



ZÁPADOČESKÁ  
UNIVERZITA  
V PLZNI

Fakulta elektrotechnická  
Katedra aplikované elektroniky a telekomunikací

# DIPLOMOVÁ PRÁCE

Realizace létajícího prostředku

Autor práce: Bc. Roman Čorba  
Vedoucí práce: Ing. Petr Weissar, Ph.D.

Plzeň 2015

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Roman ČORBA**  
Osobní číslo: **E12N0048P**  
Studijní program: **N2612 Elektrotechnika a informatika**  
Studijní obor: **Elektronika a aplikovaná informatika**  
Název tématu: **Realizace létajícího prostředku**  
Zadávací katedra: **Katedra aplikované elektroniky a telekomunikací**

### Z á s a d y p r o v y p r a c o v á n í :

Navrhněte a realizujte létající prostředek typu "quadcopter".

1. Využijte vhodné mechanické řešení.
2. Vyberte vhodnou platformy řídicí elektroniky a realizujte řídicí algoritmy.
3. Uvažujte různé způsoby předávání povelů z řídicího rozhraní.
4. Diskutujte možnosti realizovaného zařízení.

Rozsah grafických prací: **podle doporučení vedoucího**  
Rozsah pracovní zprávy: **30 - 40 stran**  
Forma zpracování diplomové práce: **tištěná/elektronická**  
Seznam odborné literatury:

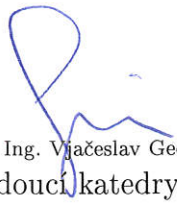
**Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.**

Vedoucí diplomové práce: **Ing. Petr Weissar, Ph.D.**  
Katedra aplikované elektroniky a telekomunikací

Datum zadání diplomové práce: **15. října 2014**  
Termín odevzdání diplomové práce: **11. května 2015**

  
Doc. Ing. Jiří Hammerbauer, Ph.D.  
děkan



  
Doc. Dr. Ing. Vjačeslav Georgiev  
vedoucí katedry

V Plzni dne 15. října 2014

# Abstrakt

Diplomová práce je zaměřena na návrh a realizaci létajícího prostředku typu "quadcopter" řízenou pomocí 32bit ARM mikroprocesoru od společnosti STMicroelectronics řady STM32F4xx. V práci jsou popsány typy různých možných mechanických řešení quadcoptér, princip stabilního letu a popsána případná vhodnost použití jednotlivých mechanických prvků. Dále je popsáno softwarové řešení řídicího algoritmu a zpracování dat z jednotlivých senzorů. V další části této práce jsou popsány možnosti předávání povelů z řídicího rozhraní a možné rozšíření realizovaného řešení.

## Klíčová slova

quadcopter, STM32F4xx, MPU6050, dron, kvadrakoptéra, mikrokontrolér

# Abstract

Čorba, Roman. Realization of flying device [Realizace létajícího prostředku]. Pilsen, 2015. Master thesis (in Czech). University of West Bohemia. Faculty of Electrical Engineering. Department of Applied Electronics and Telecommunications. Supervisor: Ing. Petr Weissar, Ph.D.

---

This master thesis is focused on the design and implementation of a flying device type "quadcopter" controlled using the 32-bit ARM microprocessor from STMicroelectronics series STM32F4xx. This thesis describes the different types of possible mechanical solutions quadcopter, the principle of stable flight, and discussed potential suitability of various mechanical elements. Further described is a software solution the control algorithm and processing data from various sensors. In another part of this work describes the options for transmitting commands from the control interface and the possible extension of this implemented solution.

## Keywords

quadcopter, STM32F4xx, MPU6050, drone, microprocessor

## Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem svou závěrečnou práci vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 270 trestního zákona č. 40/2009 Sb.

Také prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.

V Plzni dne 11. května 2015

Bc. Roman Čorba

.....

Podpis

# Obsah

<b>Seznam obrázků</b>	<b>VI</b>
<b>Seznam symbolů a zkratk</b>	<b>VII</b>
<b>1 Úvod</b>	<b>1</b>
<b>2 Quadcopter</b>	<b>2</b>
2.0.1 Druhy quadcoptér . . . . .	2
2.0.2 Výběr rámu . . . . .	3
2.0.3 Výběr motorů a vrtulí . . . . .	4
2.0.4 Výběr akumulátoru . . . . .	5
2.0.5 Výběr regulátorů . . . . .	6
2.1 Konfigurace realizovaného řešení . . . . .	6
2.1.1 Rám . . . . .	6
2.1.2 Motory . . . . .	7
2.1.3 Vrtule . . . . .	7
2.1.4 Akumulátor . . . . .	9
2.1.5 Regulátory . . . . .	9
2.1.6 Ověření konfigurace kalkulátorem . . . . .	9
<b>3 Řízení</b>	<b>12</b>
3.1 Mikroprocesor STM32F429 . . . . .	12
3.2 6DOF senzor MPU6050 . . . . .	13
3.3 Ovládací modul Turnigy 9X8Cv2 . . . . .	14
3.4 Čtení dat z MPU6050 . . . . .	14
3.5 Zpracování dat z Turnigy 9X8Cv2 . . . . .	17
3.6 Řídící algoritmus . . . . .	18
3.7 Testování realizované quadcoptéry . . . . .	21
3.7.1 Blokové schéma . . . . .	21
3.7.2 Testování motorů a ladění PID . . . . .	22
3.7.3 Možnosti realizované quadcoptéry . . . . .	23
3.8 Možnosti rozšíření realizované quadcoptéry . . . . .	25

3.8.1	Modul kompasu a altimetru . . . . .	25
3.8.2	WiFi modul . . . . .	25
3.8.3	GPS modul . . . . .	26
3.8.4	Xbee modul . . . . .	27
<b>4</b>	<b>Závěr</b>	<b>28</b>
	<b>Reference, použitá literatura</b>	<b>29</b>



# Seznam obrázků

2.1	Typy orientace quadcoptér. . . . .	3
2.2	Uhlíkový rám Turnigy Talon. [2] . . . . .	7
2.3	NTM Prop Drive 2826 1000kv 235W. [2] . . . . .	8
2.4	APC 8045 CW a CCW s adaptéry.[2] . . . . .	8
2.5	Akumulátor Zippy 3000mAh 4S1P 20C. [2] . . . . .	9
2.6	Regulátor Hobby King 20A ESC 3A UBEC. [2] . . . . .	10
2.7	Vypočtené data pomocí kalkulátoru [4] . . . . .	10
2.8	Charakteristika výkonu motorů [4] . . . . .	11
3.1	STM32F429 Discovery . . . . .	12
3.2	MPU6050 kit . . . . .	13
3.3	Turnigy TGY9X . . . . .	14
3.4	Přijímač Turnigy 9X8Cv2 [2] . . . . .	15
3.5	Diagram čtení dat z MPU6050 . . . . .	16
3.6	Naměřená doba mezi sadou pulzů . . . . .	17
3.7	Naměřená doba trvání pulzu kanálu 1 . . . . .	18
3.8	Zjednodušený diagram řídicího algoritmu . . . . .	19
3.9	Blokové schéma . . . . .	21
3.10	Držák pro testování quadcoptéry . . . . .	23
3.11	Testování realizované quadcoptéry . . . . .	24
3.12	XTRINSIC-SENSE Board . . . . .	25
3.13	WiFi modul ESP8266 . . . . .	26
3.14	NEO-6M GPS Modul [2] . . . . .	27
3.15	Xbee modul (vlevo) a Xbee PRO modul [3] . . . . .	27

# Seznam symbolů a zkratek

QUADCOPTER	Quad motor flying device. <i>Čtyřmotorový létající prostředek.</i>
MPU	Motion procesor unit. <i>Pohyb zpracovávající jednotka.</i>
DMP	Digital Motion Procesor. <i>Digitální pohybový procesor.</i>
MCU	Micro Controller Unit. <i>Programovatelné hradlové pole.</i>
LCD	Liquid crystal display. <i>Displej z tekutých krystalů.</i>
I2C	Inter-Integrated Circuit. <i>Multi-masterová sériová sběrnice.</i>
GPIO	General-purpose I/Os. <i>Vstupně výstupní porty mikrokontroléru.</i>
SPI	Serial Peripheral Interface. <i>Sériové rozhraní pro přenos dat.</i>
PWM	Pulse Width Modulation. <i>Pulzně šířková modulace.</i>
PPM	Pulse Position Modulation. <i>Pulzně polohová modulace.</i>
DOF	Degree of Freedom. <i>Stupně volnosti.</i>

# 1. Úvod

Diplomová práce je zaměřena na návrh a následnou realizaci létajícího prostředku typu "quadcopter" neboli čtyřmotorového vrtulníku s použitím vhodné řídicí platformy. Pro řízení byl zvolen mikroprocesor STM32F429 na kterém probíhá následné zpracování dat z řídicího rozhraní a připojených senzorů. Je zde popsán výběr mechanických a elektronických prvků realizované quadcoptéry. Dále také vhodnost použití různých typů rámů, motorů, vrtulí, akumulátorů, regulátorů a senzorů a případné kritéria pro možnost výběru alternativní kombinace jednotlivých dílů.

