g*
1	1	±16 *g*

Tabulka 3: Nastavení rozsahu AHRS

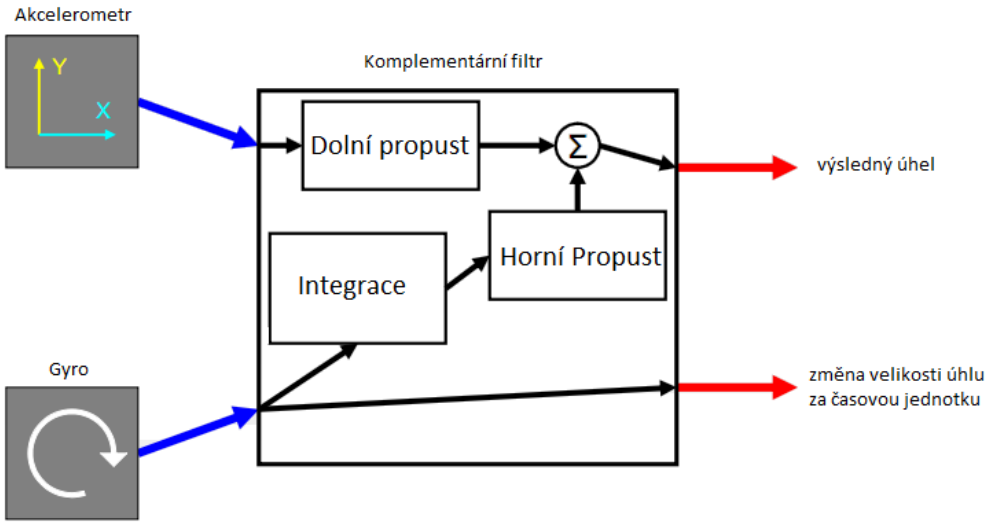
3.3.2.1.1 Register 0x31—DATA_FORMAT (Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
SELF_TEST	SPI	INT_INVERT	0	FULL_RES	Justify	Range	

Tabulka 2: Nastavovací registr

Tato úprava pomohla zmenšit chybu, ale zmenšení chyby nebylo dostatečné. Dalším krokem pro odstranění kmitání bylo filtrování výstupní hodnoty. Nejprve jsme do programu implementovali algoritmus komplementární filtrace.

Princip komplementární filtrace je patrný z obrázku 8 (Colton)a matematického vyjádření.



Obrázek 9:Komplementární filtr

Matematicky lze filtraci provést následovně:

Nejprve je nutné převést hodnoty všech zrychlení na informaci o úhlu. To je možné za předpokladu, že je letoun v klidu vůči zemi, což znamená, že součet všech zrychlení je roven 1 (a tedy i zrychlení k zemi). V případě, že je tato podmínka splněna, úhel mezi vektory zrychlení lze spočítat podle uvedeného vztahu:

$$x_{acc} = \operatorname{atan}(\frac{ax}{az})$$

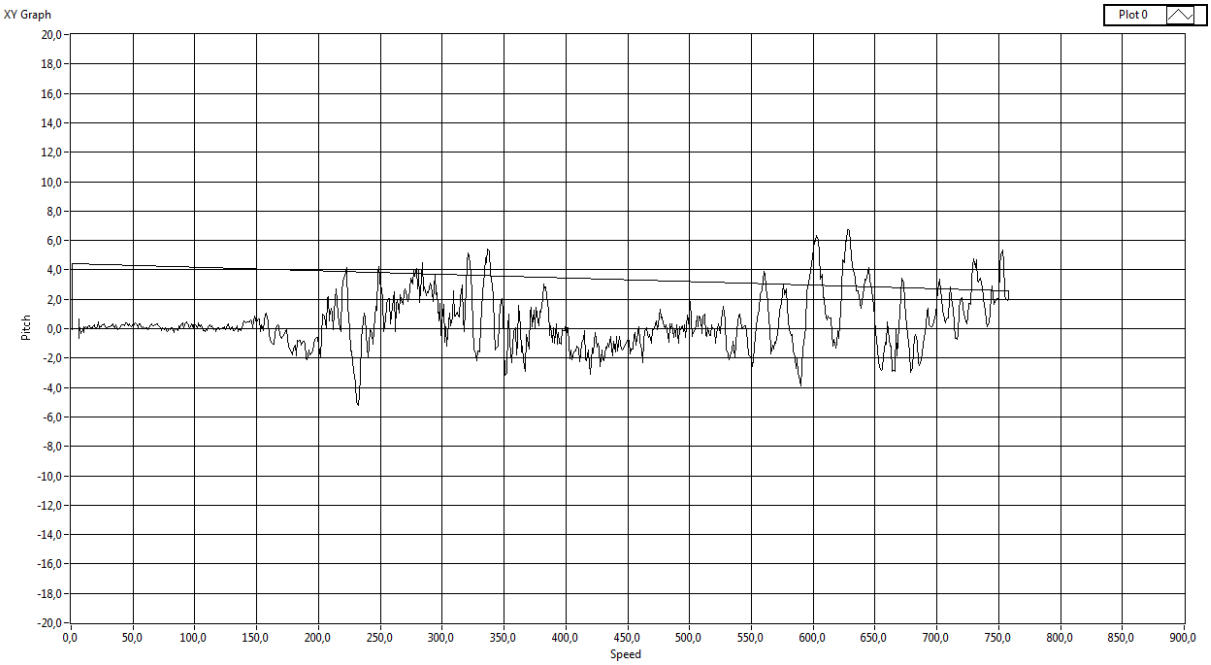
Kde „*ax*“ představuje zrychlení v ose počítaného úhlu a „*az*“ zrychlení v ose Z. Výsledná hodnota lze použít do vzorce pro komplementární filtraci, který je:

$$\alpha' = coeficient * \left(\alpha' + (x * time)\right) + \left((1 - coeficient) * x_{acc}\right)$$

$$coeficient = \frac{\tau}{\tau + time}$$

Výsledná veličina po filtraci je rovna „α´“. Proměnná „*x*“ reprezentuje informaci z gyra (°/s). Konstanta „τ“ byla zvolena 0,7, tato hodnota vykazovala nejlepší výsledky. Proměnná „*time*“ (s) je rovna časovému intervalu mezi jednotlivými vzorky a je důležitá pro výpočet aktuálního úhlu z informace o změně úhlu, kterou poskytuje gyro.

Výsledná informace měla po filtraci následující chybu (Graf 2).



Graf 2: Komplementární filtrace

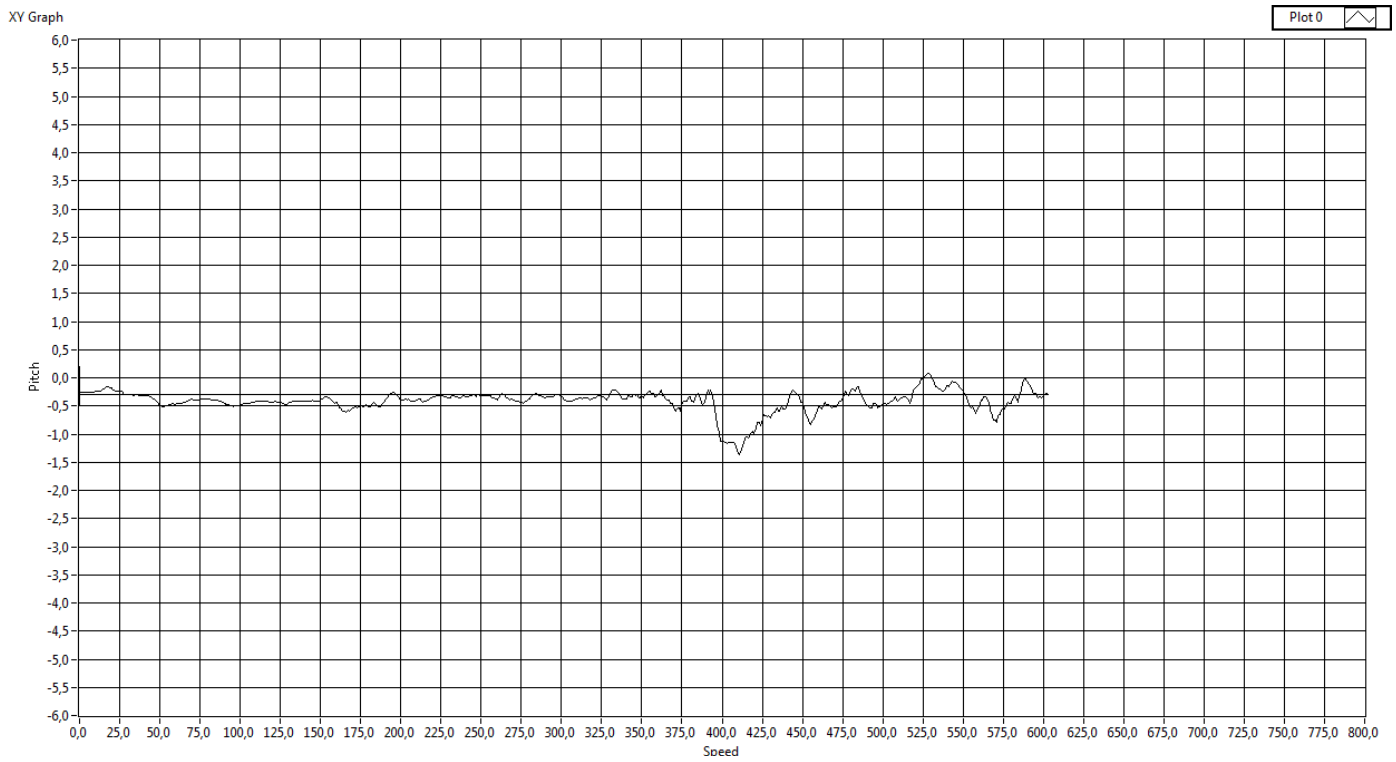
Přesto, že velikost chyby stále nedovolovala kvalitní regulaci, komplementární filtr výrazně tuto chybu omezil. Po těchto zkušenostech jsme AHRS modul odšroubovali od základní desky a pružně jej umístili na gumičky 2cm od desky. Pružné uchycení DPS zmenšilo chybu na polovinu předchozí odchylky. Následující testy ukázaly, že filtrace omezila chybu na úkor rychlosti žádané informace, která tak znemožnila dostatečně rychlou reakci.

Druhý SW, který byl k AHRS modulu výrobcem vyvinut, již obsahuje přepočet na úhel rotace a filtraci, konkrétně se jedná o Kalmanovu filtraci.

Hodně zjednodušeně by se dalo říci, že Kalmanův filtr je takový „vylepšený" odhad plovoucího průměru. Toto vylepšení spočívá v rozdělení algoritmu do dvou kroků:

predikci nového stavu a korekci integrací nového měření. (Winkler, 2005)

S tímto filtrem, zvětšením rozsahu na ±16 g a pružným uchycením DPS AHRS modulu jsme dosáhli nejlepších výsledků. Graf 3 znázorňuje průběh chyby v závislosti na otáčkách.



Graf 3: Kalmanova filtrace

3.3.2.2 Úprava softwaru

Úprava softwaru byla provedena v prostředí „Arduino alpha“, které slouží pro programování Arduino modulů.

Knihovna I2C

Nastavení rozsahu na ±16 g

```
void Accel_Init()
{
  Wire.beginTransmission(AccelAddress);
  Wire.send(0x2D);  // power register
  Wire.send(0x08);  // measurement mode
  Wire.endTransmission();
  delay(5);
  Wire.beginTransmission(AccelAddress);
  Wire.send(0x31);  // Data format register
  Wire.send(0x0B);  // set to full resolution (16g)
  Wire.endTransmission();
  delay(5);
  // Because our main loop runs at 50Hz we adjust the output data rate to 50Hz (25Hz bandwidth)
  Wire.beginTransmission(AccelAddress);
  Wire.send(0x2C);  // Rate
  Wire.send(0x09);  // set to 50Hz, normal operation
  Wire.endTransmission();
  delay(5);
}
```

Obrázek 10: Úprava rozlišení

Knihovna Output

Sestavení dat k odeslání a jejich úprava pro přenos bez desetinné čárky

```
void printdata(void)
{
      Serial.print("$,");

      #if PRINT_EULER == 1
      Serial.print((int)(ToDeg(roll)*100));
      Serial.print(",");
      Serial.print((int)(ToDeg(pitch)*100));
      Serial.print(",");
      Serial.print((int)(ToDeg(yaw)*100));
      Serial.print(",");
      Serial.print((int)(ToDeg(Gyro_Vector[2])));
      Serial.print(",#");
```

Obrázek 11: Přenos bez desetinné čárky

Nejprve byl změněn rozsah akcelerometru (Obrázek 9) v inicializaci akcelerometru. Další úprava byla provedena v knihovně „Output“ (Obrázek10), kde jsme upravili výstupní datový paket tak, aby byl snadno zpracovatelný na přijímací straně. Odesílaná data jsme před odesláním vynásobili stem a tím odstranili desetinou čárku. Kromě hodnot rotací ve třech osách (jaw, pitch, roll) jsme k paketu přidali i informaci z gyra v ose „Z“. K tomuto kroku jsme byli donuceni, jelikož rotace v ose „Z“ reprezentovaná proměnnou „YAW“ je počítána z hodnot magnetometru (kompasu), který není spolehlivý a je značně náchylný na magnetické pole (rušení od motorů) a na magneticky vodivé předměty (přítomnost oceli).

