



## **Středoškolská technika 2023**

**Setkání a prezentace prací středoškolských studentů na ČVUT**

# **AUTONOMNÍ ŘÍZENÍ DRONU S OBRAZOVÝM SNÍMÁNÍM**

**Daniel Smutný**

**Střední průmyslová škola strojnická a Střední odborná škola profesora Švejcara, Plzeň,  
Klatovská 109, 301 00**

# STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18 Informatika

## Autonomní řízení dronu s obrazovým snímáním

### Autonomous drone with robot vision

**Autor:** Daniel Smutný  
**Škola:** Střední průmyslová škola strojnická a Střední odborná škola profesora Švejcara, Plzeň, Klatovská 109  
**Kraj:** Plzeňský kraj  
**Konzultant:** Ing. Petr Hlávka

Plzeň 2022/2023

# Prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupnění této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Plzni dne 18. 3. 2023 .....

Daniel Smutný

## **Poděkování**

Tímto bych chtěl poděkovat panu Ing. Petru Hlávkoví za podporu při zhotovování projektu. Dále také zástupkyni ředitelky Ing. Petře Maškové za umožnění testování projektu v tělocvičně školy. Rád bych též poděkoval Michalovi Smutnému za konzultaci a poskytnutí prostorů a vybavení pro úspěšné zhotovení projektu.

## **Anotace**

Tato práce se zabývá autonomním řízením dronu za pomoci obrazového snímání. Dron je schopen autonomního letu, během kterého vyhledává barevné kódy, nad které se poté umí autonomně navigovat a následně na ně přistávat. Barevné kódy jsou rozpoznávány za pomoci obrazového snímání kamery, která je na dronu zavěšena. Řízení dronu je řešeno pomocí naprogramovaných mikrokontrolérů Multiwii a arduino. Informace z mikrokontrolérů se následně odesílají do řídicí desky dronu.

## **Klíčová slova**

autonomní řízení, dron, arduino, obrazové snímání

## **Annotation**

This thesis is focusing on autonomous control of a drone using robot vision. The drone is capable of autonomous flight, during which it searches for color codes to which it can navigate itself and subsequently land on them. The color codes are recognized using the Robot vision of a camera that is mounted on the drone. The control of the drone is done by programmed microcontrollers Multiwii and arduino. Information from the microcontrollers is then sent to the drone's flight controller.

## **Keywords**

autonomous controlling, drone, arduino, Robot vision

## Obsah:

1	Úvod.....	7
2	Hardware.....	8
2.1	Řídící deska SpeedyBee F7 V3.....	8
2.2	Přijímač FrSky V8FR-II HV .....	9
2.3	Vysílač.....	10
2.4	Kamera Pixy.....	11
2.4.1	Obrazové snímání .....	11
2.5	Gimbal T3D III .....	12
2.6	Mikrokontroléry Arduino pro miny a MultiWii SE v0.2.....	13
2.6.1	Arduino pro miny.....	13
2.6.2	MultiWii SE v0.2.....	14
2.7	Díly vytisknuté na 3D tiskárně.....	15
2.7.1	Přistávací nožičky .....	15
2.7.2	Plocha pro připevnění Mikrokontrolérů .....	15
2.7.3	Držák kamery.....	16
2.8	Blokové schéma systému .....	17
3	Software.....	18
3.1	Program pro řízení dronu (MultiWii).....	18
3.2	Program v arduino pro miny .....	19
3.3	INAV.....	20
4	Závěr .....	21
5	Použitá literatura .....	22
6	Seznam obrázků a tabulek .....	23
7	Příloha 1: PROGRAM MultiWii SE v0.2 .....	24

# 1 ÚVOD

Projekt byl zaměřen hlavně na použití získaných dovedností během studia na střední škole, a to především z oboru Informační a komunikační technologie (IKT) a Mechatronika (MRA). Projekt byl též zaměřen na dovednost nacházení vhodných zařízení pro jeho zhotovení a jejich následného propojení do jednoho systému.

Prvotním úkolem projektu bylo vytvoření konceptu a následné vytipování dílů, které splňují potřebné specifikace pro projekt. Práce též zahrnovala vytváření dílů v AutoCAD Inventor a jejich tisk na 3D tiskárně. Hlavní náplní projektu bylo však vytváření programu řízení dronu pro mikrokontrolér MultiWii psaný v programu Arduino IDE a jeho testování.