Zároveň je popsána komunikace a algoritmy pro získávání dat z jednotlivých komponentů quadcoptéry. Následně je popsáno jejich zpracování a popsány jednotlivé kroky řídicího algoritmu, který zabezpečuje stabilní let quadcoptéry.

V poslední části této práce je zhodnoceno výsledné realizované řešení a uvedeny další možné rozšíření o přídatné moduly, které dokáží zvýšit použitelnost pro další možné využití. Vzhledem k použití vysoce výkonného mikroprocesoru, který v realizovaném řešení má ještě dostatek dalšího výpočetního výkonu. Můžeme tuto quadcoptéru dále vylepšovat nepřeborným množstvím dalších možných modulů bez omezení stability stávajícího řešení.

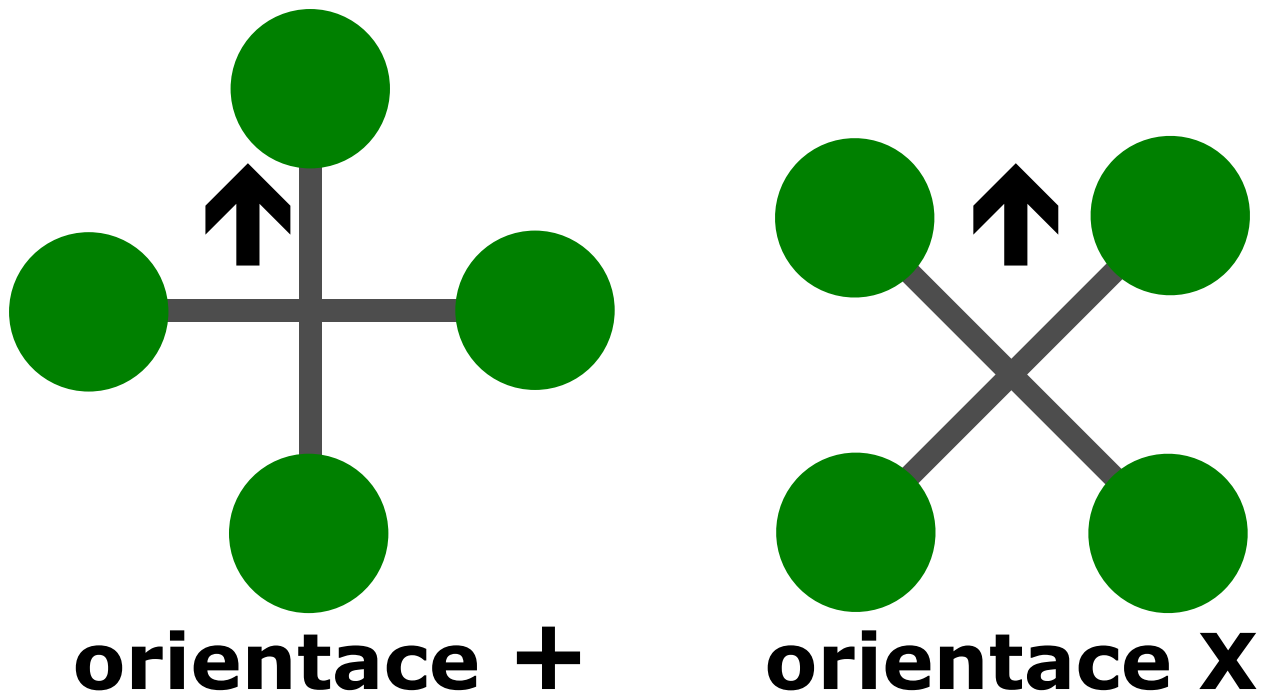
## 2. Quadcopter

Quadcopter počeštěně "kvadrakoptéra"někdy také "dron". Jedná se o létající prostředek který se v současnosti dostává na první místa v prodeji bezdrátových modelů. Tento velký rozmach tohoto létajícího prostředku je způsoben hlavně cenovou dostupností, a tím že se jedná o jednoduše ovladatelný vrtulník. Avšak myšlenka čtyřmotorového létajícího prostředku vnikla již v roce 1907, kdy Louis Breguet navrhl "Breguet-Richet Gyroplane" který však létal pouze několik stop nad zemí [5]. Jak již vyplývá z názvu jedná se o čtyřmotorový vrtulník, který je poháněn čtyřmi nezávisle kontrolovanými elektrickými motory osazenými většinou dvoulistými vrtulemi. Principem fungování je jednoduchá myšlenka. Použitím čtyř motorů osazenými vrtulemi, které vytvářejí svou rotací vztlakovou sílu se tento objekt dokáže vznášet. Avšak problémy nastanou při pokusu o realizaci takového zařízení. Hlavní problém je najít vhodně výkonné mechanické řešení dostačující pro námi zvolený cíl použití. A to nejen z finančního hlediska, kde čím větší nosnost má quadcoptéra mít, tím se bude exponenciálně navyšovat pořizovací cena jednotlivých komponentů. Protože, čím větší potřebný tah od vrtulí potřebujeme, tím potřebujeme větší vrtule a silnější motory. Tímto se nám zároveň velmi navýší hmotnost, a hlavně nutnost zajistit potřebnou dodávku proudu od akumulátoru, který je opět s rostoucí kapacitou dražší a těžší. Největší problém je však nutnost precizního řízení tahu jednotlivých motorů pro zajištění stabilního letu a jejich ovládání pro změnu směru a zamezení rotace způsobenou vířivým proudem vzduchu od jednotlivých motorů. Nechtěnou rotaci omezíme použitím jednoho páru vrtulí které se budou otáčet ve směru hodinových ručiček tzv. "CR"neboli clock rotation a druhého páru vrtulí které se budou otáčet opačným směrem, proti směru hodinových ručiček tzv."CCR"neboli counter clock rotation.

### 2.0.1 Druhy quadcoptér

Quadcoptéry dělíme na dva druhy, a to typ "X" a typ "+"tzv."plus"znázorněné na obrázku 2.1. Hlavním rozdílem je orientace směru letu vůči čtyřem motorům, jejichž vzdálenost od středové platformy je vždy stejná pro každý jednotlivý motor. Typ "+" byl první který se začal používat, a to hlavně z důvodu jednoduchého řízení. Protože v případě, že chcete způsobit pohyb dopředu, stačí zvýšit tah zadního motoru a snížit tah předního motoru. Tato změna způsobí pohyb směrem kde bude působit tah více horizontálně než vertikálně. Pro pohyb do stran opět regulujeme pouze 2 motory, a to levý a pravý, kdy při snížení tahu levého motoru a navýšení tahu pravého motoru se bude quadcoptéra pohybovat směrem doleva a naopak při obrácené změně tahu. Při použití typu

"X" se quadcoptéra ovládá pro změnu směru změnou tahu všech čtyř motorů současně, protože zde je orientace taková, že máme přední levý a přední pravý motor a zadní levý a zadní pravý motor. Pro pohyb dopředu omezíme tah obou předních motorů a zvýšíme tah obou zadních motorů. Tato změna tahu způsobí naklonění vůči zemi a tím získáme tah vodorovným směrem v námi požadovaném směru pohybu. Hlavní výhodou typu "X" je především to, že při použití kamery získáme větší pozorovací úhel.



Obr. 2.1: Typy orientace quadcoptér.

## 2.0.2 Výběr rámu

Rám pro konstrukci quadcoptéry tvoří středová platforma, na kterou jsou připevněny 4 ramena s držáky pro motory. Velikost konstrukce se značí vzdáleností středů protilehlých motorů. Při výběru velikosti rámu musíme již vědět za jakým účelem quadcoptéru stavíme. V případě že chceme akrobatický model, zvolíme menší rozměr okolo 350 mm mezi motory. Avšak jestli chceme použít větší kameru nebo fotoaparát, potřebujeme stabilní let. A proto zvolíme větší velikost konstrukce 500 až 650 mm mezi motory. Základní pravidlo je, aby tato ramena byla stejně dlouhá a tím dodržena stejná vzdálenost jednotlivých motorů od středu a zároveň vzdálenost jednoho motoru vůči druhému. Při nedodržení dostatečné vzdálenosti motorů vůči sobě, nám vznikne problém s působením víření vzduchu od vrtulí. U výběru rámu nás zajímá tedy hlavně vzdálenost motorů, která nám omezí velikost možných variant použitelných vrtulí. Dále je potřeba zvolit pevný rám, který se nebude vlivem působení tahu motorů deformovat. Avšak nesmíme opomenout to, že se u konstrukce snažíme minimalizovat její váhu.

Proto máme na výběr z těchto běžných variant:

- Dřevěná konstrukce
- Plastová konstrukce
- Kovová konstrukce
- Uhlíková konstrukce

Dřevěná konstrukce rámu quadcoptéry je bezesporu nejlevnější řešení, které si můžeme vyrobit doma. Tato varianta je však většinou těžká a v případě že použijeme slabší materiál tak je křehká a vibrující při zapnutí motorů. Pro první pokusy se toto řešení dá použít už jen z důvodu, že můžete okamžitě vyrobit jiné individuální rozměry rámu.