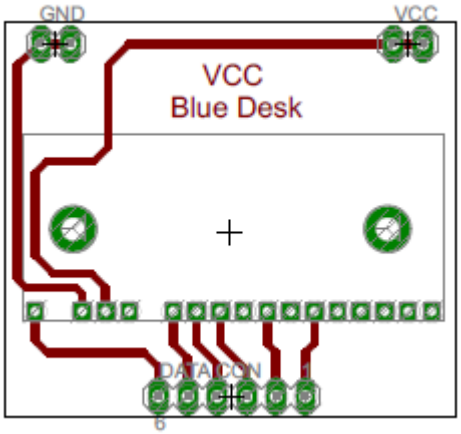
3.4 Komunikace letounu s PC

Pro přenos dat mezi jednotkou a PC, je v současnosti použit bluetooth modul firmy „Spezial electronic“ „OEMSPA310“ (Obrázek 11) (Speziel Electronic)a to z důvodu jeho dostupnosti a kompatibility s periferií počítače (není nutný další HW). Další variantou je využití šesti kanálů, které jsou připraveny pro připojení RC (modelářské vysílačky). Tím ovšem ztratíme možnost plné kontroly letounu a zpětnou vazbu.

Obrázek 12: Bluetooth modul

Jedná se o základní model bluetooth modulu, který nedisponuje příliš velkým dosahem (75m). Samotný modul je umístěn na redukci (Obrázek 12), kterou jsme vytvořili pro jednoduchou manipulaci s modulem a možnost snadné výměny za XBee modul. Výměna modulu je nezbytná v případě používání letounu na větší vzdálenosti. Nevýhodou zmiňovaného Xbee modulu a obecně standardu IEEE 802.15.4 je jeho omezení z hlediska přenosové rychlosti, která je maximálně 57 600Baud. Při přechodu na Xbee je také nutné vytvoření obousměrně komunikujícího modulu, který zajistí spojení s PC.

V praxi to může znamenat propojení stejného Xbee modulu s FTDI čipem, pomocí něhož může být modul připojen k PC přes USB.



Obrázek 13: Redukce pro komunikační modul

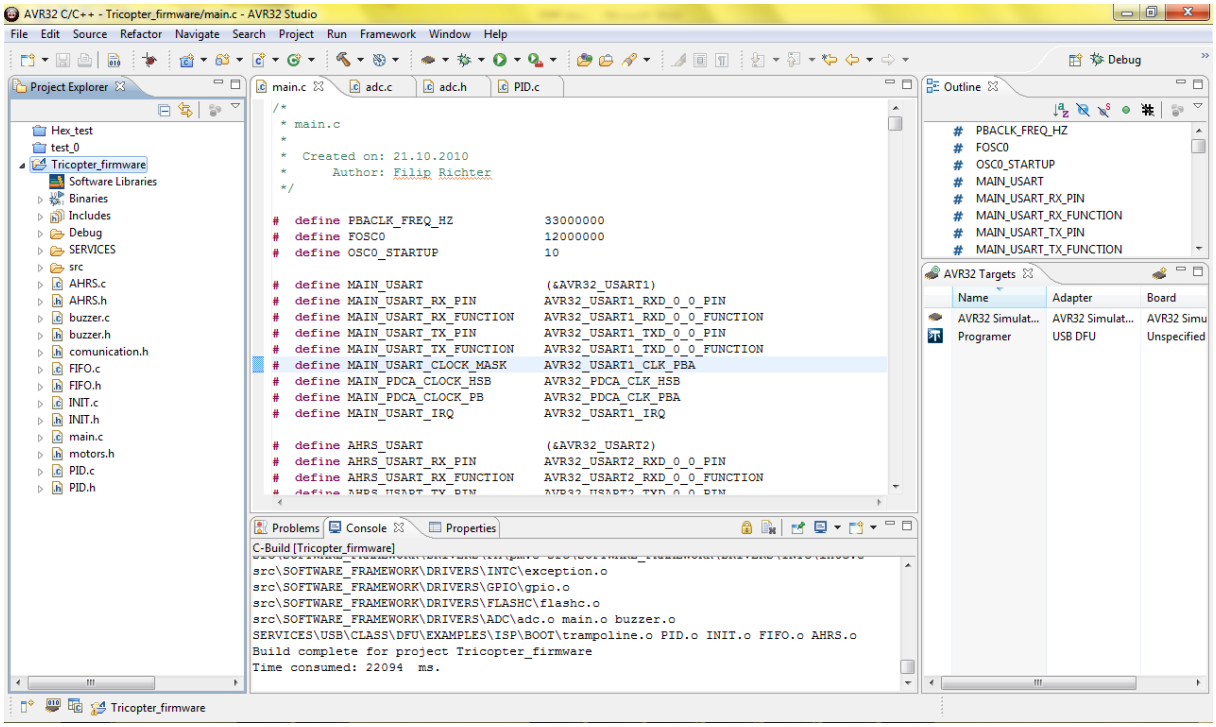
Kompatibilita je zajištěna v případě použití modulu stejné firmy (Spezial Electronic). Jedná se o standard IEEE 802.15.4 (zigbee), pracující na frekvenci 2,4 GHz. Konkrétní modul vykazuje dosah až 400m ve volném prostranství. Dalším důvodem, proč jsme zvolili právě tento modul, je jeho identická signalizace s „OEMSPA310“ modulem, kterou zajišťují 3 LED a stejné rozměry děr pro uchycení, tudíž není výměna komunikačního standardu ničím omezena a oba standardy realizované zmiňovanou firmou, jsou kompatibilní se základní deskou. Pro konfiguraci obou komunikačních modulů slouží SW „Seriál Port Adapter Toolbox“ vyvinutý výrobcem. V něm je možné nastavit přenosovou rychlost a další parametry komunikace (v současnosti 115200, 8b, bez parity, 1 stop bit), heslo pro komunikaci atd.

3.5 Napájení

Letoun je navržen na 3 Li-Pol akumulátory o minimální kapacitě 2200mA/h. Napětí každého akumulátoru je 11,1V a jsou zapojeny paralelně. V této konfiguraci je doba letu přibližně 12 minut. Pro jednotku je napětí z akumulátorů stabilizováno spínaným 5V modelářským zdrojem (až 3A). Zvolený typ baterie je náchylný na přebití, podvybití a pro jeho nabíjení je nutná speciální nabíječka. Aby nedošlo k poškození baterií, je nutné, aby všechny tři baterie měly před zapojením stejné napětí!

4 SOFTWARE ŘÍDÍCÍ JEDNOTKY

Řídící jednotka obsahuje hlavní procesor firmy ATMEL z řady AVR32 AT32UC3A1512. Program pro něj je napsán v jazyce C, ve vývojovém prostředí AVR32 studio, které dodává přímo firma ATMEL jako příslušenství. AVR32 Studio je založeno na prostředí Eclipse, s doinstalovaným frameworkem (ATMEL, 2010) pro procesory AVR32.



Obrázek 14: Vývojové prostředí AVR32 Studio

Program obsahuje několik knihoven (hlavně pro ovládání vnitřních periferii) pocházejících z originálního frameworku od společnosti ATMEL. Dále bylo několik knihoven speciálně vytvořeno přímo pro projekt Tricopter. Základ programu pracuje na bázi obsluhy přerušení i hlavního programu. Velká většina vstupů i výstupů je řešena přes interní periferie. Vzhledem k jejich dostupnosti a jednoduchosti se nevyplatí softwarově zpracovávat protokoly přímo z vstupně-výstupních pinů.

4.1 PWM

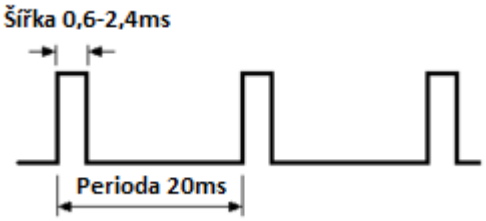
Procesor AT32UC3A1512 obsahuje sedmi-kanálový generátor pulzního šířkově modulovaného signálu. Tento generátor pracuje s šestnácti-bitovým rozlišením, což

poskytuje 65536 možných kroků nastavení střídy. Interní PWM generátor původně není určený jako zdroj signálu pro modelářské prvky.

4.1.1 Kompatibilita s modelářským protokolem

Modelářský PWM protokol je založen, obdobně jako čisté PWM, na šířce pulzu. U standartního PWM se průběh definuje frekvencí (v Hz) a střídou (v procentech). Střída udává procentuální zastoupení aktivní části signálu. V případě, že je aktivní část signálu reprezentována logickou nulou, mluvíme o PWM v inverzním režimu.

Standartní modelářský protokol využívá PWM jen část z celého rozsahu. Pracuje na frekvenci 50Hz, a hodnota je definována nikoliv procenty, ale časovým údajem v milisekundách, respektive v mikrosekundách. Obvykle u servomotorů je rozsah v rozmezí 600µs a 2400µs. Hodnota 1500µs bývá označována, jako středová poloha. Pracovní rozsah regulátoru motoru je určen podle návodu k použití u něho přiloženého, v nastavení regulátoru jde však změnit, aby vyhovoval požadavkům.



Obrázek 15: Signál modelářského protokolu

Vzhledem k uvedeným rozsahům je třeba omezit celý rozsah PWM generátoru na hodnoty přepočtené vzhledem k celému rozsahu, jenž při 50Hz odpovídá 20ms. Výsledek pro modelářský servomotor odpovídá 3% až 12%.

Přestože, by měl být výpočet univerzální je použita funkce, která nastavuje každý kanál samostatně, je tak možno omezit signál, například kvůli mechanickým součástem navazujícím na servomotor a jejich omezením. Pro omezení jsou vytvořena pole obsahující minimální, maximální a počáteční hodnoty.

```
int pwm_max[] = {1900,1900,1900,1900,2250,2250,1900};
int pwm_min[] = {1100,1100,1100,800,750,680,750,1100};
int pwm_startup[]={1100,1100,1100,1500,1500,680,1500,1100};
```

Dalším pokusem bylo otestováno, že všechny použité součásti jsou schopné pracovat i na 100Hz signálu. Vzhledem k vlastnostem přenosového řetězce se však

ukázalo, že to nebude využito a tak v aktuální verzi firmwaru běží PWM generátor na 50Hz.

4.1.2 PWM výstupy pro regulátory

První tři PWM kanály jsou využity pro ovládání regulátorů. Jejich minimum a maximu je nastaveno podle návodu k použití na hodnoty minimum 1100μs a maximum 1900μs. Dalším parametrem, který vytvořené funkce využívají, je počáteční hodnota. Jelikož jde o motory, které mají být po zapnutí v klidu, je počáteční hodnota nastavena na minimální hodnotu. Funkce, která nastavuje hodnotu, přičte předanou proměnnou k minimu a tím nastaví požadovaný výkon.

4.1.3 PWM výstupy pro servomotory

Další tři jsou využity pro ovládání servomotorů. Jeden pro náklon ocasního motoru, a dva jako příprava pro servomotory ovládající fotoaparát pro fotografování leteckých snímků. Jejich rozsahy jsou nastaveny samostatně podle potřeb na rozsah mechanického pohybu servomotoru. Pro použití se zde využívají dvě různé funkce. První shodně jako u regulátorů přičítá dodanou hodnotu k minimu a nula tak znamená minimum. Druhá možnost je využití upravené funkce, kde se využívá počáteční hodnoty jako reference. Pokud se pak dodá kladná hodnota, znamená to posuv směrem k maximu o dodanou hodnotu, v případě, že jde o zápornou hodnotu, přibližujeme se k minimu. Této druhé funkce je využito u ocasního servomotoru. Výchozí hodnota je nastavena na střed a hodnota dodaná z regulace pak nabývá kladných i záporných hodnot a tím způsobí kladné i záporné výchylky náklonu motoru.