Mikrokontrolér MultiWii zajišťuje v projektu vlastní řízení dronu. Jeho program má za účel vyhodnocovat souřadnice z kamery a následně je převádět na hodnoty, které ovládají v řídicí desce výchylky dronu. Tyto hodnoty jsou odesílány do mikrokontroléru Arduino pro miny, které následně zajišťuje převod pulzně šířkové modulace (PWM) na SBus.

## 2 HARDWARE

### 2.1 Řídicí deska SpeedyBee F7 V3

Jedná se o řídicí desku určenou hlavně pro ovládání vícerotorových helikoptér. Její účel je monitorování letu a stabilizace dronu. Na této desce se dají nastavovat parametry letu jako jsou například průběhy a maximální rychlost rotace dronu kolem daných os, limity náklonu dronu a omezení síly motorů.

Tato deska je programovatelná, tudíž se na ní dají měnit všechny prvky letu od maximálních výchylek po maximální sílu, kterou mohou motory vyvinout.

Též se na ní dají propojovat zařízení jako GPS, magnetometr, barometr, LED světla a další (Tyto však v práci nejsou použity). Deska má též tu výhodu, že se její vlastnosti dají nastavovat na dálku pomocí bluetooth, což se v bylo pro projekt výhodou z důvodu neustálé potřeby úpravy vlastností dronu, v závislosti na měnění vlastností ovládací desky.

MCU	STM32F722
IMU(Gyro)	BMI270
USB Port Type	Type-C
Barometer	BMP280
OSD Chip	AT7456E chip
Power Input	3S - 6S Lipo
SmartPort	Use any TX pad of UART for the SmartPort feature.
Supported Flight Controller Firmware	BetaFlight(Default), EMUFlight, INAV
Firmware Target Name	SPEEDYBEEF7V3
Mounting	30.5 x 30.5mm( 4mm hole diameter)
Dimension	41(L) x 38(W) x 8.1(H)mm
Weight	10.7g
SmartPort	Use any TX pad of UART for the SmartPort feature.
Supported Flight Controller Firmware	BetaFlight(Default), EMUFlight, INAV
Firmware Target Name	SPEEDYBEEF7V3

Tab. 1: Přehled parametrů řídicí desky SpeedyBee F7 V3



Obr. 1: Řídicí deska SpeedyBee F7 V3



## 2.2 Přijímač FrSky V8FR-II HV

Jedná se o zařízení, které na dálku přijímá signál z vysílače a mění je na signál odesílaný pomocí kabelů.

Přijímač použitý v projektu má 8 kanálů a signál vysílá pomocí pulzně šířkové modulace (PWM). Pracuje na kmitočtu 2.4 GHz. Je stavěný tak, aby měl vysokou stabilitu signálu a měl velkou odolnost oproti rušení



Obr. 2: Přijímač FrSky V8FR-II HV

## 2.3 Vysílač

Je zařízení, které slouží k ovládání modelu. Vysílače mohou být v podobě pákových či volantových ovladačů.

U letadel a dronů se jedná o pákové ovladače. To znamená, že ovladač má dvě páky, které ovládají osy letu a plyn. Také se na ovladači nachází další nastavovací a ovládací prvky, jako jsou například tlačítka a potenciometry. Podle nastavení, v jakých polohách se tyto ovládací prvky nachází, vysílá ovladač určité signály.