Plastová konstrukce je běžně dostupná v modelářských potřebách, tato konstrukce se vyznačuje hlavně svou pevností a odolností vůči deformaci vznikajícím tahem motorů. Takovéto konstrukce se dají pořídit v řádu stovek až tisíců korun, dle požadovaných rozpětí a kvality. V současné době se vyrábí i plastové rámy s integrovanými napájecími cestami, které nám usnadní rozvod napájení od regulátorů k motorům. Tento druh konstrukce se však běžně používá pouze u menších rozpětí motorů, do 450 mm.

Kovová konstrukce, nejběžněji používaná konstrukce pro větší modely quadcoptér o velikostech od 500 mm. Tato konstrukce odolává deformacím lépe než předchozí dva druhy. Vyrábí se z hliníkových případně duralových profilů. Dá se běžně zakoupit již hotové řešení připravené pro osazení motory a další elektronikou. Avšak takovouto konstrukci si mnoho modelářů staví svépomocí z profilů které jsou běžně dostupné v kterémkoliv železářství.

Uhlíková konstrukce je nejlepší možná varianta rámu, avšak také nejdražší. Tato konstrukce nejlépe odolává vibracím i deformacím vznikajícím prudkými změnami tahu motorů. Zároveň tato konstrukce neovlivňuje elektroniku, a nemůže na rozdíl od železné konstrukce vzniknout zkrat elektroinstalace při poškození izolace napájecích kabelů. A nejdůležitější výhodou této konstrukce je vysoká pevnost při zlomkové hmotnosti v porovnání s předchozími rámy. Tento druh rámu se vyrábí v celém spektru použitelných rozměrů, avšak nejběžnější velikost je 500 mm.

### 2.0.3 Výběr motorů a vrtulí

Motory používané pro quadcoptéry se vyrábí ve dvou variantách. Stejnoseměrné motory, které se používají pouze u malých modelů do 100 mm. A BLDC motory využívané pro většinu dnešních quadcoptér. Hlavní parametr kterým se liší jednotlivé motory tohoto typu je kromě výkonu parametr "Kv" neboli počet otáček za minutu na volt. Motory s 500-1300Kv se používají především pro stabilnější pomalé létání. A motory s 1400-4000Kv jsou určeny pro akrobatické letouny.

Běžně se pro quadcoptéry používají akumulátory LiPo tří až čtyř-člankové. Tříčlankový akumulátor neboli "3S" má jmenovité napětí 11,1V a pro příklad použití s 1000Kv motorem bude

nezatížený motor mít  $11,1 \cdot 1000 = 11100$  otáček za minutu. Případně 14800 otáček za minutu při použití čtyř-článekového akumulátoru.

Výběr vhodných vrtulí závisí na námi použitým motoru. Vrtule se používají především plastové, ale můžeme se setkat i uhlíkovými vrtulemi. Vrtule mají označení například 1045, což znamená 10" průměr a 4,5" stoupání za jednu otáčku.

Při výběru vhodného motoru a velikosti vrtule již musíme znát přibližnou celkovou váhu modelu. Takže v případě že náš model bude vážit celkem i s motory 1000 gramů, je nutno dodržet pravidlo že celkový tah motorů musí být minimálně dvakrát větší než hmotnost modelu. Proto pro takovýto model vybereme kombinaci čtyř motorů a vrtulí, které mají tah alespoň 500 gramů. Je potřeba si však dát pozor na předimenzování tahu motorů, protože tím si snížíme možný rozsah regulace otáček a bude problém v tomto rozsahu nastavit výsledný tah motorů pro stabilní let. Pro výpočet tahu námi zvolené konfigurace motoru s vrtulí můžeme použít jednu z mnoha kalkulaček běžně dostupných na internetu například <http://www.ecalc.ch/motorcalc.php> [?]. Případně si tah vypočítat ručně pomocí několika vzorečků v níž budeme potřebovat znát konstanty efektivity námi zvolené vrtule.

## 2.0.4 Výběr akumulátoru

Akumulátor je součástí quadcoptéry s největší hmotností, proto je potřeba zvolit vhodný typ a jeho kapacitu tak aby jsme zbytečně nezatěžovali náš model. V dnešní době se nejčastěji používají Lithium Polymerové akumulátory neboli "Li-Po" a to hlavně z důvodu jejich nízké váhy vzhledem k vysoké kapacitě. Tyto akumulátory se vyrábějí několika článkové a pro quadcoptéry velikosti okolo 500mm se používají 3S případně 4S. Výběr akumulátoru závisí především na tom, jak dlouho má quadcoptéra setrvat ve vzduchu na jedno nabití.

U akumulátorů jsou pro nás důležité tyto parametry:

- Počet článků
- Kapacita
- Vybíjecí proud
- Hmotnost

Počet článků akumulátoru nám udává jeho jmenovité napětí. Jeden článek Li-Po akumulátoru má jmenovité napětí 3,7 V. Značíme například 4S1P, kde číslovka před "S" znamená počet sériově zapojených článků a číslovka před "P" počet paralelních zapojení. Výsledné napětí takového akumulátoru je  $3,7 \cdot 4 = 14,8$ V.

Kapacita na akumulátorech je udávána v mAh. Akumulátor o kapacitě 3000mAh dokáže dodávat proud 3A po dobu jedné hodiny.

Vybíjecí proud akumulátoru je statický a špičkový. Akumulátor dokáže špičkově dodat větší proud, avšak provoz dimenzujeme pod statickým vybíjecím proudem. Tento parametr je označen

například 20C kde "C" znamená kapacitu akumulátoru a číslovka před ním udává násobek této kapacity. Takže 3000mAh 20C akumulátor dokáže dodávat stabilně proud 60A.

Hmotnost akumulátoru je nezanedbatelný parametr při výběru vhodného typu. S vyšší kapacitou akumulátoru dokáže náš model déle letět, ale větší kapacita znamená i násobně větší váhu akumulátoru. Běžné 4S1P Li-Po s kapacitou 3000mAh a vybíjecím proudem 20C váží přibližně 346 gramů a při stejných parametrech ale kapacitou 5000mAh se dostaneme na váhu 632 gramů.

## 2.0.5 Výběr regulátorů

Regulátor otáček, kterým ovládáme otáčky našich BLDC motorů. Takovýto regulátor pracuje na jednoduchém principu přivedení napětí na jednotlivé póly při různé frekvenci dle námi požadovaných otáček. Regulátor obsahuje většinou MCU které dokáže dekódovat požadavek otáček zasláný pomocí pulzu o délce 1ms pro vypnutý stav a 2ms pro maximální výkon o frekvenci až 400Hz. Po připojení regulátoru na akumulátor je nutné zaslat několik impulzů o délce 1ms a tímto postupem ho odbrzdíme. Vstupní napětí takovýchto regulátorů pro náš účel je běžně 2-4S. Pro quadcoptéry používáme jednosměrné, bez možnosti zpětného chodu. Většina regulátorů dnes obsahuje BEC neboli výstup stabilizátoru napětí, na který můžeme připojit naši elektroniku, toto napětí nejčastěji bývá 5V.

Důležitým parametrem je hodnota pracovního proudu dodávaného regulátorem. Proto vybíráme regulátory až jako poslední součástku. Zde musíme vzít v úvahu maximální odběr námi vybraných motorů a zvolit výkonnější regulátor, tak aby jsme nepracovali na limitech regulátoru.

## 2.1 Konfigurace realizovaného řešení

Hlavním cílem mnou vybrané konfigurace bylo, aby takováto quadcoptéra dokázala stabilně letět i s připevněnou zátěží o váze několika kilogramů. Proto jsem původně zvolil hliníkový rám o rozměru 550mm který byl dostatečně dimenzován. Avšak po prvním testovacím letu a pádu z výšky přibližně 50-ti metrů z prvního prototypu zbyli pouze motory a regulátory. Proto nastal výběr nových komponent.

### 2.1.1 Rám

Finální rám pro mnou realizovaný poslední model řešení je 500 mm rám z uhlíkových vláken od společnosti Turnigy který nese označení "Talon Carbon Fiber Quadcopter Frame" viz 2.2. Tento rám vyniká především svou pevností a velmi nízkou váhou pouze 240 gramů. Avšak nemá připravené držáky motorů pro průchozí hřídel motorů, proto je potřeba zvolit motory takzvané "Short



shaft" které mají umožněnou montáž na ramena rámu z vrchu. Případně motory ručně upravit, stejně jako v mém řešení.



**Obr. 2.2:** Uhlíkový rám Turnigy Talon. [2]

## 2.1.2 Motory

Pro mou konfiguraci 500 mm rámu jsem pomocí kalkulačků tahu vybral BLDC motory "NTM Prop Drive Series 2826 1000kv" viz 2.3. Při použití s 4S baterií mají 14800 otáček za minutu. Výkon udávaný výrobcem je 230 Wattů. Tyto motory mají kvalitní konstrukci a vhodný výkon pro mé použití.