4.2 PWM výstup pro regulaci výkonu reflektoru

Poslední výstup je rezervován pro ovládání výkonové LED reflektoru. LED reflektor by mohl být v budoucnu použit s fotoaparátem pro pohyb v neosvětlených prostorách. Pro jeho připojení stačí použít modul s výkonovým tranzistorem, ten připojit na výstupní konektor a vhodně nastavit maxima a minima PWM rozsahu.

4.3 Čítač/časovač

Hlavní procesor obsahuje také tříkanálový interní čítač/časovač jeho rozlišení je shodné jako u PWM, šestnácti-bitové. V aktuální verzi firmwaru jsou využity dva a jeden je připraven pro budoucí využití na výstup.

4.3.1 Detekce chyby v komunikaci s deskou senzorů

Nultý kanál je využit využitý jako watchdog komunikace mezi deskou senzorů polohy a řídící jednotkou, zároveň výstup přiveden na signálku. V případě že se čítač, taktovaný vnitřní sběrnicí PBA, pravidelně nevynuluje příchodem celého paketu dat ze senzorů polohy, dojde k přerušení a vyslání chybové zprávy telemetrickým kanálem do řídícího počítače. Kromě přenosu zprávy je chybový stav signalizován blikáním žluté signálky.

4.3.2 Akustická signalizace

Na výstupu prvního čítače je připojen piezoelektrický měnič, jenž je využíván pro signalizaci chybových a provozních stavů. Pro jeho ovládání byla vytvořena jednoduchá knihovna „buzzer.c“, která obsahuje funkce pro nastavení určité frekvence tónu. Zároveň obsahuje funkci pro inicializaci celého kanálu čítače.

4.3.3 Druhý kanál čítače

Poslední kanál připojen na opticky oddělený výstup a je tak připraven pro budoucí využití.

4.4 RTC

Obvod RTC je v řídící jednotce využíván jako zdroj pravidelných impulzů o frekvenci 1Hz. Tento signál se využívá pro spouštění pravidelných akcí. Příchod přerušení je nastaven na nejnižší prioritu, aby nedocházelo ke zbytečnému prodlení v operacích s vyšší prioritou.

Přerušení je využíváno pro kontrolu spojení s ovládacím počítačem. Z počítače je ovládacím kanálem pravidelně posílán bezpečnostní znak, a pokud do daného času nepřijde, řídící jednotka se přepne do nouzového režimu. V případě, že je model v testovacím režimu, tak namísto přechodu do nouzového režimu je provedeno nouzové vypnutí.

4.5 USART

V řídící jednotce jsou dále využity, pro náš projekt jedny z nejdůležitějších, interní moduly USART. Hlavní procesor obsahuje moduly USART čtyři, v aktuální verzi jsou využity dva. Moduly mají každý samostatné nastavení parametrů. Knihovna

z originálního frameworku je na první pohled velmi jednoduše ovladatelná. I tak se však po prvních pokusech s tímto rozhraním se však objevil problém. Sám modul i knihovna obsahují velmi obsáhlou správu chybových stavů, a modul se do zavolání funkce na správu chyb zablokuje. To způsobovalo problém, kdy data přicházela v okamžiku hardwarově vyvolaného resetu. Moduly pak byly v chybovém stavu a neumožňovali další používání. Pro vyřešení tohoto problému byla vytvořena funkce na správu chyb a ta je volána při každém příchodu chybového stavu.

Komunikace probíhá ve znakové podobě. Všechny odeslané i přijaté znaky jsou interpretovány podle ASCII tabulky. Jediná výjimka je skupina řídících znaků, ty mají jediné využití. Je nutná jejich bezchybná interpretace i v případě částečné ztráty dat. Parametry a zařízení na nich připojené jsou vypsány v tabulce (Tabulka 1).

Název Usartu	Cílové zařízení	Směr komunikace	Přenosová rychlost
USART 0	Měřící deska	Rezervováno	----------
USART 1	Ovládací počítač	Obousměrný	115 200 Bd
USART 2	Modul AHRS	Pouze příjem	57 600 Bd
USART 3	GPS Přijímač	Rezervováno	----------

Tabulka 4: Seznam periferií na sběrnicích USART

4.6 Program

Program je rozdělený na několik částí, jsou to inicializace, hlavní smyčka a obsluhy přerušení.

4.6.1 Inicializace

Na začátku inicializace je procesor přepnut z vnitřního zdroje signálu 115kHz na násobičku PLL, která používá jako zdroj signálu 12MHz krystal a ten je pak násoben a procesor používá o rychlosti 33MHz.

V této části jsou nastaveny veškeré výše zmíněné periferie. Dále jsou zde vytvořeny některé struktury nutné pro nastavení a běh programu. Jako poslední je nastaven správa přerušení, zaregistrována veškerá přerušení, a nakonec odeslána inicializační sekvence do počítače a povoleno globální přerušení.

4.6.2 Hlavní smyčka

V hlavní smyčce se v každém průběhu zkontroluje proměnná AHRS_counter, a v případě, že je její hodnota 6, což znamená příchod celého paketu z AHRS, je provedena regulace ocasního servomotoru a provedena funkce update_motors(), jejímž cílem je výpočet a nastavení výkonů jednotlivých motorů v závislosti na proměnné gas (výkon motorů podle ovládání), a regulací levopravá (roll) a předozadní (pitch) osy regulace.

Druhým úkolem je kontrola změny režimu (proměnná mode), v případě, že proběhla změna, odešle se aktivita do řídícího počítače. V návaznosti na to se přenastaví parametry regulací, na hodnoty odpovídající nově zvolenému režimu.

Posledním úkolem hlavní smyčky je odeslání chybových hlášení z centrální správy chyb do ovládacího počítače.

V hlavní smyčce zbývá dostatek výpočetního času pro přidání dalších funkcí nebo výpočtů, je však nutné dodržet, aby se hlavní smyčka opakovala tak často, aby nebyla brzděna regulace.

4.6.3 Obsluhy přerušení

Program obsahuje několik obsluh přerušení. Jsou zaregistrovány ve správě přerušení a jsou rozděleny do skupin podle priorit.

Nejvyšší prioritu má přerušení od komunikačního rozhraní s počítačem a komunikační rozhraní s modulem senzorů polohy. Tyto přerušení je nutné provádět v co nejkratším čase důvodu nutnosti pravidelného opakování regulace. Je však také nutné znaky ze sériových rozhraní dostatečně rychle číst, aby nedošlo k přepsání znaku dalším příchozím znakem.

4.6.3.1 Přerušení od sériové linky USART 1

Přerušení je rozdělené na dvě části, jeden modul USART, totiž generuje pouze jedno přerušení. Druh přerušení je rozlišen podle obsahu registru CSR. Pokud prvek struktury rxrdy je vyhodnocen jako pravda, pak se jedná o přerušení z důvodu úspěšného přijetí znaku a je spuštěna první část přerušení.

První část přerušení má za úkol rozhodnout, která proměnná bude nastavována, podle prvního znaku. Dále pak při každém dalším spuštění přerušení je přijatý znak uložen do pole odpovídajícího prvnímu znaku. Pokud je přijatý znak středník znamená

to, že je to konec hodnoty. Následně je spuštěna funkce, která má za úkol převést pole znaků na číslo, jež je její návratovou hodnotou.

Popsaný způsob platí pro předávané proměnné. V případě, že je předáván pouze příkaz bez hodnoty, nespustí se funkce na převod čísla, jelikož je následující znak hned středník. Místo funkce na převod čísla je zavolána funkce odpovídající příkazu.

Druhá část je vlastně přípojný bod softwarové realizace paměti FIFO. Slouží pro reakci na prázdný odesílací registr. V okamžiku příchodu přerušení je do odesílacího registru zapsán další znak z paměti FIFO.

4.6.3.2 Přerušení od sériové linky USART 2

Vzhledem k tomu, že modul senzorů AHRS je připojen pouze jednosměrně, je obsluha přerušení o to jednodušší. Principiálně je třídění dat velmi podobné třídění dat z ovládacího počítače. Rozdíl je pouze v tom, že AHRS posílá všechny hodnoty postupně a nejsou odděleny úvodními symboly ale pouze čárkou. Od toho se odvíjí samotný třídící protokol.

4.6.3.3 Přerušení od modulu RTC

Přerušení obvodu RTC je momentálně využíváno jako zdroj pravidelného signálu pro watchdog komunikace s ovládacím počítačem. Toto přerušení je připraveno pro pravidelné časování dalších akcí. Přerušení je zařazeno do kategorie skupiny INT0, což znamená, že má nejnižší prioritu.

4.6.3.4 Externí přerušení

Externí přerušení je využito pro obsluhu tlačítka na jednotce. Tlačítko slouží pro změnu režimu ovládání. Přerušení je nastaveno na detekci sestupné hrany a má povolenou filtraci, která omezí zákmity tlačítka během stisku.

4.7 Vlastní vytvořené knihovny

Součástí celého programu je několik knihoven vytvořených na míru. Tyto knihovny vznikli sdružením funkcí stejného významu, které byly původně vytvořeny v hlavním souboru.

4.7.1 AHRS

Tato knihovna již není aktuálně využívána. V předchozích verzích se však používala pro zpracování čistých dat z AHRS a jejich přepočet a integraci v závislosti na čase mezi jednotlivým měřením.

Dále obsahuje matematickou funkci atan(), s přepočtem z radiánů na stupně. Tato funkce je využívána pro výpočet úhlu z hodnot zrychlení.

Poslední funkcí je komplementární filtrace, ta měla za úkol z dodaných čistých dat pomocí komplementárního filtru vypočítat aktuální úhel. Funkce tohoto výpočtu se jevila jako velmi dobrá, ačkoliv během dalších testů se objevil problém související s rychlostí zpracování. Data zpracovaná touto metodou byla s únosnou chybou, problémem však bylo, že vlivem komplementární filtrace docházelo nadměrnému zpoždění regulace.

Po přechodu na nový firmware modulu AHRS, který používá Kalmanovu filtraci, již není knihovna využívána.

4.7.2 Buzzer

Knihovna Buzzer obsahuje funkce pro ovládání signalizačního piezo-měniče. Jsou zde vytvořeny funkce pro spuštění měniče se zadanou frekvencí, změny frekvence a vypnutí měniče. Měnič je připojen na výstup jednoho kanálu interního modulu čítače. Modul čítače musí být po spuštění nastaven, proto knihovna obsahuje funkci, která je volána z inicializace a nastaví čítač a připraví ho pro použití. Jediná složitější část této knihovny obsahuje přepočet z frekvence na hodnotu pro registry čítač, jinak knihovna prakticky obsahuje pouze přiřazování hodnot do registrů.