Obr. 3: Vysílač OPTIC 6

## 2.4 Kamera Pixy

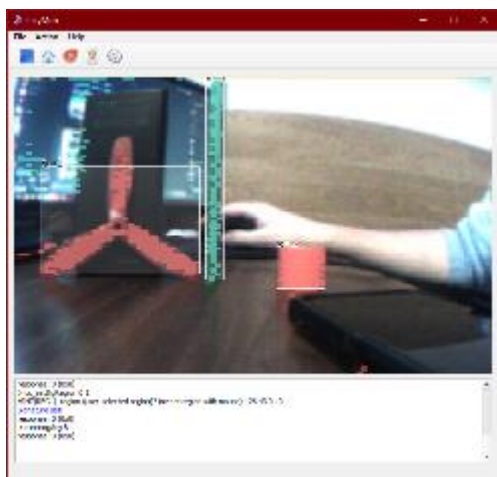
Jedná se o open-source kameru, která v sobě má již program pro obrazové snímání, jenž se dá velmi dobře nastavovat za pomoci aplikace, která je pro něj přímo vytvořená pod názvem PixyMon. Podle nastavení kamera snímá uživatelem vybrané barvy. Také dokáže z více barev tvořit barevné kódy. Další výhodou je dobrá kompatibilita s mikrokontroléry Arduino. Kameru jsem zvolil také pro velice rychlé snímání oproti jiným řešením. Zároveň se osvědčila lépe pro účely programu, než tomu bylo při prvotních pokusech s kódem vytvořeným pouze mnou.



Obr. 4: Kamera Pixy

### 2.4.1 Obrazové snímání

Obrazové snímání je funkce, při které program postupně projíždí obraz zachycený na kameře a hledá pixely s barvou, kterou se snažíme najít. Program je nastavený tak, aby vyhledával skupiny těchto pixelů, a poté je označil do bloku. Program zaznamenává souřadnice tohoto bloku a odesílá je do MultiWii, kde jsou dále zpracovávány pro řízení dronu.



Obr. 5: Ukázka obrazového snímání

## 2.5 Gimbal T3D III

Gimbal je zařízení, jež se využívá pro stabilizaci natáčení a focení, aby na kameře nebyly vidět otřesy nebo náklony. Toto zařízení drží stabilitu ve třech osách tak, aby kamera zůstala ve vodorovné poloze. To zajišťují 3 motory, které jsou ovládané programem a gyroskopem, jenž tyto otřesy a náklony vyhodnocuje. Na dronu je nainstalován Gimbal T3D III, jehož součástí je také aplikaci pro jeho nastavování. Zajišťuje udržení kamery ve vodorovné poloze i přes náklon dronu.

Rozsah řízení	
Kolem svislé osy (PAN)	$\pm 330^\circ$
Klopení (TILT)	$-135^\circ \sim +45^\circ$
Klonění (ROLL)	$\pm 48^\circ$
Podpora přijímačů	S-Bus / PPM / DSM
Pracovní teplota	$-20^\circ\text{C} \sim +50^\circ\text{C}$
Maxi. rozměry (LWH)	60 mm * 75 mm * 100mm

Tab. 2: Přehled parametrů Gimbalu T3D III



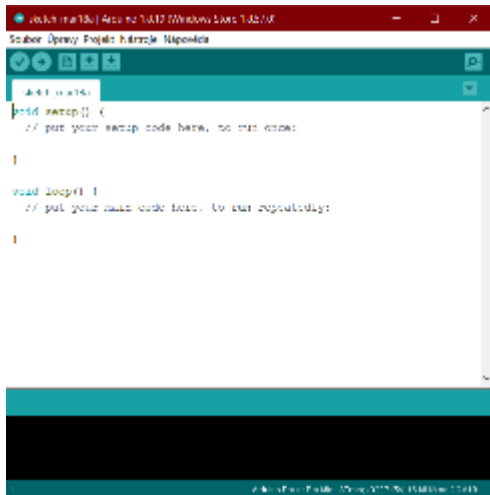
Obr. 6: Gimbal T3D III

## 2.6 Mikrokontroléry Arduino pro miny a MultiWii SE v0.2

U obou desek se jedná o open-source programovatelnou desku. Všechny tyto desky fungují za pomoci čipů ATmega. Pro jejich programování je využíván jazyk Wiring což je odnož c/c++.

Výhodou arduino je jeho nízká cena a jeho velká popularita díky které se dá o něm najít velmi velké množství informací a inspirovat se z mnoha programů jiných lidí.

K jeho programování se využívá program Arduino IED.