## 2.1.3 Vrtule

Pro rámy o velikosti 400-600 mm nejčastěji používáme 8-10" vrtule. První kalkulace použití vrtulí 1045, neboli vrtulí o průměru 10" a stoupání 4,5" na otočku byl s mnou vybranými motory vypočten tah 1335 gramů pro každý motor. Při využití všech motorů na maximální výkon by tato konfigurace měla výsledný tah až 5400 gramů což při aktuální váze quadcoptéry okolo 1300 gramů znamenalo velký problém s předimenzováním. Tím že jsem původně zvolil tuto konfiguraci, vznikl problém



**Obr. 2.3:** NTM Prop Drive 2826 1000kv 235W. [2]

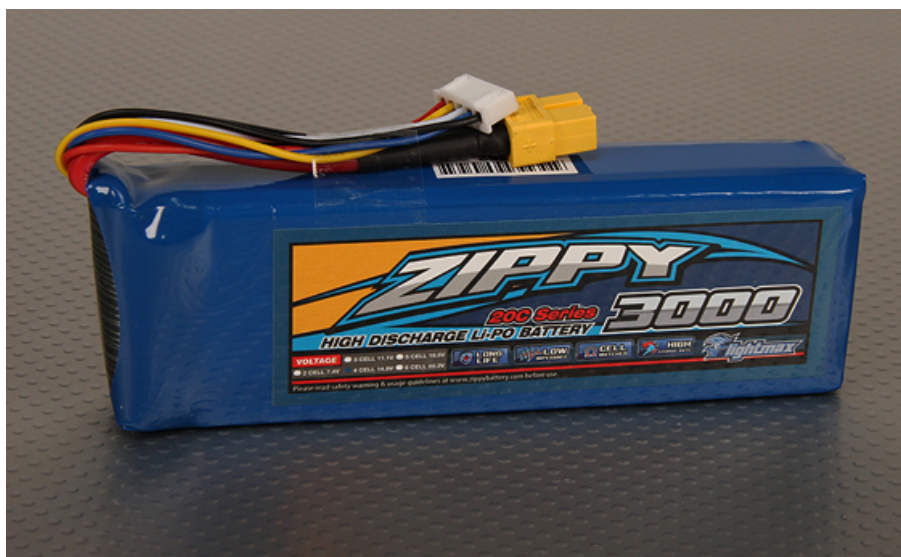
až při pokusech o stabilizaci letu. Proto jsem následně zvolil menší vrtule o velikosti 8045 od společnosti APC viz 2.4, které mají výsledný kalkulovaný tah v této konfiguraci 915 gramů. Tímto jsme se dostali na 3660 gramů maximálního tahu při použití 1000kV motorů a 4S akumulátoru. Proto tyto vrtule zvládnou bez problému pohánět až 1890 gramovou quadcoptéru.



**Obr. 2.4:** APC 8045 CW a CCW s adaptéry.[2]

## 2.1.4 Akumulátor

Mnou zvolený akumulátor byl typu Li-Po od společnosti Zippy o konfiguraci 4S1P 3000mAh 20C viz 2.5. Tento akumulátor dokáže trvale dodávat 60A při 14,8V, což je dostačující pro naše požadované využití quadcoptéry. Tento akumulátor váží 313 gramů, a je to nejtěžší součástka celé konstrukce.



Obr. 2.5: Akumulátor Zippy 3000mAh 4S1P 20C. [2]

## 2.1.5 Regulátory

Výběr vhodných regulátorů je jednoduchý. Vzhledem k tomu, že motory v této konfiguraci mají maximální proudový odběr 16 Ampér. Výpočtem kalkulátorem bude maximální odběr při váze 1350 gramů 11 Ampér. Zvolil jsem proto 20 Ampérové regulátory od společnosti Hobby King s označením "20A ESC 3A UBEC" viz 2.6. Tyto regulátory dokáží dodávat 20 Ampér trvale a 25 Ampér při dočasném zatížení. Zároveň má každý z těchto regulátorů stabilizovaný výstup pro napájení dalšího vybavení quadcoptéry o 5V a 3A.

## 2.1.6 Ověření konfigurace kalkulátorem

Nyní máme hlavní součásti vybrány a proto je potřeba ověřit tuto konfiguraci výpočtem. Pro výpočet můžeme použít některý z dostupných modelářských kalkulátorů. A to hlavně z důvodu, že vzorce pro výpočet letového režimu jsou náročné na vstupní data, které nejsou běžně dostupné od výrobců našich komponentů. Proto jsem zvolil online kalkulátor <http://www.ecalc.ch/xcoptercalc.php> [4] kde zadáme námi zvolené komponenty a hmotnost modelu. Výstupem kalkulátoru je mnoho zásadních informací, jako například zda tato konfigurace má dostatečný tah pro vzlet, a také jaký výkon motorů potřebujeme nastavit pro udržení quadcoptéry ve vzduchu.



**Obr. 2.6:** Regulátor Hobby King 20A ESC 3A UBEC. [2]

Do kalkulátoru se vkládá také hmotnost našeho modelu, proto je potřeba si tuto hodnotu vypočíst. Případně zvážit kompletní řešení, pokud již máme všechny součásti dostupné. Realizovaná konfigurace quadcoptéry obsahuje tyto části:

Talon Carbon Fiber Quadcopter Frame	240g
4x NTM Prop Drive Series 2826 1000kv	216g
Zippy 3000mAh 4S1P 20C	313g
4x Hobby King 20A ESC 3A UBEC	120g
Kabely a ostatní komponenty	456g
<b>Celková hmotnost</b>	<b>1335g</b>

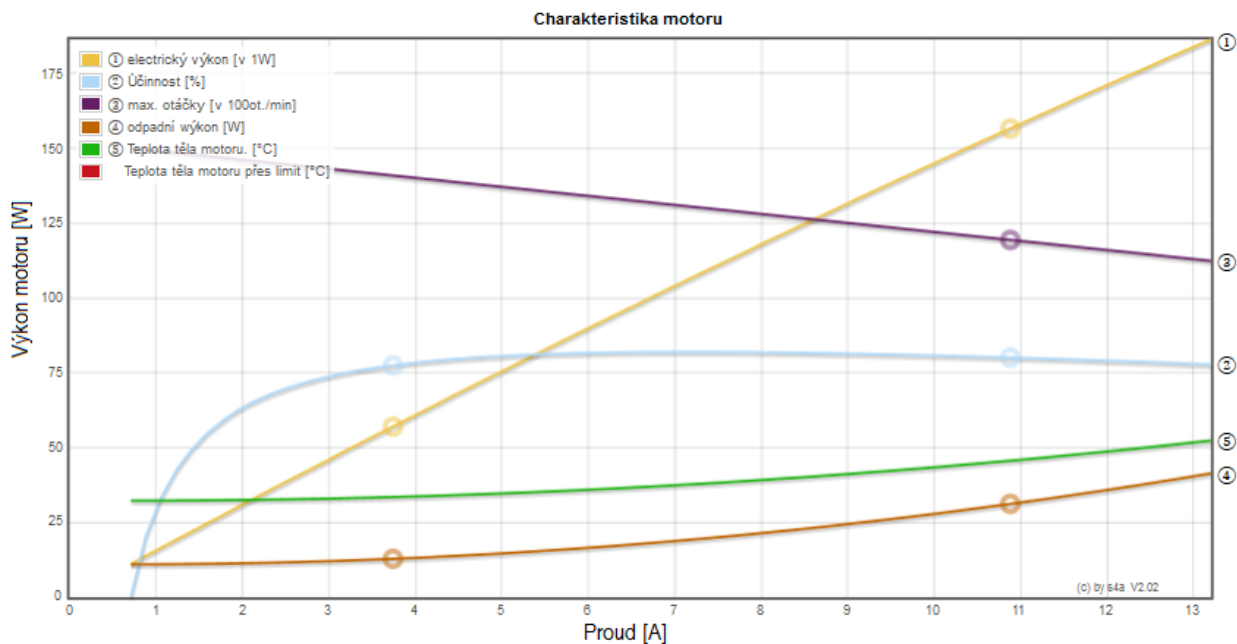
Po zadání naší konfigurace do kalkulátoru máme k dispozici tyto data 2.7:

<b>Akumulátory</b>		<b>Motor @ Hover</b>		<b>Motor @ Maximum</b>		<b>Multicopter</b>	
Zatížení:	14.67 C	Proud:	3.82 A	Proud:	11.00 A	Letová hmotnost:	1335 g
Napětí:	14.49 V	Napětí:	15.23 V	Napětí:	14.38 V		47.1 oz
Jmenovité napětí:	14.80 V	Plyn (lineární):	53 %	Otáček*:	11901 ot./min	Dodatečné zatížení:	631 g
Kapacita:	3000 mAh	Příkon:	58.2 W	Příkon:	158.2 W		22.3 oz
Energie:	44.4 Wh	Mech. výkon:	45.2 W	Mech. výkon:	126.5 W	max. klopení:	47 °
Letový čas:	4.1 min	Účinnost:	77.7 %	Účinnost:	80.0 %	max. rychlost:	60 km/h
Kombinovaný letový čas:	7.4 min	Dosažená teplota:	34 °C	Dosažená teplota:	46 °C		37.3 mph
Letový čas ve visu:	10.0 min		93 °F		115 °F		
Hmotnost:	335 g	Specifický tah:	5.74 g/W				
	11.8 oz		0.2 oz/W				

**Obr. 2.7:** Vypočtené data pomocí kalkulátoru [4]

Z těchto dat vyplývá, že naše konfigurace quadcoptéry dokáže při 53 procentech nastaveného požadovaného výkonu motorů setrvat ve vzduchu bez toho, aby klesala nebo stoupala. A při tomto

nehybném stavu dokáže 10 minut setrvat, než dojde k vybití akumulátoru. Dále zde vidíme, že můžeme přidat dalších 631 gramů dodatečné zátěže což je dostačující například pro připevnění kamery nebo fotoaparátu. Na grafu 2.8 vidíme jaký rozsah regulace máme dostupný pro manévrování.