4.7.3 Errors

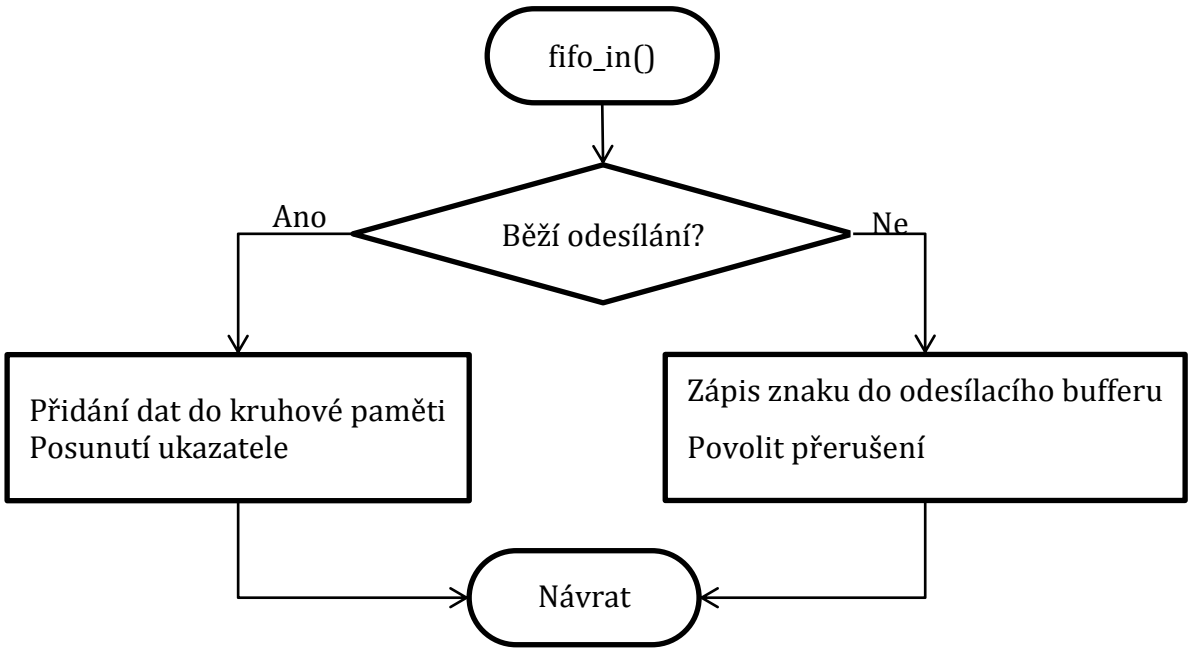
Z důvodu množících se detekcí chyb v jednotlivých částech programu, bylo třeba vytvořit jednotnou databázi a správu chybových stavů. Jelikož samotná řídící jednotka nemá široké možnosti pro indikaci chybových stavů, veškeré kódy chyb se odesílají do ovládacího počítače. Zde se dále přivádí na uživatelsky čitelnou chybu, pomocí databáze chyb, ta je pak zobrazena v grafické stránce ovládacího softwaru. Knihovna pro správu

chyb si také uchovává poslední chybu a chybu s nejvyšší prioritou. Prioritu chyby určuje její kód, jehož nejnižší hodnota je zároveň nejvyšší priorita.

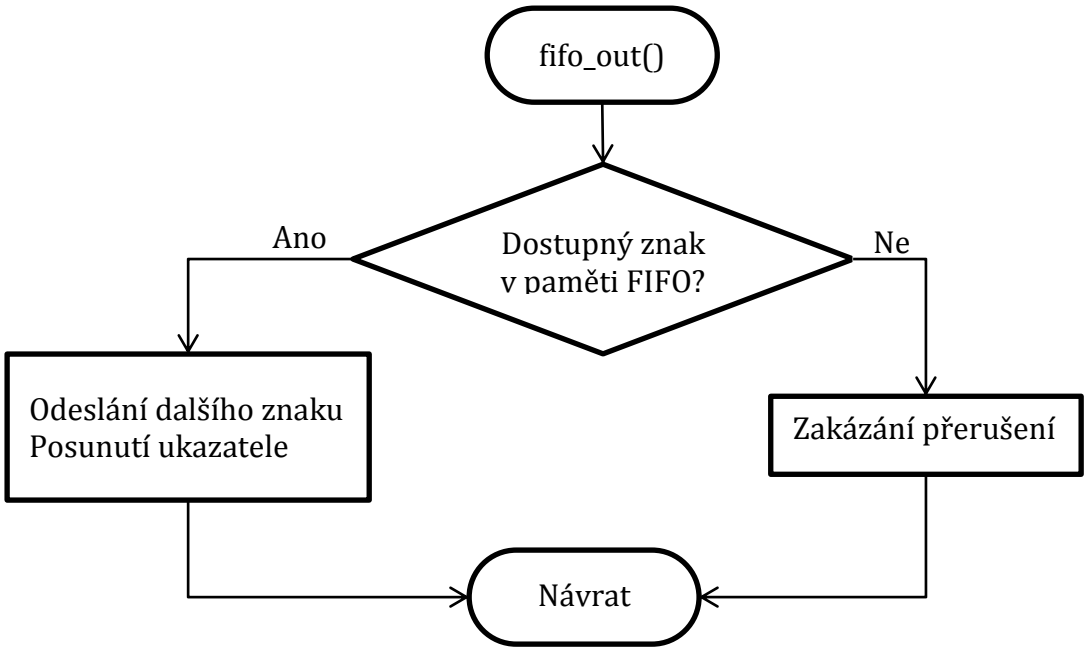
Knihovna obsahuje definici struktury, obsahující chybové kódy. Ta je potom v hlavním programu vytvořena jako globální proměnná.

V případě, že nastane situace, vyžadující chybové hlášení, je zavolána funkce, která nastaví hodnotu proměnné pro odesílání, poslední chyby, i chyby s nejvyšší prioritou. Další obsažené funkce jsou pro vyčtení jednotlivých uložených chyb. Poslední zatím obsažená funkce slouží pro smazání chybových hlášení a připravení pro další provoz. Jediné co zůstává, až do resetu procesoru je hodnota počtu chyb.

4.7.4 FIFO



Obrázek 16: Vývojový diagram funkce fifo_in()



Obrázek 17: Vývojový diagram funkce fifo_out()

Knihovna FIFO byla vytvořena v závislosti na stavbě programu. Parametry předávané do ovládacího počítače se odesílají rychlostí 115 200Bd, což není zdaleka rychlost procesoru a tak mezi odesíláním jednotlivých znaků zbývá výpočetní čas. Tento čas však nebylo možné využít v závislosti na víceznakovém blokujícím odesílání. Problém bylo třeba nějak řešit, jelikož základ regulace a auto-stabilizace probíhá v hlavní smyčce. Proto nesmí být procesor příliš dlouho blokován.

Řešení, které se přímo nabízí, je spolupráce přerušení a kruhové paměti FIFO[1]. Jako inspirace pro vytvoření paměti FIFO byla použita Wikipedie. Vytvořená knihovna však neobsahuje žádnou část kódu ze serveru Wikipedia (Queue (data structure)). Funkce byly přepracovány a upraveny do podoby odpovídající našim požadavkům.

Místo přímého zápisu na do modulu USART se nyní používají funkce fifo_in (zápis po znaku) (Obrázek 16), fifo_string (zápis řetězce) a fifo_number (zápis celočíselné hodnoty). Poslední dvě vyjmenované funkce provedou převod na jednotlivé znaky a přes funkci fifo_in se zapíší do kruhové fronty. V případě, že v okamžiku zápisu neprobíhalo vysílání, je vysílání okamžitě zahájeno provedením neblokující funkce odeslání dat a povolení přerušení pro příznak prázdného odesílacího registru. V opačném případě se znaky zapíší do kruhové paměti.

[1] FIFO – First in First out – Kruhová paměť tzv. Fronta

V okamžiku příchodu přerušení se rozhodne (Obrázek 17), zda kruhová paměť obsahuje další znaky pro odeslání. Pokud ano, další znak se začne odesílat (obdobně jako v předchozím případě neblokujícím odesíláním) a je proveden návrat do přechozí prováděné operace. Pokud není přítomen další znak pro odeslání, je přerušení zakázáno a proveden návrat.

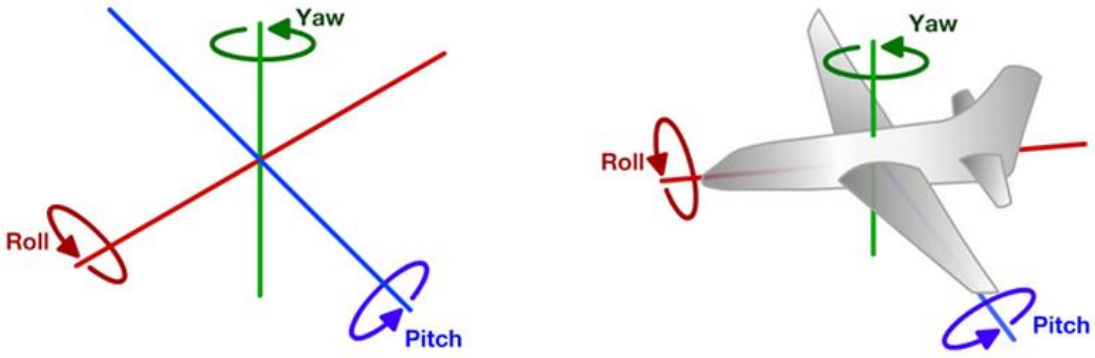
4.7.5 Motors

Jelikož model obsahuje tři motory, není souměrný, z čehož vyplývá nutnost přepočtu tříosého řízení (Obrázek 18) (Tourning machine, Pitch, Roll and Yaw) na třímotorový model ve tvaru trojúhelníku. Přepočet je založen na přičítání a odčítání od celkového výkonu, patřícího všem motorům, v závislosti na hodnotách jednotlivých os.

K předním dvěma motorům přičítá hodnota z předozadní osy. Dvojnásobek této hodnoty je odečítán od zadního motoru. Kromě toho se ještě přičítá a odečítá hodnota levopravé osy, ale pouze k levému a pravému motoru, jelikož zadní motor je v ose, tudíž není třeba jeho výkon upravovat.

Funkce z této knihovny zároveň kontrolují minimální hodnotu, pro běh motoru, kdy je motor ještě schopen se otáčet. Je-li hodnota mezi minimálním potřebným výkonem a nulou, udržuje se na minimální hodnotě, aby se motor nezastavil.

Popsané principy jsou tvořeny jedinou funkcí update_motors(), ta má za úkol, v závislosti na dodané hodnotě z regulace popř. manuálního řízení, provést výpočet a nastavit potřebné výstupy řadiče PWM.



Obrázek 18: Rozložení jednotlivých os

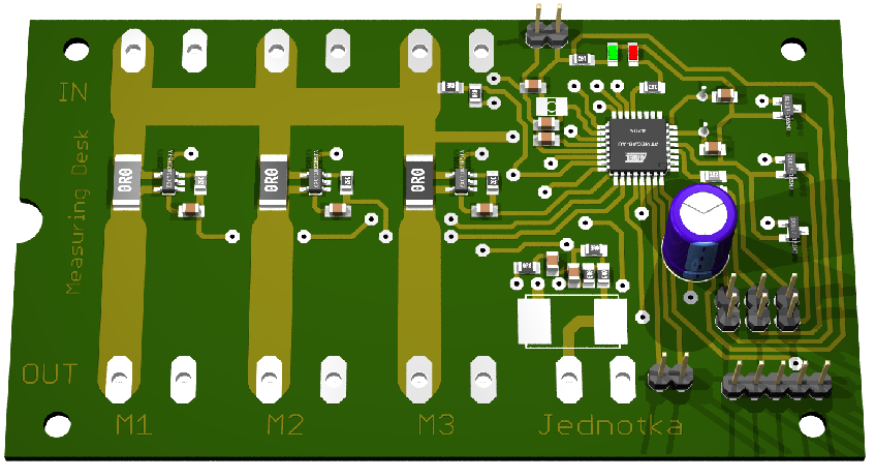
4.7.6 PID

Pro regulaci v jednotlivých osách bylo vytvořeno několik funkci soustředěných v knihovně PID. Knihovna obsahuje také definici struktury, jež je potřebná pro práci několika regulací souběžně. Veškeré nastavení je uloženo v samostatné struktuře pro každou regulaci. Pro nastavení se používá funkce PID_setup(), které se předá ukazatel na strukturu, jež se nastavuje. Jako další se funkci předají čtyři parametry (zesílení složek P, I, D a omezující faktor, který omezí maximální hodnotu složky I). Funkce pak zapíše jednotlivé parametry do výše zmíněné struktury, aby s nimi mohla pracovat funkce PID.

Funkce PID dostává jako parametr ukazatel na strukturu regulace a dvě proměnné (žádanou a aktuální hodnotu). Z žádané a aktuální hodnoty se provede výpočet regulační odchylky. Z regulační odchylky vynásobené zesílením složky P, pak vychází proporcionální složka regulace. Do integrační složky se při každém průběhu přičte aktuální odchylka vynásobená odpovídajícím zesílením. Integrační složka je pak oříznuta v případě překročení hodnoty dané omezujícím faktorem. Jako poslední se vypočítává derivační složka, jež je tvořena rozdílem aktuální a předcházející hodnoty. Jednotlivé složky se sečtou a výsledek se vrací jako návratová hodnota.