Obr. 7: Arduino IED

### 2.6.1 Arduino pro miny

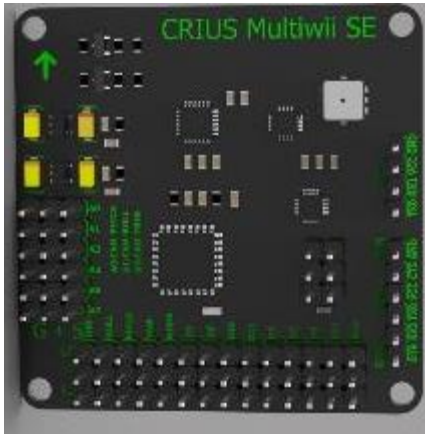
V Projektu využíváme Arduino pro miny hlavně kvůli jeho malé velikosti a váze. Také kvůli nízké ceně. Toto arduino slouží k převodu z pulzně šířkové modulace(PWM) na SBus.



Obr. 8: Arduino Arduino pro miny

## 2.6.2 MultiWii SE v0.2

MultiWii bylo pro projekt vybráno, protože má vyhovující rozložení vstupů a výstupů a jejich pojmenování, což zjednodušuje orientaci v konstrukci dronu. Ačkoliv se v mnoha případech využívá MultiWii jako řídicí jednotka, v případě tohoto projektu byla deska naprogramována namísto softwaru řídicí jednotky softwarem mým, který slouží pro vyhodnocování dat a zasílání pokynů vlastní řídicí jednotce dronu.



Obr. 9: MultiWii SE v0.2

## 2.7 Díly vytisknuté na 3D tiskárně

Všechny vytisknuté díly jsou vytvořeny pomocí programu AutoCAD Inventor. Použita byla 3D tiskárna TRILAB DeltiQ 2.

### 2.7.1 Přistávací nožičky

Jedná se o dva identické díly, které jsou vyztuženy uhlíkovými páskami. Tyto dva díly jsou k sobě připevněny pomocí uhlíkových tyček. Ke dronu jsou přichyceny pomocí šroubů.

Nožičky neplní svou funkci pouze při přistávání, jejich funkce je především významná pro upevnění gimbalu na tělo dronu a ochranu gimbalu při letu i přistání. Při testování se po nouzovém vynutí motorů tento systém osvědčil jako vysoce odolný a vydržel kromě jiného i volný pád z výšky dvou metrů bez poškození.



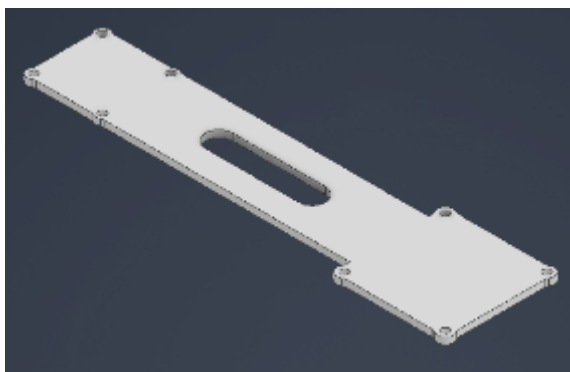
Obr. 10: Přistávací nožičky v AutoCAD Inventor



Obr. 11: Přistávací nožičky hotové

### 2.7.2 Plocha pro připevnění Mikrokontrolérů

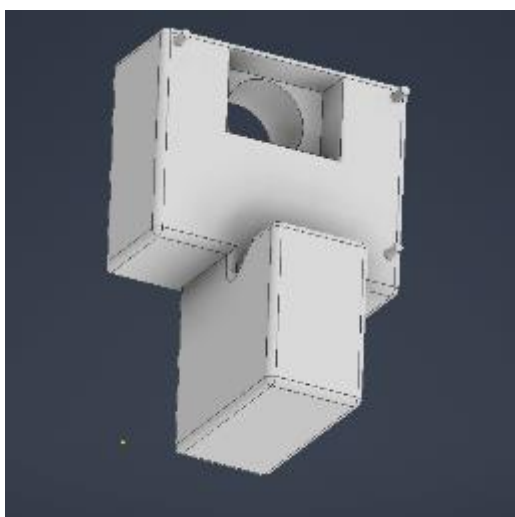
V průběhu testování a vytváření dronu jsem narazil na potřebu většího prostoru v rámci snadného uchycení a manipulace s použitými komponenty. Z toho důvodu byl tento díl vytvořen.