Obr. 2.8: Charakteristika výkonu motorů [4]

## 3. Řízení

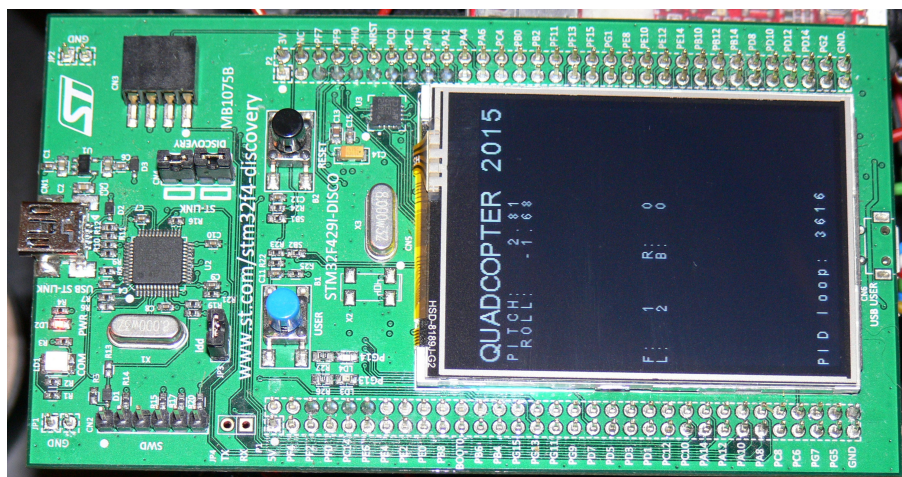
K řízení byla vybrána platforma 32Bit ARM procesoru společnosti ST Microelectronics řady STM32F4xx. Pomocí tohoto MPU se zpracovávají veškeré údaje ze sensorů a řídicího rozhraní. Následně jsou zpracovaná data použita k výpočtům požadovaných hodnot výkonů motorů a odeslána regulátorům, které nastaví požadovaný výkon motorů. Tento proces výpočtů se opakuje každých 5 milisekund. Vzhledem k výkonu zvolené platformy máme dostatek času pro budoucí implementaci dalších rozšíření.

### 3.1 Mikroprocesor STM32F429

Pro zpracování dat ze sensorů a výpočty jsem zvolil nejdříve kit STM32F407 Discovery, který jsem později nahradil STM32F429 Discovery kitem 3.1. Tento vývojový kit má oproti předchozímu modelu výhodu v integrovaném 2.4" QVGA TFT LCD pomocí kterého můžeme v průběhu stavby a implementace algoritmů zobrazovat aktuální důležité údaje.

Tento vývojový kit je vybaven 32-Bitovým ARM procesorem založený na jádře Cortex-M4 s FPU neboli jednotkou která umí počítat s plovoucí desetinnou čárkou. Zároveň procesor umí DSP instrukce a pracuje až na 180Mhz. Pro námi vytvářený program máme dostupné 2 Mb Flash paměti a 256Kb SRAM pro proměnné.

Pro nás je využívána především sběrnice I2C pro komunikaci s LCD a MEMS senzory.

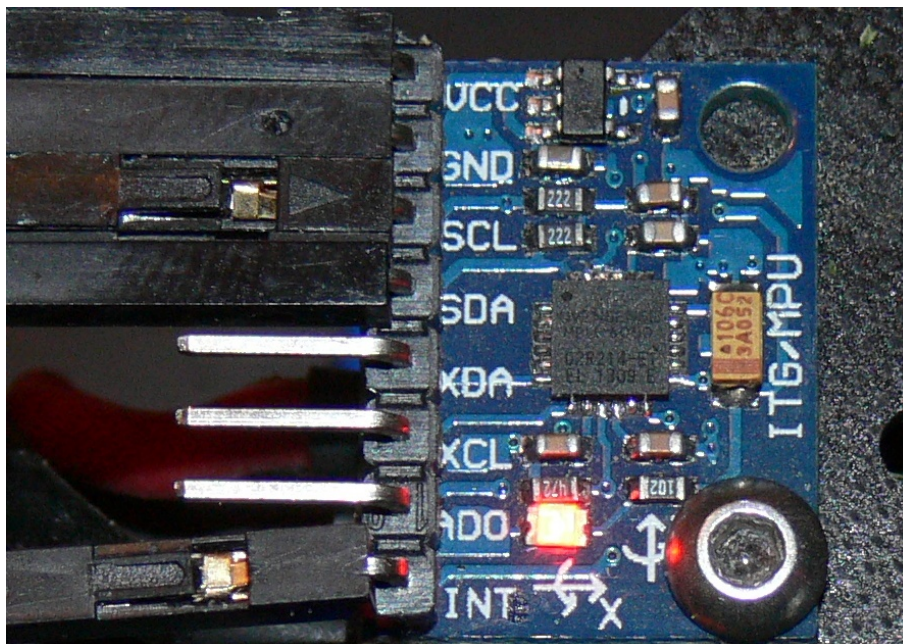


Obr. 3.1: STM32F429 Discovery

## 3.2 6DOF senzor MPU6050

Pro stabilní let quadcoptéry potřebujeme znát především její polohu vůči zemi. Tuto informaci vypočteme pomocí kombinace akcelerometrů a gyroskopů. S touto informací již můžeme vypočítat výkony jednotlivých motorů pro stabilizaci požadované polohy. Na výběr máme mnoho akcelerometrů a gyroskopů, mnou zvolený MPU6050 od společnosti Invensense viz.3.2 obsahuje tříosý MEMS akcelerometr a gyroskop.

Tento senzor obsahuje také DMP jednotku která umí data z akcelerometru a gyroskopu zpracovat a po sběrnici I2C nebo SPI nám odešle již zpracovaná data ve formátu quaternionů.



Obr. 3.2: MPU6050 kit

MPU6050 je první senzor na světě který kombinuje použití 3-osého akcelerometru a 3-osého gyroskopu pomocí DMP neboli Digitálního Pohybového Procesoru. Za pomocí tohoto DMP se senzor dokáže za provozu sám kalibrovat a vypočítávat absolutní náklon vůči zemi. Obsažený gyroskop pracuje v 16-ti bitovém rozlišení a volitelném rozsahu od +/- 250 do +/- 2000 stupňů za vteřinu. Akcelerometr pracuje také v 16-ti bitovém rozlišení s rozsahy od +/- 2 do 16g. Dále tento senzor obsahuje digitální teploměr, který je využíván především DMP pro kalibraci za různých provozních teplot. Data lze ze senzoru číst například nezpracované pomocí I2C při rychlosti 400kHz, avšak při použití DMP pro zpracování dat čekáme na přerušení, které nám oznámí dostupnost nových dat a to při rychlosti 200Hz. MPU6050 podporuje také rozšíření na 9DOF pomocí externího připojení magnetometru.

### 3.3 Ovládací modul Turnigy 9X8Cv2

Pro příjem dat z řídicího rozhraní máme možnost výběru z mnoha variant. V mém realizovaném řešení jsem se rozhodl použít modelářskou 8-mi kanálovou vysílačku Turnigy 9x8C 3.3 která pracuje na frekvenci 2,4Ghz a má dosah přibližně 3000 metrů dle provozních podmínek, jako je například zarušení prostředí. Tuto variantu jsem zvolil zejména z důvodu běžné dostupnosti takovýchto vysílaček.



Obr. 3.3: Turnigy TGY9X

Data z vysílačky jsou přijímány modulem 3.4, který tyto informace dekoduje do osmi PWM výstupů. Těchto 8 signálů nám udává informaci o poloze jednotlivých ovládacích prvků. Máme na výběr jak tyto signály dekodovat, a to buď všech 8 kanálů PWM dekodovat zvoleným mikrokontrolérem což potřebuje uvolnění 8-mi časovačů, nebo použít k dekodování některý z dostupných modulů pro převod 8-mi kanálů které dokáží převést PWM na jeden kanál PPM případně přímo zpracovaná data zasílat přímo po sběrnici I2C.

### 3.4 Čtení dat z MPU6050

Tento senzor který pracuje s určením 6-ti stupňů volnosti. Nám dokáže po sběrnici I2C vracet surová data, případně zpracovaná pomocí DMP. Nejdříve proto konfigurujeme I2C sběrnici pro komunikaci s mikroprocesorem. Následně nastavíme příjem přerušení od MPU6050, které nám oznamuje dostupnost nových dat. Po připojení a nastavení sběrnice již můžeme senzor inicializovat pro získávání dat.