5 MĚŘÍCÍ DESKA

Pro diagnostiku letounu z hlediska jeho bezpečného chodu, jsme navrhli a vyrobili měřící desku (Obrázek 13), která má za úkol, měření napětí, proudů, případně teploty a dalších veličin (k tomu slouží rozšiřující konektor, ke kterému je možné připojit analogové senzory nebo senzory 1-wire). Tyto informace jsou nutné, pro výpočet kapacity baterií a tak stanovení maximální doby letu, pro diagnostiku selhání motoru, regulátoru nebo přítomnost zkratu. Dále byly na desku umístěny výstupy pro 3 LED, které jsou situovány na koncích ramen letounu a uživatele informují o orientaci trikoptéry. Měřící deska pracuje nezávisle na řídící jednotce, které pouze odesílá informace o měřených veličinách pomocí sběrnice USART. Toto rozdělení jsme zvolili, abychom oddělili výkonovou část od té logické, umožnili tak snadnou modifikaci a předešli chybám. Měřící deska byla vyrobena dodatečně a stále se na ní pracuje. Nutnost měření základních veličin vyplynula v průběhu vývoje.



Obrázek 19: Měřící deska

5.1 Hardwarové řešení

Pro měřící desku, jsme vybrali MCU firmy Atmel „ATmega8“ (Atmel) s jádrem AVR. Tento mikrokontrolér je pro zvolenou aplikaci plně dostačující a vzhledem k našim zkušenostem s tímto jádrem, nebyl důvod volit jiný MCU. Pro měření napětí a proudů jsme využili integrovaný 10b AD převodník s referenčním napětím 2,56V (zvolena vnitřní reference).

V praxi to znamená, že napětí na každém vstupu AD převodníku bude reprezentováno hodnotou od 0 do 1023 (0V až 2,56V), napětí 2,56 je tedy maximální hodnota měřená převodníkem. Protože nevyužíváme externí referenční napětí, na vstupu „AREF“ externí reference je připojen kondenzátor proti zemi (Atmel). Deska je osazena třemi proudovými sledovači, které jsou napájeny z rozdílu potenciálů mezi zemí a snímacím vstupem VSESE+, minimální napětí pro správnou funkci obvodu je 2,5V. Toto napětí je úbytek na snímacím odporu a na zátěži do které se proud měří.

5.2 Měření napětí

Měření napětí baterií je realizované přes dělič napětí s dolní propustí. Jelikož je maximální velikost napětí omezena referencí, musíme napětí baterie upravit tak, aby bylo měřitelné. Baterie při maximálním nabití vykazuje 12,6V, tuto hodnotu jsme si zvolili jako maximum, na jejímž základě jsme vypočítali dělič napětí, který má poměr 1:4,92, to znamená, že při vstupním napětí na děliči 12,6V bude na výstupu děliče 2,56V. Vzhledem k nedostupnosti potřebných hodnot rezistorů pro realizaci děliče v daném poměru, je skutečný poměr o něco větší, tudíž výstupní napětí děliče je menší. Tato skutečnost se promítla do kalibrace daného kanálu. Dolní propust nám zajistí odfiltrování vysokých frekvencí, které jsou nežádoucí při měření. Citlivost měření je závislá na rozlišení převodníku a dělícím poměru. Přesnost měření se dá spočítat jako podíl referenčního napětí, k rozlišovací schopnosti AD převodníku, tedy 2,56/1024 = 0,0025V. jelikož je měřené napětí vydělené hodnotou 4,92, musíme touto hodnotou vynásobit citlivost, abychom získali skutečnou měřící citlivost, která je 0,011325V.

5.3 Měření proudu do zdroje 5V

Měření proudu do zdroje 5V a tedy i jednotky a servomechanismů je uskutečněno pomocí měřícího rezistoru. Napětí před rezistorem je známé z předchozího měření (3.2 měření napětí baterie), stejným způsobem je měřeno i napětí za rezistorem, rozdílem těchto dvou hodnot je úbytek napětí na rezistoru, který je přímo úměrný proudu. Volba velikosti měřícího rezistoru závisí na požadovaném úbytku napětí při určitém proudu a to zase ovlivňuje výkon, na který musí být rezistor navržen.

Při doplnění několika výkonnými servomotory je předpokládaný proud do této větve maximálně 3A, z této hodnoty vychází i náš návrh, návrh je úmyslně předimenzován oproti běžnému provozu, aby nedošlo ke zničení rezistoru vlivem proudu. Měřící rezistor byl zvolen 0,27Ω/3W. Pokud by rezistorem procházel proud 3A, pak by byl úbytek na rezistoru 0,81V a výkon na rezistoru by činil 2,43W. Rozlišovací schopnost můžeme vypočítat jako podíl rozlišovací schopnosti u měření napětí, k hodnotě odporu měřícího rezistoru. Výsledkem je velikost přesnosti měření proudu, která činí 41,94 mA.

5.3.1 Měření proudů do motorů

Vzhledem k velikosti měřeného proudu (špičkově až 25A), bylo nutné zvolit měřící rezistor o co nejmenším odporu, abychom na něm zbytečně neztráceli velký výkon. Z tohoto důvodu jsme zvolili rezistor 0,003Ω/3W v kombinaci s proudovým sledovačem ZXCT1021E5TA, tímto zapojením jsme dosáhli malého úbytku na rezistoru a zároveň desetinásobné zesílení tohoto úbytku pro přesnější měření. Zesílení je nutné, kvůli menším proudům, které vytvářejí malý úbytek napětí na měřícím rezistoru, který není měřitelný 10b AD převodníkem s referencí 2,56V. Citlivost takového měření se dá vypočítat následovně.

$$\Delta I = \frac{\frac{U_{ref}}{Rozsah\ ADC} * zesílení}{R_{měřící}} = \frac{\frac{2{,}56}{1024} * 10}{0{,}003} = 83,\bar{3}mA$$

5.3.2 Signalizace

Měřící deska obsahuje dvě signalizační diody, LED1 slouží k varovným účelům, kdy je například překročen měřící rozsah, LED2 je připojena k napájecímu napětí a indikuje přítomnost baterií. Součástí měřící desky jsou i tři tranzistory spínané výstupy pro LED, které jsou umístěny na koncích ramen letounu a zpřehledňují orientaci vznášedla.

5.3.3 Konektor pro rozšířené měření

Pro připojení dalších senzorů, je na DPS vyveden konektor s pěti nevyužitými piny mikrokontroléru, který obsahuje tři vstupy AD převodníku a tím umožňuje připojení i analogových senzorů.

5.4 Softwarové řešení

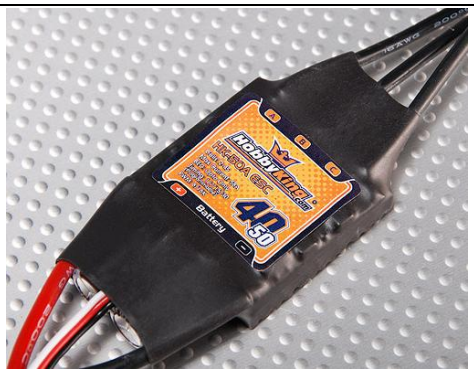
Program má za úkol cyklické měření napětí na vstupech AD převodníku v intervalu 100ms, v němž odesílá naměřené hodnoty do řídící jednotky. Komunikace s jednotkou je obousměrná, jednotka může pomocí sběrnice USART vyzvat měřící desku k přepnutí režimu svitu LED umístěných na ramenech letounu. Aby bylo zpracování naměřených dat v jednotce jednoduché, měřící deska vysílá data ve tvaru: písmeno udávající o jakou veličinu se jedná a samotnou naměřenou informaci (od 0 do 1023) zakončenou středníkem, za středníkem následuje další naměřená veličina. Změřená informace se nepřepočítává na skutečnou hodnotu napětí či proudu, aby se nemusela přenášet desetinná čárka, tím se zjednoduší třídění dat na přijímací straně. Měřící deska reaguje na data z jednotky, která jsou ve tvaru „L“ číslo režimu svitu diod zakončené středníkem. Režimy svitu obsahují i různé druhy blikání realizované ve vektoru přerušení časovače v režimu CTC. Pomocí časovače jsme zabránili zbytečnému zpomalení programu v čekacích smyčkách. Program do MCU ATmega8 byl napsán v prostředí „AVR Studio 4“, které volně poskytuje výrobce pro rodinu 8b AVR mikroprocesorů.

6 POHONNÝ SUBSYSTÉM

Pohonný subsystém letounu je sestaven ze tří BLDC motorů „KD A22-20L“, umístěných na koncích ramen konstrukce. Motory mají maximální výkon 275W, téměř 12 000ot/s a jsou řízeny prostřednictvím „HK-SS50A“ regulátorů s maximálním proudem 40A. Do pohonného subsystému patří i servomotor „GWS Naro Super 19g“, který slouží k náklonu zadního motoru. Náklonem motoru se kompenzuje otáčivý pohyb, který vzniká rotací všech motorů stejným směrem. Natočením motoru proti směru rotace se vytvoří síla působící proti silám, které mají snahu rotovat s letounem.

6.1 Regulátory

Regulátor „HK-SS50A“ (Obrázek 14) (RC Model Shop) byl zvolen zejména díky své dostupnosti. Prakticky jediný požadavek, který byl na regulátor při vývoji kladen, je jeho proudová zatížitelnost, která musí být větší, než je maximální proud do motoru (>25A). Regulátory jsou napájeny přímo z baterií přes měřící desku a ovládají se PWM signálem s frekvencí 50Hz. Regulace je bez-senzorová.



Obrázek 20: Regulátor

6.2 Motory

Motory (Obrázek 15) (Hobby King) jsou navrženy vzhledem k předpokládané hmotnosti letounu, která byla odhadnuta na 2,5kg i s externí zátěží (fotoaparát). Každý motor má v kombinaci s vrtulí 9x4,7, tažnou sílu zhruba 850g. Je možné použít i jiné rozměry vrtule, ale musí být zajištěn maximální proud do motoru 25A!

Obrázek 21: BLDC Motor

V grafu 4 je vidět závislost výkonu všech motorů s rostoucími otáčkami. Hodnoty pro graf byly naměřeny s vrtulemi 9x5.

Graf 4: Závislost výkonu na otáčkách

7 SOFTWARE PRO OVLÁDACÍ POČÍTAČ

Software na ovládací počítač původně vůbec neměl být součástí této práce. Okolnosti nás však přinutili na začátku jednoduchý program vytvořit. Program je tvořen v prostředí LabVIEW. Toto prostředí vyniká svojí jednoduchostí a univerzálností. Program je tvořen tzv. grafickým programováním, což znamená, že se jednotlivé komponenty propojují cestami, které jsou různého typu. Typ cesty odpovídá datovému typu hodnoty, pro niž je cesta určena. Graficky jsou cesty rozlišeny barevně, takže lze na první pohled určit, o jaký typ dat jde.

Program v LabVIEW má svoji programovou stránku a pak svoji vizuální stránku která slouží uživateli. Pro vizuální i programovou stránku je zde mnoho komponent, které je možno využít. Jsou zde například pro knihovny pro práci s herním zařízením, které je používáno pro ovládání modelu. Lze tak dosáhnout velmi příjemného rozhraní, při nízkých nárocích na složitost programu. Program napsaný jedním z tradičních jazyků, by při stejných nárocích na kvalitu, byl obsahově na samostatnou dlouhodobou maturitní práci.

Pro práci s modelem byly vytvořeny dvě verze programu, liší se svým ovládáním a využitím. V obou programech jsou barevně rozlišeny všechny ukazatele odpovídající jednotlivým motorů.

7.1 Ovládací software

Prvním programem je tzv. Ovládací program, jeho určení je přímé a pouhé ovládání modelu. Program obsahuje indikátory aktuálního stavu, výběr režimu ovládaní a nastavení citlivost. Program je rozdělen na čtyři hlavní smyčky, které běží paralelně. Po spuštění se otevře sériová linka, kde je připojen model, případná chyba je zaznamenána. Další podmínkou pro úspěšné spuštění je přítomnost zvoleného joysticku. Joystick jako jediná možnost pro ovládání je také otevřen ihned po spuštění. V případě, že spuštění proběhne úspěšně, jsou spuštěny všechny čtyři smyčky.

Program je rozdělen do jednotlivých smyček následovně:

1) Odesílání statických parametrů, bezpečnostní sekvence, stisků kláves a změn hodnot.
2) Čtení z herního zařízení, odesílání změn, ovládání aplikace herním zařízením.

3) Příjem hodnot z modelu, třídění a zobrazení ve vizuální stránce.
4) Odhad kapacity baterie, potřebné výpočty, správa chybových hlášení.

Jednotlivé smyčky běží paralelně a v případě, že dojde k události neslučitelné s dalším provozem (ztráta herního zařízení, neexistující vybraný port), je aplikace ukončena.

Protokol je optimalizován pro pouhé ovládání. Některé parametry potřebné pro nastavování se v rámci šetření času na komunikační lince neposílají.