Obr. 11: Plocha pro připevnění Mikrokontrolérů

### 2.7.3 Držák kamery

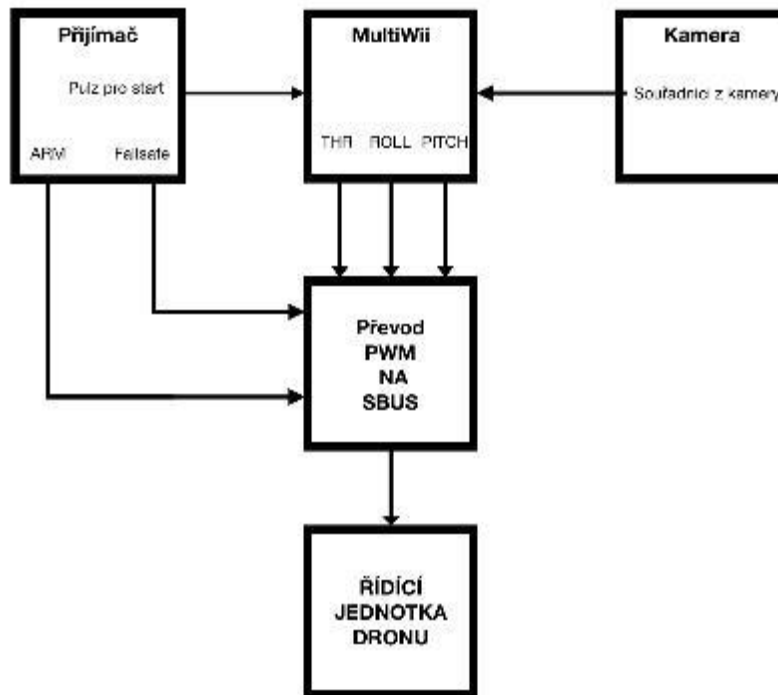
Díl slouží jako držák kamery. Je vytvořený tak, aby přesně nasedal do úchytu na gimbalu a zároveň pevně držel kameru. Návrh tohoto dílu také zohledňuje požadavek na správné vyvážení a zatížení gimbalu.



Obr. 12: Držák kamery



## 2.8 Blokové schéma systému



Obr. 13: blokové schéma

Blokové schéma zobrazuje kompletní zapojení všech hardwarových prvků systému. Dále je na něm také možno vidět, odkud a kam jsou signály odesílány. V blokovém schématu je možnost vidět, že do řídicí desky jdou dva signály přímo z přijímače. Jsou označeny ARM a Failsafe. Signál ARM povoluje a zakazuje otáčení motorů. Pokud přijde signál Failsafe, tak se automaticky zapíná nouzové přistání, kdy jsou na dronu zapnuty určité otáčky, na které dron klesá k zemi. THR je označení pro signál, který ovládá plyn. Signál označený jako ROLL ovládá natáčení dronu. Výchylinky jsou ovládány pomocí signálu označeného PITCH.

## **3 SOFTWARE**

### **3.1 Program pro řízení dronu (MultiWii)**

Viz Příloha 1

Program zajišťuje automatické vyhodnocování informací o pohybu a autonomní řízení dronu s jeho navedením na cíl.

Program vyhodnocuje na ose  $x$ ,  $y$  vzdálenost dronu od kamerou vyhledaného barevného kódu. Při pohybu od cíle jsou ovládány výchylky dronu v závislosti na vzdálenosti. Při pohybu k cíli program snižuje výchylky tak, aby se dron k cíli při přibližování pohyboval bezpečnou kontinuální rychlostí před přistáním. Tuto část programu zajišťuje systém vyhodnocování vektorů pohybu.

Zároveň program ovládá řízení výkonu motorů pro vzlet, let i přistání dronu na zvoleném (barevném kódu). Tyto kroky jsou řízeny časově z důvodu nepřesnosti informací přijímaných z barometru.