**Obr. 3.4:** Přijímač Turnigy 9X8Cv2 [2]

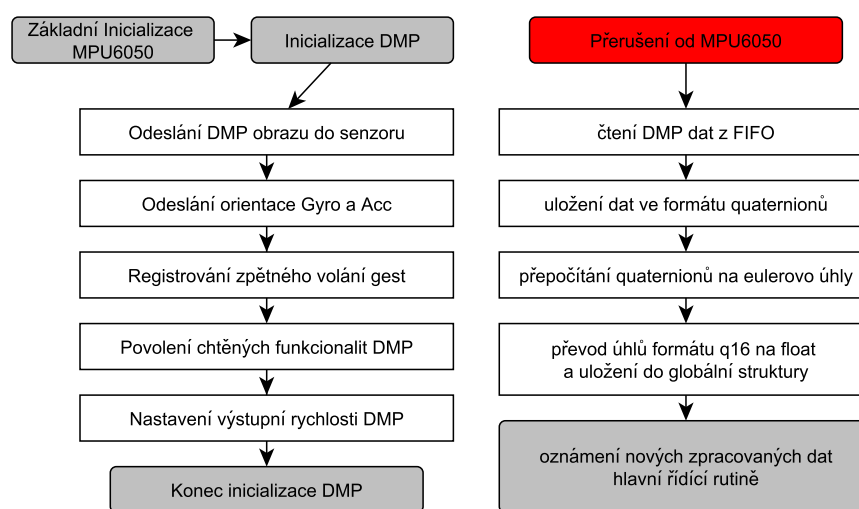
Po inicializaci máme na výběr ze dvou variant a to získávat ze senzoru surová data z akcelerometru a gyroskopu, které následně mikroprocesorem zpracujeme. Tyto data nám udávají zrychlení v každé ze třech os XYZ a rychlost rotace v těchto osách ve stupních za vteřinu. S těmito dvěma sadami informací dokážeme pomocí komplementárního filtru případně využitím Kalmanovo filtru vypočítat absolutní náklon vůči zemi ve stupních. Tato varianta byla také z důvodu jednoduché implementace mnou zvolená jako první verze pro určení pozice quadcoptéry. Avšak tato zpracovaná data jsou nepřesná a mají velký offset od skutečných hodnot, který se s časem ještě navyšuje. Proto jsem se rozhodl k získávání dat o náklonu využít druhou složitěji implementovatelnou variantu a to získávání dat předzpracovaných pomocí vnitřního digitálního pohybového procesoru.

Společnost Invensense, výrobce MPU6050 senzoru, v nedávné době uvolnila knihovny pomocí, kterých lze komunikovat s vnitřním DMP. Avšak tyto knihovny jsou určeny především pro ukázkou co tento senzor dokáže, a zpracování dat je prováděno až na PC. Tato knihovna se jmenuje "Motion Driver" a je napsána pro procesor MSP430. Následně se však uvolnila i verze pro ARM procesory s podporou pouze překladače IAR ARM a opět určenou pro přeposílání dat z MPU6050 po USART do PC. Vzhledem k tomu, že jsem měl již rozpracovaný projekt pomocí vývojového prostředí Keil uVision, musel jsem kompletně tyto knihovny přeportovat a následně získaná data zpracovat do mnou použitelného formátu.

Data zpracovaná pomocí DMP ze senzoru dostaneme dle vývojového diagramu 3.5. Nejdříve provedeme inicializaci I2C sběrnice, poté základní inicializaci rozsahů akcelerometrů a gyroskopů a vzorkovací frekvence dle datasheetu [7], pro využití DMP je nutno zvolit Akcelerometr 2G a Gyroskop 2000 stupňů za vteřinu při maximální vzorkovací frekvenci FIFO 200Hz. Nyní je senzor připraven pro vracet nezpracovaná data z akcelerometrů a gyroskopů. Avšak my chceme využít DMP pro fúzy těchto dat a přepočítání na quaterniony. Proto musíme provést inicializaci DMP pomocí Motion Driver knihovny, kterou jsme si již implementovali pro použití v našem vývojovém prostředí. Inicializaci provedeme následujícím způsobem. Odešleme DMP obraz do paměti senzoru, tento obraz povolí funkci DMP. Poté musíme senzoru sdělit v jaké orientaci vůči zemi pracujeme, následně registrujeme zpětné volání gest, které nevyužijeme ale pro správnou funkčnost senzoru se musí registrovat. Nyní můžeme zaslat senzoru jaké funkcionality má DMP aktivovat, pro nás to budou tyto:

- DMP\_FEATURE\_6X\_LP\_QUAT - generuje quaterniony pomocí fúze akcelerometru a gyroskopu v senzoru za použití DMP.
- DMP\_FEATURE\_SEND\_RAW\_ACCEL - přidá nezpracovaná data z akcelerometru do FIFO
- DMP\_FEATURE\_SEND\_CAL\_GYRO - přidá zpracovaná data z gyroskopů do FIFO
- DMP\_FEATURE\_GYRO\_CAL - povolí automatickou kalibraci gyroskopů při nehybnosti senzoru po dobu 8 vteřin

Po odeslání těchto předvoleb senzoru, již pouze upravíme výstupní frekvenci FIFO a čekáme na přerušení od senzoru, které nám oznámí nově zpracovaná data v FIFO bufferu.



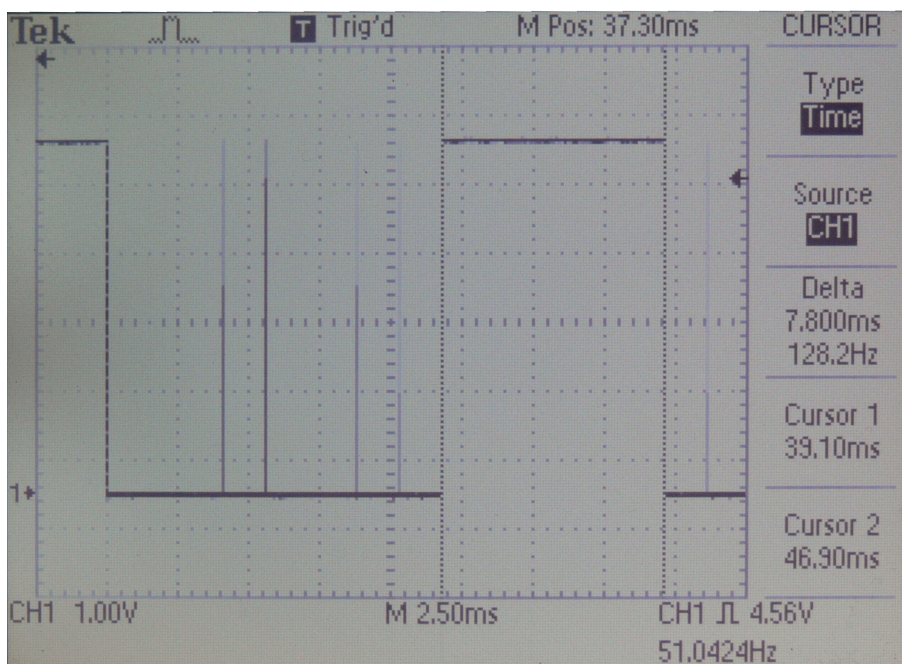
**Obr. 3.5:** Diagram čtení dat z MPU6050

Ihned po detekci přerušení od MPU6050 začneme číst data z FIFO bufferu. Tyto data jsou již zpracovaná pomocí DMP a ve tvaru quaternionů uložených do čtyřprvkového pole. Následně z těchto quaternionů vypočteme polohu senzoru v osách XYZ ve formátu Eulerovo úhlů. Tyto spočítané úhly nyní máme ve formátu Q16 Fixed Point, avšak mnou použitý mikroprocesor má FPU jednotku, proto tyto úhly převedeme na formát Float a uložíme do globální struktury. Po zpracování těchto dat oznámíme hlavní řídicí rutině informaci o nových datech.

Tyto data zpracovaná pomocí DMP jsou na rozdíl od dat zpracovaných ručně pomocí complementárního filtru nebo Kalmanovo filtru přesná a stabilní. Avšak hodnota rotace YAW je bez použití magnetometru nepřesná, použitelná pouze pro orientační informaci. Hodnota YAW se totiž neustále mění i v klidové pozici senzoru, narůstající integrační chybou. V případě že by jsme tuto hodnotu chtěli mít přesnou, je nutné implementovat digitální magnetometr a rozšířit tímto náš 6DOF senzor na 9DOF senzor. MPU6050 toto rozšíření podporuje, nicméně je již možno zakoupit MPU9250 která nativně obsahuje magnetometr. V mé quadcoptěře jsem tento MPU9250 chtěl také použít, avšak bohužel mi přišel nefunkční kus.