7.2 Nastavovací software

V případě připojení k modelu nastavovacím softwarem, řídící jednotka tuto situaci rozezná a optimalizuje protokol pro nastavování parametrů. V případě nastavování se parametry začnou vysílat, a tak je možné použít tyto data pro diagnostiku nebo jako pomůcku pro nastavení parametrů.

Programová stránka je obdobná jako u ovládacího softwaru. Vizualizační stránka je však úplně odlišná. Je rozdělena na 4 karty, jejichž přepínání je možné i tlačítky na herním zařízení.

Na jednotlivých kartách jsou rozmístěny prvky potřebné pro nastavování parametrů. První karta se nejvíce blíží ovládacímu softwaru, i když je ochuzena o některé prvky zbytečné pro nastavování (teplota, proudy, zbývající kapacita). Druhá stránka byla využívána pro měření charakteristiky vibrací v závislosti na otáčkách motorů. S vibracemi byl veliký problém, který byl z velké části vyřešen. Třetí a čtvrtá stránka se věnuje možnostmi nastavení. Na každé z nich jsou v horní části nastavovací prvky, jež umožňují „za běhu“ nastavit jednotlivé parametry regulace. Pod nimi je umístěn graf znázorňující jednotlivé složky PID a jejich součet v závislosti na čase. Tento graf se ukázal jako praktický při diagnostice, kde kmitání vzniká. Posledním prvkem na těchto dvou stránek je graf ukazující aktuální výchylku v dané ose.

8 KOMUNIKAČNÍ PROTOKOLY

Elektronika modelu se skládá z několika samostatných celků, je třeba mezi nimi přenášet data. Pro tyto účely jsou využívány dva druhy komunikačních protokolů. Oba protokoly jsou založeny na sériovém rozhraní USART a přenos probíhá téměř výhradně pomocí tisknutelných znaků ASCII tabulky. Jediná výjimka je na přenosové lince mezi počítačem a modelem. Protokol založený na ASCII tabulce byl vybrán, kvůli tzv. „Terminal friendly“ vlastnosti. To znamená, že v případě problému může být jednoduše přečten nebo odeslán z terminálu v počítači a tak se mnohem jednodušeji diagnostikuje problém v komunikaci.

8.1 AHRS protokol

První protokol je založen na standartu mezi senzory AHRS. V tomto protokolu dochází k pravidelnému vysílání celého bloku. Blok je uvozen znakem dolaru a končí znakem mřížky. Tyto znaky jsou velmi důležité, protože je třeba udržet synchronizaci údajů i při částečné ztrátě dat. Data totiž nejsou uvozeny jednoznačným znakem, ale jsou pouze oddělována znakem čárky. Takže v případě ztráty synchronizace by došlo k chybné interpretaci jako jiná hodnota.

Data z AHRS chodí pravidelně padesát-krát za sekundu. V procesoru jsou nejdříve přijata do znakových polí a poté převedena na číselnou hodnotu. Hodnoty, u nichž, je vyžadována přesnost na desetinná místa jsou vysílány jako deseti, popř. sto-násobky, čímž je vyřešen problém s detekcí a zpracováním desetinné čárky.

Tvar posílaného bloku:

```
$,roll,pitch,yaw,gz,#
```

Název proměnné	Popis	Poznámka
roll	Úhel levopravého náklonu	Odesílán 100 násobek
pitch	Úhel předozadního náklonu	Odesílán 100 násobek
yaw	Úhel natočení vůči severu	
gz	Rychlost otáčení vůči středové ose	

Tabulka 5: Proměnné protokolu AHRS

8.2 Komunikační protokol mezi počítačem a řídící jednotkou

Druhý protokol je využíván mezi řídící jednotkou a počítačem, a poměrně nově je taky přípraven pro komunikaci mezi řídící a měřící jednotkou. Rozdílem od přechozího protokolu je, že se neodesílá pravidelně. Jednotlivé proměnné respektive příkazy se odesílají se svým jedinečným identifikačním znakem. Jednotka nebo počítač pak rozliší typ proměnné, a podle toho s ní pracuje.

Identifikační znaky jsou složeny ze znaků malé a velké anglické abecedy (Tabulka 6), poté jsou následovány dvojtečkou a předávanou hodnotou. Hodnota se může skládat ze šesti znaků, což je rozsah od -99 999 do 999 999.

Jediné tři znaky, které nejsou součástí tisknutelných znaků, jsou znaky pro správu komunikace. Tyto znaky se odesílají pravidelně, a slouží pro kontoru vzájemné ho spojení. Kromě této funkce, slouží tyto znaky pro rozpoznání typu programu, kterým připojení k modelu probíhá. Detekce je nutná pro optimalizaci zpětně posílaných dat.

Znak	Popis	Poznámka	Jednotky
0x02	Nastavovací software	Odesílá se pravidelně	
0x03	Odpojení		
0x06	Ovládací software	Odesílá se pravidelně	
D	D-složka roll regulace	Odesílá se násobek 1000	---
E	Nouzové vypnutí	Vypnutí a zablokování	---
F	Fuse gain roll regulace		---
I	I-složka roll regulace	Odesílá se násobek 1000	---
L	P-složka pitch regulace	Odesílá se násobek 1000	---
M	I-složka pitch regulace	Odesílá se násobek 1000	---
N	D-složka pitch regulace	Odesílá se násobek 1000	---
O	Fuse gain pitch regulace		---
P	P-složka roll regulace	Odesílá se násobek 1000	---
R	Rotace kolem osy		stupně/s
X	Osa X	Násobek 10	Stupně
Y	Osa Y	Násobek 10	Stupně
m	Režim ovládání	0=Hover, 1=Manual	---

Tabulka 6: Ovládací příkazy z počítače

Komunikace směrem z řídící jednotky probíhá na podobném principu. Identifikační znak je nahrazen identifikační sekvencí znaků. Je tak možné udělat názvy přehlednější, přitom je potřeba dodržet přiměřený poměr srozumitelnost vůči vytížení linky. V naší aplikaci se pro identifikaci používá dvou nebo třípísmenné sekvence. Jednotlivé použité kombinace jsou zobrazeny v tabulce (Tabulka 7).

Sekvence	Popis	Poznámka
MD	ID nového režimu	
ER	Ohlášení chybového stavu	
PH	Úhel předozadního náklonu	100 násobek
RL	Úhel levopravého náklonu	100 násobek
GZ	Rychlost otáčení vůči středové ose	
ANG	Úhel natočení vůči severu	
RP	Aktuální hodnota složky P roll regulace	Pouze nastavovací program
RI	Aktuální hodnota složky I roll regulace	Pouze nastavovací program
RD	Aktuální hodnota složky D roll regulace	Pouze nastavovací program
PP	Aktuální hodnota složky P pitch regulace	Pouze nastavovací program
PI	Aktuální hodnota složky I pitch regulace	Pouze nastavovací program
PD	Aktuální hodnota složky D pitch regulace	Pouze nastavovací program
M1	Hodnota výkomu levého motoru	
M2	Hodnota výkomu pravého motoru	
M3	Hodnota výkomu zadního motoru	

Tabulka 7: Přenos zpět do počítače

9 ZÁVĚR

V rámci této dlouhodobé maturitní práce byla navržena a realizována řídící část projektu Tricopter. Jednotlivé zvolené prvky byly osazeny na mechaniku modelu. Model byl úspěšně oživen.

Projekt byl realizován pro co největší rozšiřitelnost. Při každém kroku vývoje jsme se snažili připravit veškeré věci, které by mohli být v budoucnu potřeba. Během vývoje se objevovaly různé problémy, často bylo mnoho kandidátů na řešení. Nalezení řešení obvykle předcházelo mnoho testů a hledání příčin problémů. Obvykle se pouze jedno řešení ukázalo jako využitelné.

Vývoj probíhal průběžně ve volném čase přibližně od konce školního roku 2009/2010, začali jsme studiem potřebných znalostí. Praktická realizace s osazeným modelem probíhá od října roku 2010. Během práce jsme narazili na řadu problému, řešení nám dodnes zabralo přibližně 500 hodin práce ve dvou lidech. Hodnota potřebných součástí se pohybuje přibližně okolo částky 11 000 korun.

Naším dalším cílem bude nalezení vhodného nastavení PID regulace, která je nepostradatelná pro automatickou stabilizaci. Dalším postupem je pak osazení fotoaparátem pro focení výškových fotografií s živým přenosem. Okamžitý přenos obrazu umožní ovládání i bez přímé viditelnost.

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM GRAFŮ

SEZNAM PŘÍLOH

POUŽITÁ LITERATURA

Atmel. (nedatováno). Získáno 24. 1 2011, z www.atmel.com: http://www.atmel.com/dyn/resources/prod_documents/doc2486.pdf

ATMEL. (2010). *Atmel Corporation - Home*. Získáno 10. Říjen 2010, z Atmel Corporation: http://www2.atmel.com/

Colton, S. (nedatováno). Získáno 21. 12 2010, z web.mit.edu: http://web.mit.edu/scolton/www/filter.pdf

Dipl. Ing. Alexander Dick. (nedatováno). Získáno 20. 9 2010, z www.alvidi.de: www.alvidi.de/avr32_module.html

Hobby King. (nedatováno). Získáno 5. 10 2010, z www.hobbyking.com: http://www.hobbyking.com/hobbyking/store/uh_viewitem.asp?idproduct=2106

Ježerský, J. (nedatováno). Získáno 3. 11 2010, z www.serva.cz: http://www.serva.cz/rizeni-serva-teorie/

Queue (data structure). (n.d.). Retrieved Březen 1, 2011, from Wikipedia: http://en.wikipedia.org/wiki/Queue_%28data_structure%29

RC Model Shop. (nedatováno). Získáno 5. 10 2010, z www.rcmodelshop.gr: http://www.rcmodelshop.gr/index.php?main_page=product_info&products_id=70&language=en

Sparkfun. (nedatováno). Získáno 12. 8 2010, z www.sparkfun.com: http://www.sparkfun.com/products/9623

Spezial Electronic. (nedatováno). Získáno 20. 9 2010, z www.spezial.cz: http://www.spezial.cz/connectblue/ieee-802-15-4-zigbee-oem-moduly.html

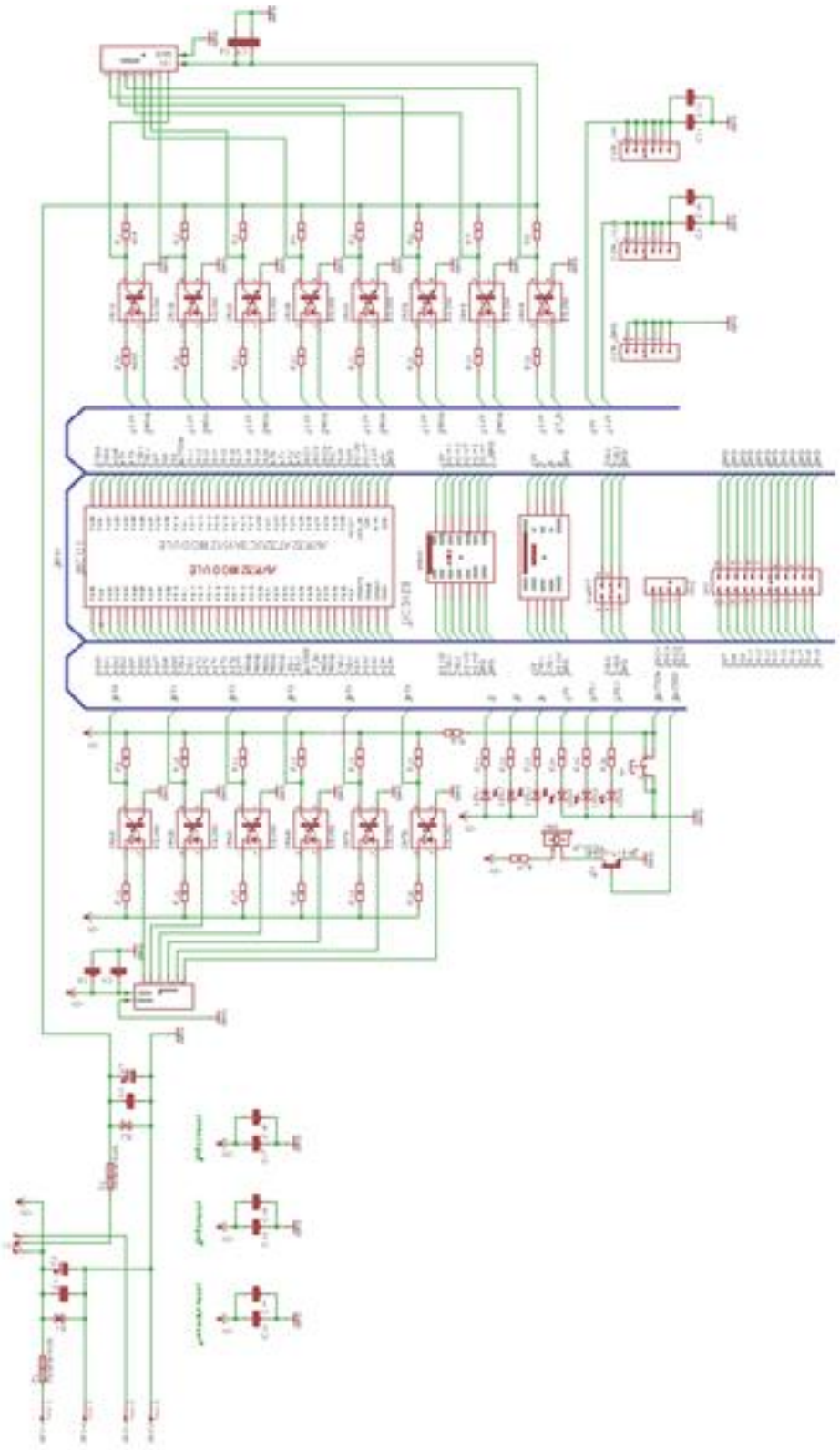
Speziel Electronic. (nedatováno). Získáno 21. 6 2010, z www.spezial.cz: http://www.spezial.cz/connectblue/bluetooth-seriove-rs232-oem-moduly.html