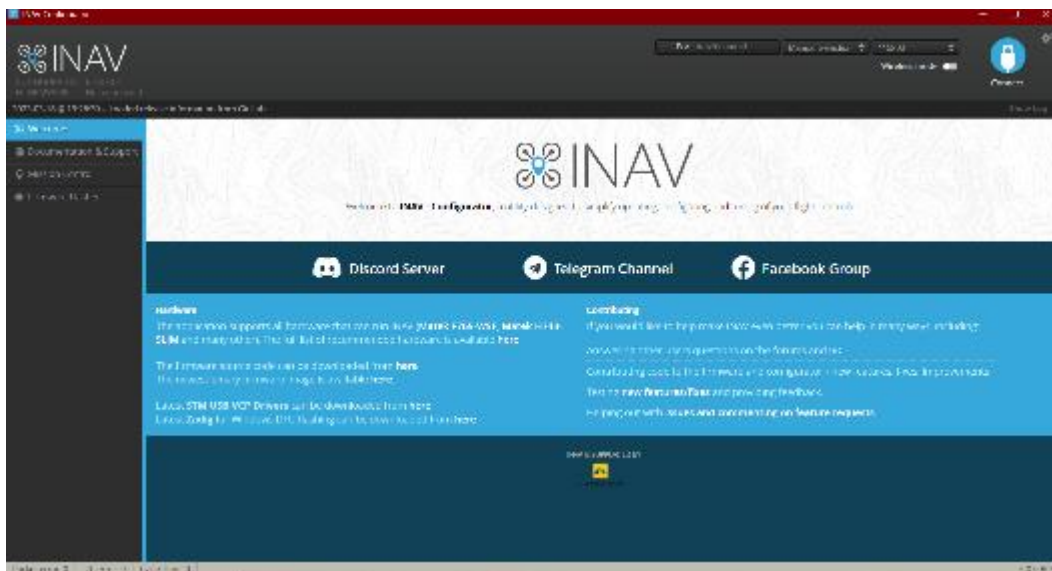
### **3.2 Program v arduino pro miny**

Program slouží k převodu pulzně šířkové modulace (PWM) na SBus. Původně jsem zamýšlel vytvoření vlastního programu, ale po mnoha neúspěšných pokusech o vytvoření plynulého přenosu signálu z PWM na SBus, jsem tedy přešel na program vytvořený uživatelem jménem Ceptimus na YouTube. Program jsem ale upravil, aby vyhovoval tomuto projektu. Použití tohoto programu bylo vynuceno faktem, že přenos signálu PWM již není podporován moderními řídicími deskami, které jsou použity v dronu.

### 3.3 INAV

Jde o open-source program, pomocí kterého lze nastavovat vícerotorové helikoptéry a letadla. Program obsahuje též velmi detailní nastavení všech vlastností řídicí desky tak, aby vyhovovaly uživateli. Slouží též k nahrávání nových verzí softwaru do řídicí desky. Další nastavení, které program zajišťuje jsou módy, jež dron může používat například pro samostatné vyrovnání se do vodorovné polohy, nastavení maximálního náklonu a podobné funkce. Pokud by na řídicí desku byla připojena GPS, uměl by se podle ní řídit a také vracet na místo vzletu (funkce není využita v tomto projektu, je zde pouze uvedena jako příklad pro další možnosti rozvoje této práce).

Tento program byl zvolen mezi jiným použitelnými programy pro jeho lepší vyhodnocovací schopnost stavu dronu při letu, čímž je dosažena vyšší stabilita dronu.



Obr. 14: program INAV

## 4 ZÁVĚR

V úvodu práce jsem si jako cíl vytkl vytvořit autonomní dron řízený za pomoci obrazového snímání. Cíl se mi podařilo splnit, ačkoliv jsem se v průběhu práce musel vypořádat s mnoha překážkami jak v oblasti materiálního zajištění, tak i v oblasti programové.

Mezi nejzávažnější obtíže patřila v programové oblasti zastaralost mnou zamýšleného komunikačního systému mezi řídicí deskou a MultiWii. Tuto nesnáz se mi však podařilo vyřešit. Dalším problémem se ukázalo použití barometru, který neměřil s dostatečnou přesností potřebnou pro bezpečné použití dronu (odchylka v měření představovala až  $\pm 1$  m). Tuto překážku by bylo možné obejít za pomoci velmi přesného laserového odměřování, což by však bylo pro účely SOČ finančně náročné. Tento fakt však spolu s případnou možností použití GPS představuje další možnosti pro pokračování na rozšíření a dalším zkvalitnění této práce.