### 3.5 Zpracování dat z Turnigy 9X8Cv2

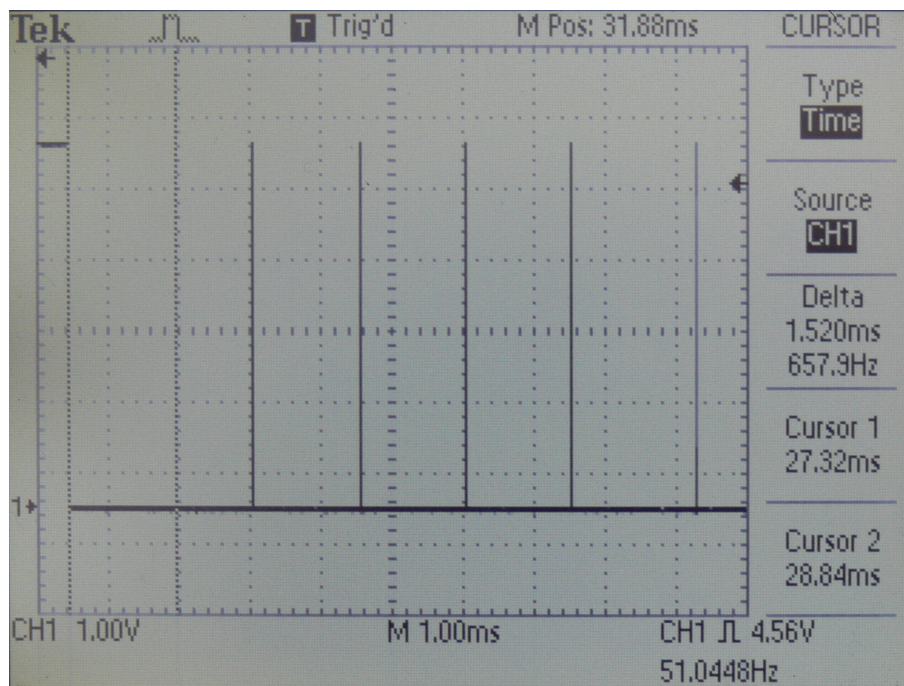
Pro zadávání požadované pozice a směru letu byla zvolena vysílačka Turnigy 9X s komunikačním modulem Turnigy 9X8Cv2. Výstupem z tohoto modulu je 8 signálů PWM dle datasheetu. Proto vznikl problém, jak tyto data zpracovat. Nechtělo se mi nakupovat drahé převodníky, proto jsem pomocí osciloskopu prověřil výstup z tohoto komunikačního modulu. Signál na každém z kanálů je o frekvenci 50Hz, a délka trvání kdy se signál nachází v logické hodnotě 1 je od 1000uS pro stav vypnuto 1500uS pro střední hodnotu a 2000uS pro stav signalizující maximální hodnotu nastavenou na vysílače. Měřením bylo dále zjištěno, že tyto kanály se nepřekrývají, tudíž mě napadlo je sečíst do jednoho signálu. Tuto myšlenku jsem následně realizoval pomocí logického obvodu NOR 74HC4078 od společnosti ST Microelectronics s 8-mi vstupy. Toto řešení lze použít zejména z důvodu, že je mezi ukončením pulzu jednoho kanálu a začátkem pulzu následujícího kanálu stav logické 0 na 3 uS. Po odvysílání všech 8-mi kanálů nastává logická 0 pro doplnění signálu jednotlivých kanálů na 50hz. Tento čas použijeme pro synchronizaci, aby jsme věděli že opět můžeme očekávat další pulz z kanálu 1 následující kanálem 2 a takto pro všech 8 kanálů. Výstupem tedy máme signál PPM, který dekódujeme jedním časovačem. Druhý časovač byl použit pro měření intervalu mezi jednotlivými pulzy, v případě že časovač naměří logickou 0 delší než synchronizační pulz mezi jednotlivými kanály, nastavíme čítání pulzů na kanál 1. Protože delší logická 0 nám říká že následující pulz bude z kanálu 1.



**Obr. 3.6:** Naměřená doba mezi sadou pulzů

Na obrázku 3.6 je znázorněna doba trvání synchronizačního pulzu mezi další sadou po sobě následujících 8-mi pulzů. Jak můžeme vidět, délka synchronizačního pulzu je v tomto případě 7,8 milisekundy. Na následujícím obrázku 3.7 je znázorněna sada pulzů, kdy trvání jednoho pulzu o logické 0 je ukončeno logickou 1 po dobu 3uS a poté následuje další pulz. Zde vidíme že první pulz

je o délce 1520uS. Tato doba po odečtení základní délky pulzu 1000uS značí 520uS v rozsahu 0uS až 1000uS, proto z této hodnoty určíme že kanál 1 je nastaven na 52 procent. Po synchronizační 3uS logické 1 měříme další kanál a takto pro všech 8 kanálů.

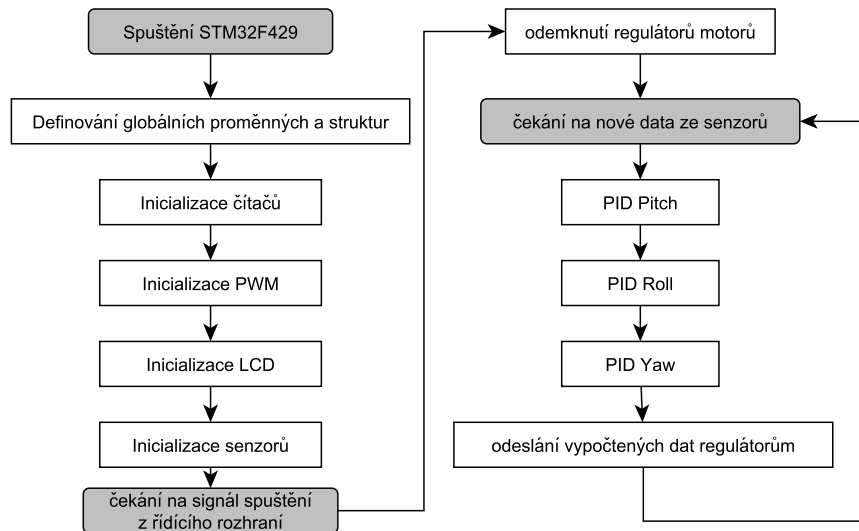


Obr. 3.7: Naměřená doba trvání pulzu kanálu 1

### 3.6 Řídící algoritmus

Řízení pomocí STM32F429 je implementováno následujícím způsobem jak je znázorněno na zjednodušeném vývojovém diagramu 3.8. Při návrhu řídicího algoritmu jsem postupoval dle mnou získaných informací o stabilizaci quadcoptér na ostatních komerčních řešení. Většina mnou prozkoumaných řešení používá akcelerometr a gyroskop, a z těchto senzorů přepočítává náklon vůči zemi. Proto jsem se rozhodl implementovat 6DOF fúzy pomocí DMP v senzoru MPU6050, tímto řešením získávám přesné informace o aktuálním náklonu. Toto řešení se aktuálně v quadcoptérech, které jsou dostupné nepoužívá. Především z důvodu, že DMP je relativně nová věc a obnáší složitější implementaci do řídicího systému. Knihovny pro použití tohoto DMP se uvolnili pro testování s MSP430 a PC minulí rok. Z tohoto důvodu je použití DMP pro získání dat ještě málo prozkoumané řešení, které se z tohoto důvodu není prozatím využíváno. Řídící algoritmus musí umět rychle reagovat na uživatelský podnět, proto je nutné zvolit řešení příjmu dat a jejich rychlé zpracování. Tento řídicí algoritmus pracuje na mnou navrženém sečtení PWM impulzů, které dekódují pomocí čítačů. Nová data s požadavky na změnu aktuální pozice vůči zemi nebo letové výšce tímto řešením získáváme zpracované každých 20ms neboli 50 krát za vteřinu. Data o aktuálním náklonu máme v mnou implementovaném řešení při vypočtená frekvenci 200Hz, proto při kombinaci těchto hodnot získáváme rozdíl požadované a aktuální polohy dostatečně rychle

pro zajištění stabilního letu s rychlou odezvou. Výpočet tahu motorů potřebných pro minimalizaci rozdílu požadovaného náklonu od aktuálního náklonu je zpracován použitím proporcionálně integračně derivačních regulátorů pro osu Roll Pitch a Yaw. Výstupní hodnotu z těchto regulátorů musíme zároveň ovlivnit požadovanou letovou výškou, která se získává pomocí rozdílu získaných dat altimetru vůči požadované výšce z řídicího rozhraní. Regulátory motorů, které zpracovávají požadavky pro námi požadovaný výkon dokáží tyto data zpracovávat při frekvenci 400Hz. Tímto získáváme dostatečně rychlou výpočtovou smyčku která ovlivňuje stabilní let quadcoptéry.



**Obr. 3.8:** Zjednodušený diagram řídicího algoritmu

Na zobrazeném zjednodušeném řídicím diagramu 3.8 je znázorněna implementace tohoto řešení řízení quadcoptéry. Spustíme mikroprocesor a následně nadefinujeme globální proměnné které využíváme v jednotlivých rutinách pro výpočty. Definujeme také strukturu "quad" do které se v průběhu zapisují aktuální informace o stavu, jako jsou požadované hodnoty náklonu v jednotlivých osách nebo také požadovaný výkon motorů před zásahem stabilizačních rutin. Dále se do této struktury zapisují aktuální informace z PID, senzorů a motorů.