Tourning machine, Pitch, Roll and Yaw. (nedatováno). Získáno 9. Březen 2011, z http://www.touringmachine.com/Articles/aircraft/6/

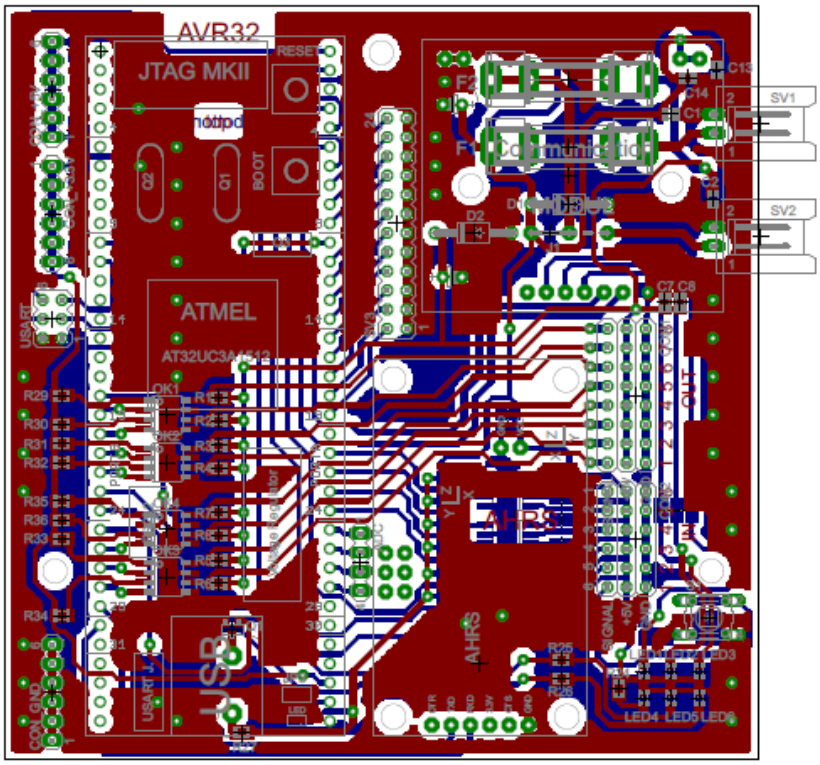
Winkler, Z. (11. 10 2005). Získáno 7. 12 2010, z Robotika.cz: http://robotika.cz/guide/filtering/en