## 5 POUŽITÁ LITERATURA

Příspěvatelé Wikipedie, *Rádiem řízený model* [online], Wikipedie: Otevřená encyklopedie, c2021, Datum poslední revize 15. 09. 2021, 09:10 UTC, [citováno 18. 03. 2023] <[https://cs.wikipedia.org/w/index.php?title=R%C3%A1diem\\_%C5%99%C3%ADzen%C3%BD\\_model&oldid=20474755](https://cs.wikipedia.org/w/index.php?title=R%C3%A1diem_%C5%99%C3%ADzen%C3%BD_model&oldid=20474755)>.

*Speedybee F7 V3 Flight Controller*. Speedy Bee. (n.d.). Retrieved March 18, 2023, from <https://www.speedybee.com/speedybee-f7-v3-flight-controller/>

PixyCam – Pixy is the easiest way to add vision to your robot!. *PixyCam – Pixy is the easiest way to add vision to your robot!* [online]. Copyright © 2023 [cit. 18.03.2023]. Dostupné z: <https://pixycam.com/>

Hořejší - Gimbal T3D III pro GP3/4. *Modely Hořejší* [online]. Copyright © 2010 Hořejší s.r.o. [cit. 18.03.2023]. Dostupné z: <https://horejsi.cz/Pages/DetailProdukt.aspx?objcislo=3104>

Free CAD Designs, Files & 3D Models | The GrabCAD Community Library. *GrabCAD Making Additive Manufacturing at Scale Possible* [online]. Copyright © 2023 Stratasys Inc. [cit. 18.03.2023]. Dostupné z: <https://grabcad.com/library/crius-multiwii-se-1>

Arduino - Home. *Arduino - Home* [online]. Dostupné z: <https://www.arduino.cc/>

Arduino Pro Mini 328 - 5V / 16MHz - SparkFun DEV-11113 Botland - obchod s robotikou. *Online obchod s elektronikou Botland - obchod s robotikou* [online]. Copyright © Copyright 2023 [cit. 18.03.2023]. Dostupné z: <https://botland.cz/arduino-hlavni-moduly/1595-arduino-pro-mini-328-5v-16mhz-sparkfun-dev-11113-5903351240673.html>

ceptimus - YouTube. *YouTube* [online]. Copyright © 2023 Google LLC [cit. 18.03.2023]. Dostupné z: <https://www.youtube.com/@ceptimus/videos>

## 6 SEZNAM OBRÁZKŮ A TABULEK

Obr. 1: Řídící deska SpeedyBee F7 V3.....	7
Obr. 2: Přijímač FrSky V8FR-II HV.....	8
Obr. 3: Vysílač OPTIC 6.....	9
Obr. 4: Kamera Pixy.....	10
Obr. 5: Ukázka obrazového snímání.....	10
Obr. 6: Gimbal T3D III.....	11
Obr. 7: Arduino IED.....	12
Obr. 8: Arduino Arduino pro miny.....	12
Obr. 9: MultiWii SE v0.2.....	13
Obr. 10: Přistávací nožičky v AutoCAD Inventor.....	14
Obr. 11: Přistávací nožičky hotové.....	14
Obr. 12: Držák kamery.....	15
Obr. 13: blokové schéma.....	16
Obr. 14: program INAV.....	19
Tab. 1: Přehled parametrů řídicí desky SpeedyBee F7 V3.....	7
Tab. 2: Přehled parametrů Gimbalu T3D III.....	11

## 7 PŘÍLOHA 1: PROGRAM MULTIWII SE v0.2

```
#include <SPI.h>
```

```
#include <Pixy.h>
```

```
Pixy pixy;
```

```
#include <Servo.h>
```

```
Servo servox;
```

```
Servo servoy;
```

```
Servo servot;
```

```
#define RCSpin 12
```

```
static int i = 0;
```

```
int x = 0;
```

```
int y = 0;
```

```
bool arm = 0; //bool arm
```

```
int state = 0; //int state
```

```
int RCStart;
```

```
unsigned long pasttime = 0;
```

```
unsigned long levl_time;
```

```
int last_x = 158;
```

```
int last_y = 158;
```



```

int x_servo = 1500;

int y_servo = 1500;

int x_vector;

int y_vector;

uint16_t blocks;

void setup() {

  Serial.begin(9600);

  // Serial.print("Starting...\n");

  pixy.init();

  servot.attach(2);

  servox.attach(5);

  servoy.attach(4);

  levl_time = millis();

  // pastime = millis();

  pinMode(RCSpin , INPUT);

}

void loop() {

  blocks = pixy.getBlocks();

  if(arm == 0 ){          // čekání na příkaz pro start

    servot.write(1059);

    RCStart = pulseIn(RCSpin, HIGH);