PŘÍLOHY

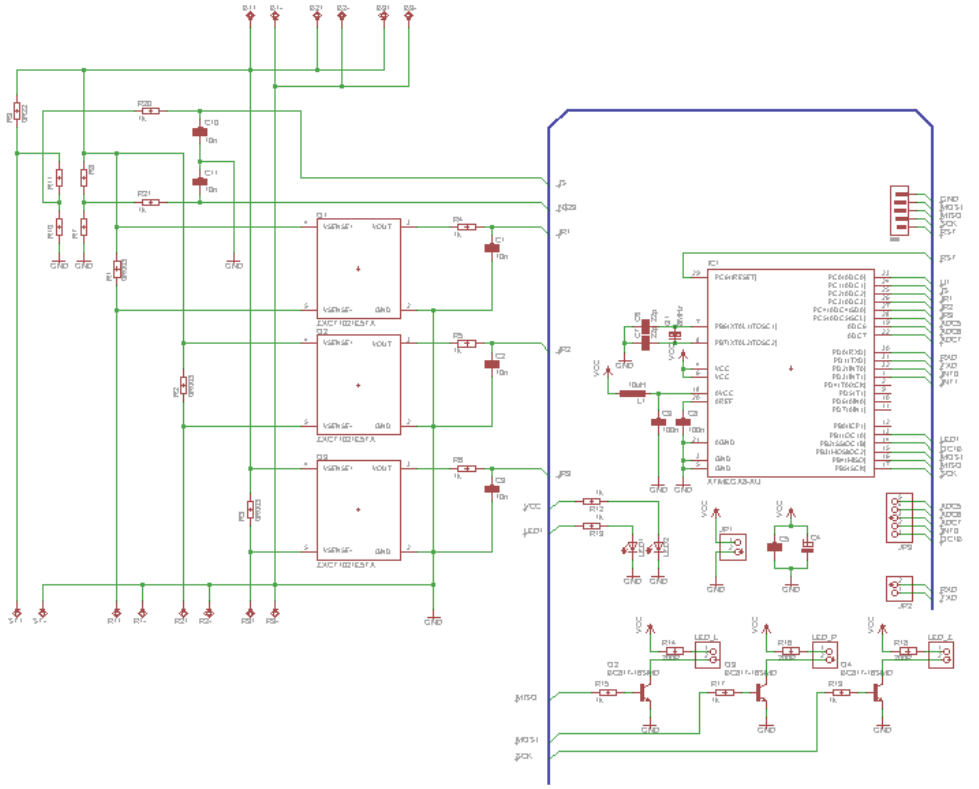
1. SCHÉMA ŘÍDÍCÍ JEDNOTKY



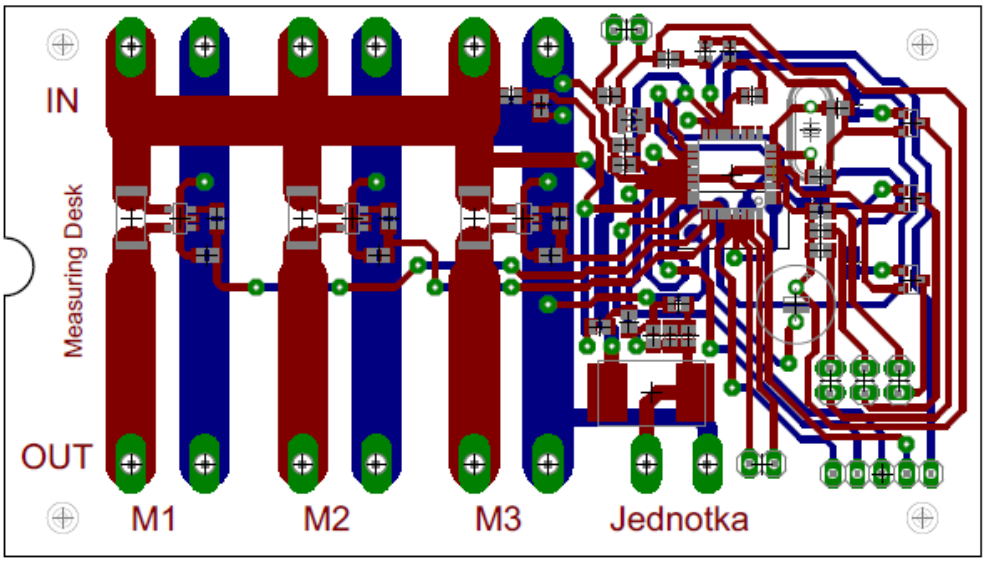
2. DPS ŘÍDÍCÍ JEDNOTKY



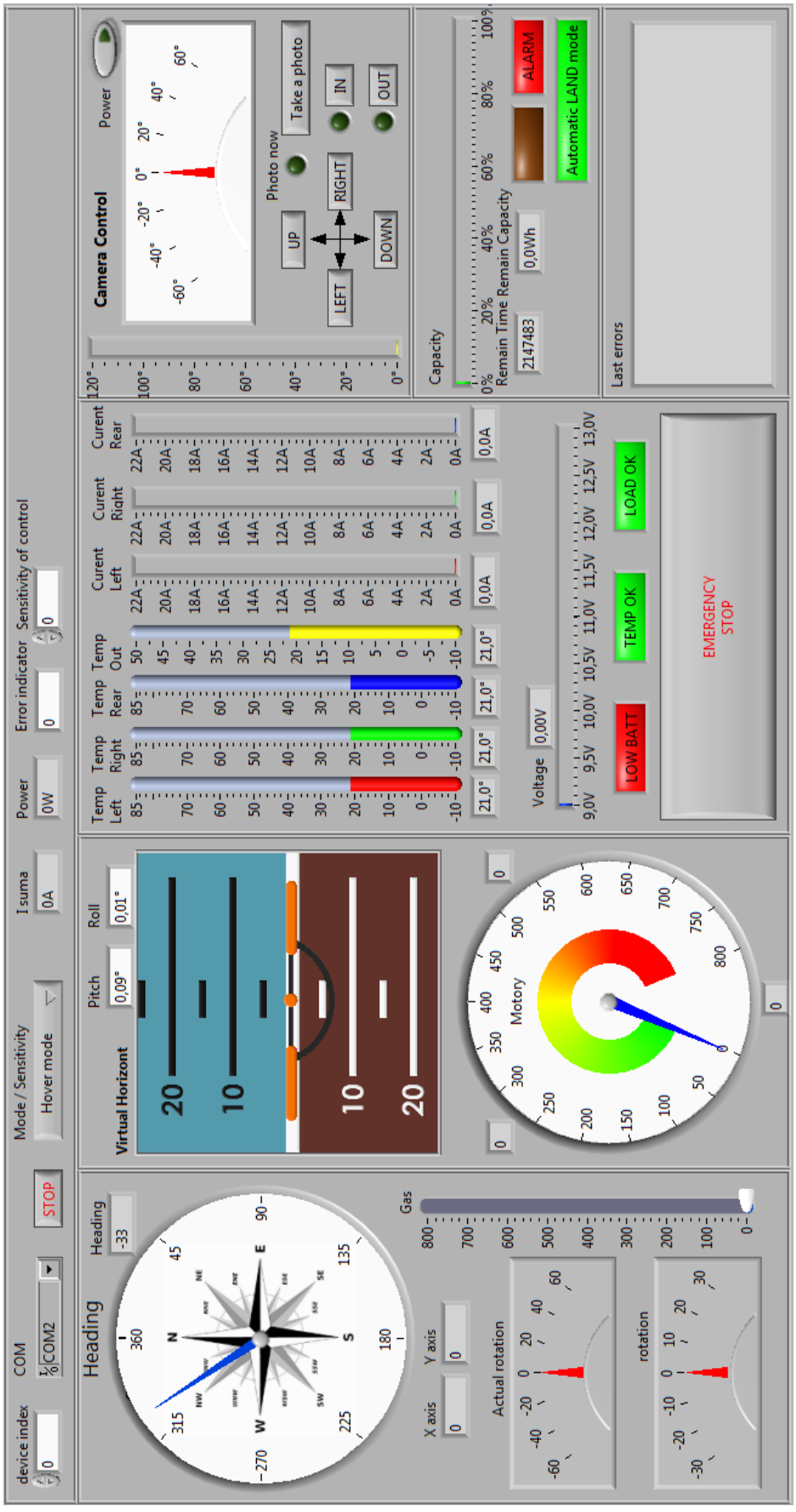
3. SCHÉMA MĚŘÍCÍ DESKY



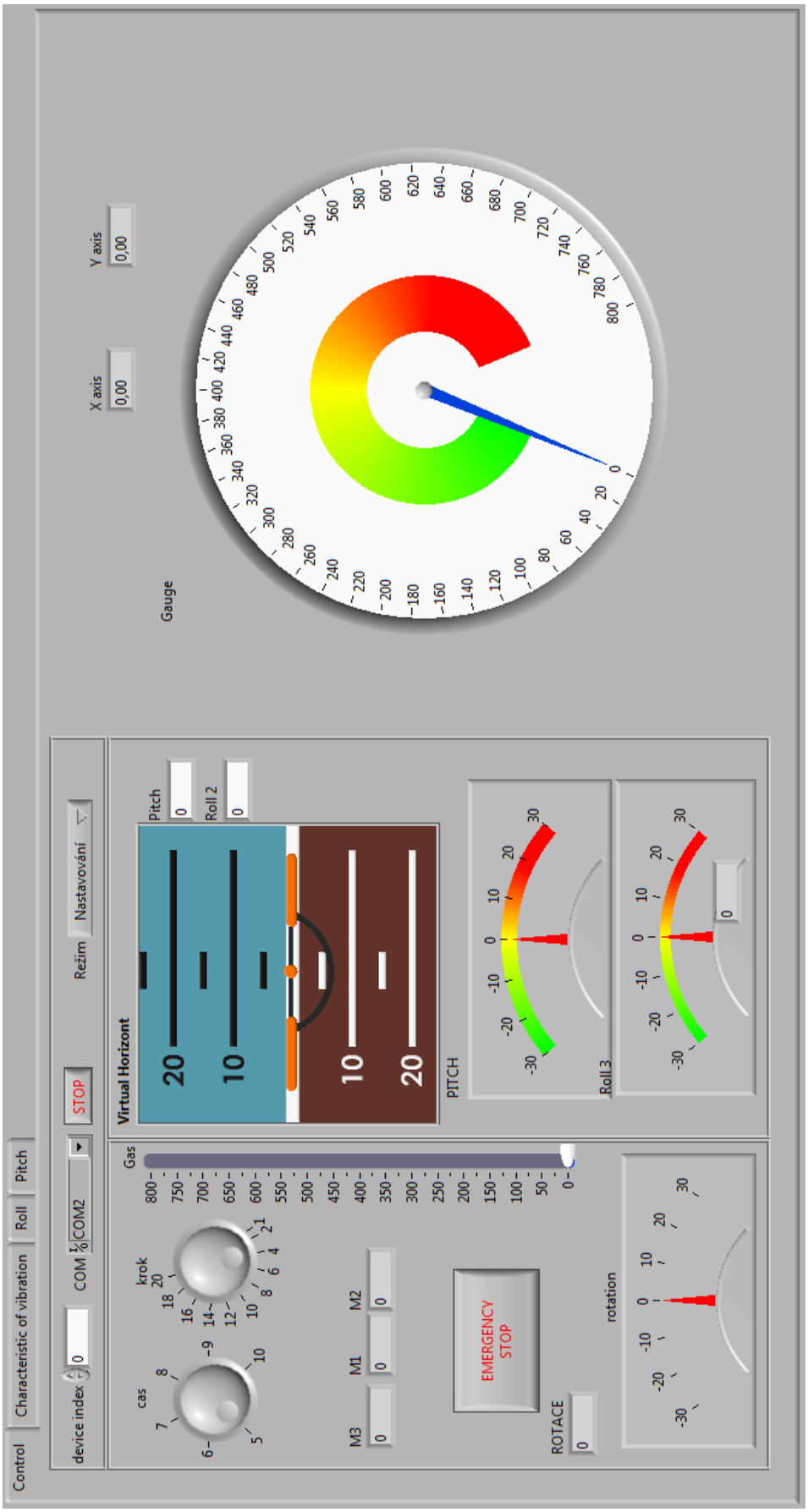
4. DPS MĚŘÍCÍ DESKY

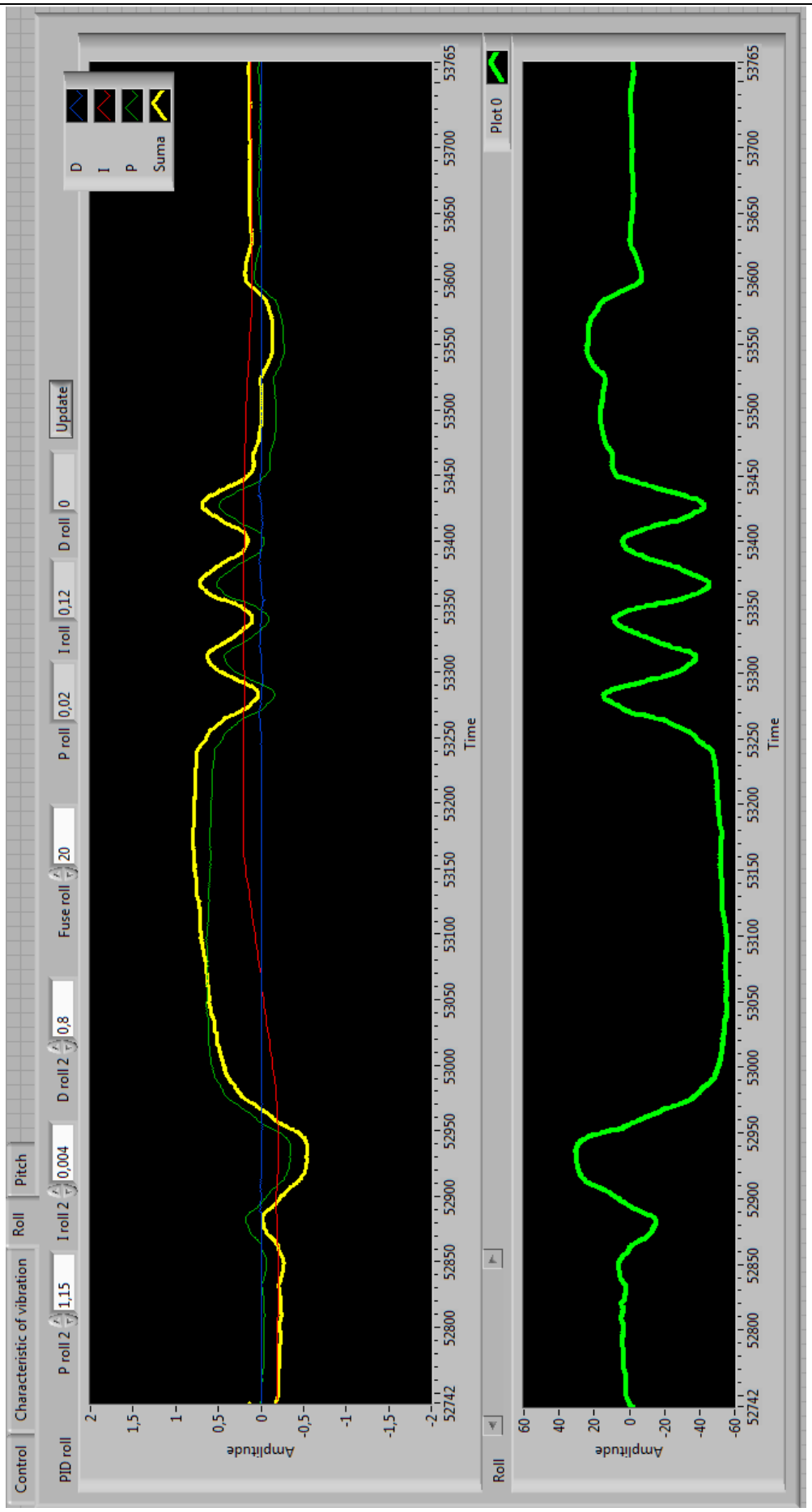


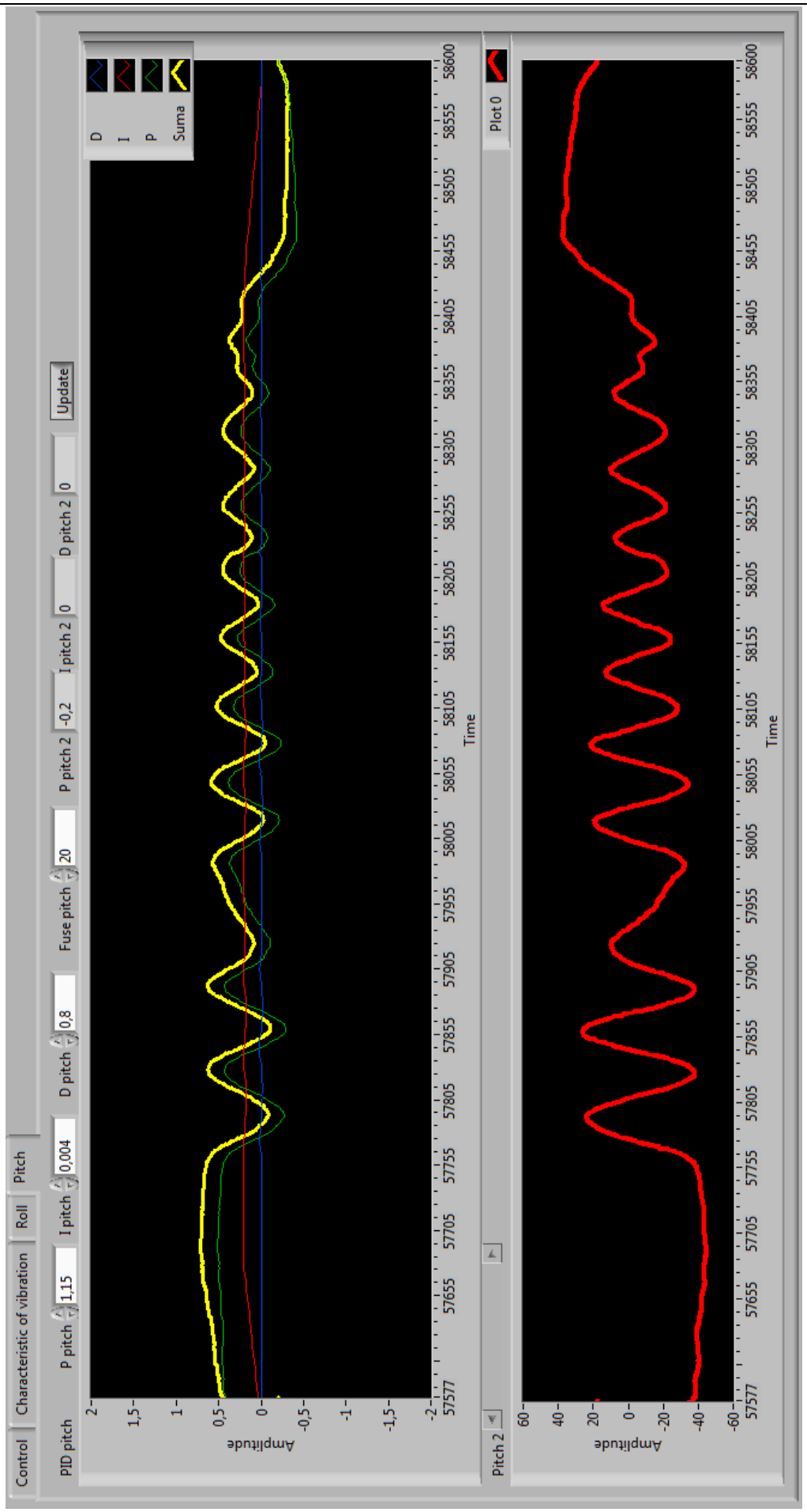
5. VIZUÁLNÍ STRÁNKA OVLÁDACÍHO PROGRAMU



6. VIZUÁLNÍ STRÁNKA NASTAVOVACÍHO PROGRAMU




Control
Characteristic of vibration
Roll
Pitch
PID roll
P roll 2 1,15
I roll 2 0,004
D roll 2 0,8
Fuse roll 20
P roll 0,02
I roll 0,12
D roll 0
Update
D
I
P
Suma
Amplitude
Time
Plot 0
Roll


Control
Characteristic of vibration
Roll
Pitch
PID pitch
P pitch 1,15
I pitch 0,004
D pitch 0,8
Fuse pitch 20
P pitch 2 -0,2
I pitch 2 0
D pitch 2 0
Update
D
I
P
Suma
Amplitude
Time
Plot 0
Pitch 2

7. FOTODOKUMENTACE