```

```

    // Serial.println(RCStart);

    }

if(RCStart >= 1750){

arm = 1;

RCStart = 0;

state = 0;

}

if(arm == 1){

motor();          // zapnutí chodu letu

}

if((blocks == 0)||(blocks > 1)){    //pokud není nalezen block nebo jich je víc dron se rovná
do nulové polohy

    if(millis() >= levl_time + 200){ //dron i přes strátu bloku ještě 0,2 s letí směrem kde je
naposledy viděl

        servox.write(1500);        // středová poloha výchylek dronu

        servoy.write(1500);

        last_x = 158;

        last_y = 158;

        x_servo = 1500;

        y_servo = 1500;

        x_vector = 0;

        y_vector = 0;

        }

} else {    // řízení osy X a Y pokud je nalezen 1 block

    xout(); // řízení osy X

```

```

        yout();    // řízení osy Y

        lev1_time = millis();

    }

// Serial.println (blocks);

}

void xout(){      //program pro sledovaání osy x a navadění se nad ní

last_x = x;

x = pixy.blocks[i].x; // získání souřadnic z pixy

x_vector = x - last_x; // zjišťování vektoru pohybu

if(x < 158){

    if(x_vector < 0){

        x_servo = map(x, 0, 158, 1900, 1500); // mapování souřadnic na výchilky

    }else{

        x_servo = map(x, 0, 158, 1600, 1500); // mapování souřadnic na malých výchilek výchilky

        //x_servo = 1500;// spíše nepoužívat

    }

}

}

}else if(x > 158){

    if(x_vector > 0){

        x_servo = map(x, 158, 316, 1500, 1100);

    }else{

        x_servo = map(x, 158, 316, 1500, 1400);

        //x_servo = 1500; //spíše nepoužívat

    }

}

```

```

}else x_servo = 1500;

//x = map(x, 0, 316,1900 ,1100 ); // test

if(x_servo > 1900) x_servo = 1900;

if(x_servo < 1100) x_servo = 1100;

servox.write(x_servo);

}

```

```

void yout(){          //program pro sledovaání osy Y a navadění se nad ní

last_y = y ;

y = pixy.blocks[i].y;

y_vector = y - last_y;

if(y < 104){

if(y_vector < 0){

    y_servo = map(y, 0, 104, 1900, 1500);

}else{

    y_servo = map(y, 0, 104, 1600, 1500);//y_servo = 1500;

    //y_servo = 1500;

}

}else if(y > 104){

if(y_vector > 0){

    y_servo = map(y, 104, 208, 1500, 1100);

```

```

}else{

  y_servo = map(y, 104, 208, 1500, 1400); //y_servo = 1500;

  //y_servo = 1500;

}

}else y_servo = 1500;

if(y_servo > 1900) y_servo = 1900;

if(y_servo < 1100) y_servo = 1100;

//y = map(y, 0, 208,1900 ,1100); //přehozeno

servoy.write(y_servo);

}

void motor(){      // program pro řízení motorů

switch (state) {

  case 0:

    state = 1;

    pasttime = millis();

    break;

  case 1:      //stoupání

    servot.write(1480); // hodnota určující sílu motorů

    // Serial.println("1");

    if(millis() - pasttime >= 400){ // přenastavení času

      state = 2;

      //Serial.println("1");

      pasttime = millis();

```

```

    }
break;

case 2:    //držení výšky

    if (blocks == 1) { //zjišťování počtu bloků

        servot.write(1460); // hodnota určující sílu motorů

        if((x<100 && x>200) && (y<65 && y>130)){ // čekání na to až se dron dostane do
určitého bodu nad cílem

            if(millis() - pasttime >= 1000){ //přenastavení času

                state = 3;

                pasttime = millis();

            }

        }

    }else {

        state = 3;

        pasttime = millis();

    }

break;

case 3:    //klesání

    servot.write(1438); // hodnota určující sílu motorů

break;

}

}

```