Následně inicializujeme Timer 2 kanál 2, Timer 4 kanál 2 a Timer 9 kanály 1 a 2 které využíváme pro generování výstupního PWM signálu pro regulátory při frekvenci 400Hz. Dále inicializujeme čítač Timer 6, který nastavíme na čítání po 1 uS s přetečením 2500uS. Tento čítač je použit pro dekódování PPM signálu z přijímače od řídicího rozhraní. Přetečení 2500uS je z důvodu, že při delším pulzu bude následovat nové čítání od kanálu 1 po kanál 8. V případě přetečení se vyvolá přerušení, které nastaví globální proměnnou informující o aktuálním zapisovaném kanálu na kanál 1.

Poté inicializujeme hodnoty výstupního PWM, tak aby odesílaly požadovanou hodnotu 50 procent výkonu na regulátory všech čtyř motorů, neboli 1500uS délky setrvání pulzu v logické hodnotě 1, toto je nutné pro udržení regulátorů v uzamknutém stavu. Následně inicializujeme LCD pro zobrazování aktuálních informací o běhu programu a výpis případných chyb při inicializaci senzorů a dat z řídicího rozhraní.

Dále již inicializujeme sběrnici I2C a senzory připojené na tuto sběrnici s komunikační rychlostí 100kHz. Využíváme I2C3 pro komunikaci s LCD i senzory, proto již po inicializaci LCD je sběrnice aktivní a v tomto kroku pouze zkontrolujeme její aktivitu a nastavení. Toto je nutné, protože v případě že následně nebudeme potřebovat LCD, nebyla by I2C3 sběrnice aktivní, a quadcoptéra by nemohla létat bez získávaných dat ze senzorů.

Po kompletní inicializační rutině čekáme na požadavek aktivace z řídicího rozhraní. Toto je nutné pro zajištění bezpečnosti tohoto modelu, a to hlavně z důvodu zamezení možných zranění z nechtěného spuštění motorů. Poté co je detekován požadavek z řídicího rozhraní, uvedeme regulátory motorů do aktivního stavu. Regulátory se odemknou při odeslání několika pulzů s požadavkem na výkon motorů 0 procent, neboli 1000 uS délka trvání logické 1 na výstupu z PWM pro konkrétní regulátor.

Nyní máme aktivní regulátory motorů a již můžeme zpracovávané data ze senzorů a řídicího rozhraní zasílat na výpočet potřebných výkonových nastavení jednotlivých motorů. Data ze senzorů zpracováváme dle rutin popsanych v předchozích bodech této práce, tyto data se zpracovávají na základě vyvolávaných přerušeni asynchronně dle požadavků na čtení z jednotlivých senzorů.

V rutinách PID regulátorů probíhají hlavní výpočty nastavení motorů. PID regulátor dostane data o aktuální pozici quadcoptéry ze senzorů a také požadavky z řídicího rozhraní, dále má tento algoritmus také přístup k celé globálně dostupné struktuře obsahující konstanty pro P I a D člen osy Pitch Roll a Yaw. Do této struktury si zároveň ukládá průběžně zpracovaná data jako poslední rozdíl požadované a aktuální hodnoty nebo kumulativní součet rozdílu této hodnoty pro každou osu zvlášť. Za pomoci těchto dat PID regulátor použitý quadcoptéře pracuje na následujícím principu. Dle aktuálních požadovaných dat o náklonu v jednotlivých osách a dat ze senzorů určujících aktuální náklon v jednotlivých osách Pitch Roll a Yaw vypočte novou hodnotu tohoto rozdílu. Tuto hodnotu rozdílu vynásobí konstantou P. Následně se provede kumulativní součet této hodnoty vynásobený konstantou I s globálně dostupnou hodnotou tohoto součtu pro osu Roll Pitch a Yaw zvlášť. Tato výsledná hodnota členu I tohoto regulátoru je následně omezena maximální možnou hodnotou, z důvodu aby nám I člen nemohl integrovat na příliš vysoké hodnoty, které by následně mohli negativně ovlivňovat stabilitu za měnících se podmínek při letu. Dále se vypočte D člen regulátoru, který se vypočte rozdílem posledního rozdílu požadované a skutečné hodnoty náklonu, a následným vynásobením konstantou D. Toto se opět provede pro každou osu quadcoptéry zvlášť. Výstupem tohoto regulátoru je součet P I a D členu, který se porovná se přičte k aktuálně požadovanému výkonu motorů pro jednotlivé osy.

Výsledek z těchto regulátorů je zpracován rutinou odesílající data regulátorům, tato rutina dekoduje z požadavků informaci kterému motoru má nastavit jaký výkon, a tyto požadavky odešle ve formátu PWM pro každý motor zvlášť.

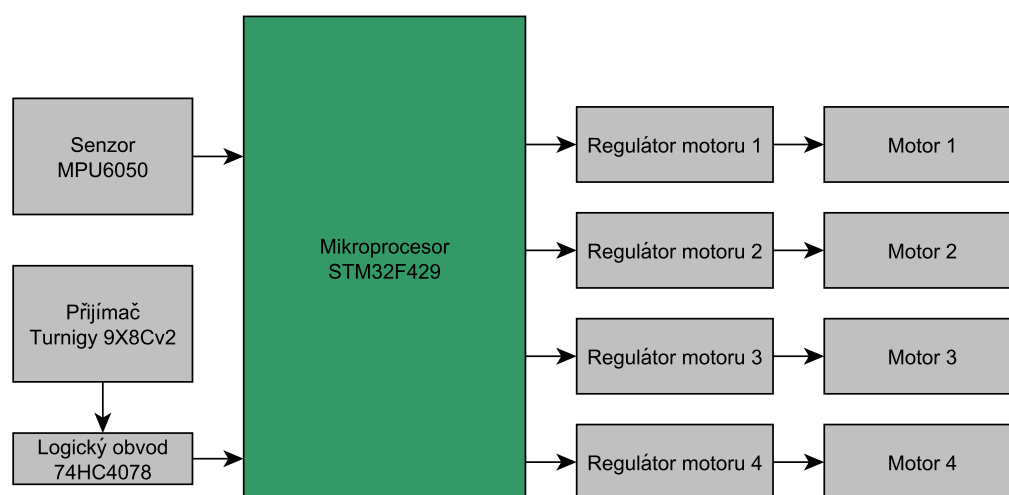
Tento celý postup výpočtů opakujeme pokaždé, kdy získáme nové data ze senzorů, výsledně tedy 200 krát za vteřinu neboli každých 5 ms. Tato výpočtová rutina má 4 cykly (20 milisekund) na stabilizaci požadovaných hodnot z řídicího rozhraní, než získá nové požadované data. Celkový čas výpočtů PID i s předáním výsledných dat pro regulátory motorů trvá 1180uS, proto máme ještě dostatečnou výkonovou rezervu pro možnosti dalšího rozšíření quadcoptéry.

Pro ošetření možných chyb v běhu programu a případných problémů s připojenými moduly je implementována rutina, která kontroluje zda přijímáme data ze všech připojených zařízení. V případě, že nastane chyba s přijímanými daty, nastavíme všechny motory na 0 procentní výkon pro zamezení možných bezpečnostních rizik při nekontrolovaném letu dle posledních přijatých informací ze senzorů nebo řídicího rozhraní. Tímto docílíme nekontrolovaný pád, proto ihned po vypnutí motorů restartujeme mikroprocesor a v případě že se spojení obnoví, rutiny pro stabilizaci ihned budou quadcoptéru opět stabilizovat.

## 3.7 Testování realizované quadcoptéry

### 3.7.1 Blokové schéma

Před testováním algoritmů a funkčnosti quadcoptéry je nutné ověřit správné zapojení všech částí. Proto tento krok provedeme jako první před připojením napájení. Blokové schéma 3.9 znázorňuje vzájemné propojení jednotlivých komponent quadcoptéry.



Obr. 3.9: Blokové schéma

Senzor MPU6050 je připojen pomocí sběrnice I2C k mikroprocesoru. Pin SCL který přenáší hodinové pulzy připojíme k pinu PA8 mikroprocesoru. Pin SDA který přenáší data po sběrnici připojíme k pinu PC9 mikroprocesoru. Pin INT vyvolává přerušování při připravených datech z DMP připojíme na pin PE2 mikroprocesoru. Napájení VCC tohoto senzoru připojíme na 3V pin vývojové desky s mikroprocesorem. Zemní pin připojíme na GND pin mikroprocesoru.

Přijímač Turnigy 9X8Cv2 připojíme následujícím postupem k logickému členu 74HC4078. Kanál 1 až 8 z tohoto přijímače zapojíme v libovolném pořadí k logickému obvodu na vstupy číslo 2,3,4,5,9,10,11,12. Pro napájení a uzemnění využijeme libovolný výstup regulovaného 5V napětí (červený) a zemního vodiče (černý) z regulátorů 1 až 4.

Logický obvod 74HC4078 má připojené vstupy z předchozího popisu zapojení přijímače. Následně připojíme napájení z přijímače a výstupní signálový vodič z pinu číslo 1 připojíme k pinu

PE3 mikroprocesoru.

Mikroprocesor má čtyři výstupní PWM signály pro regulátory motorů. Signál z pinů PB3, PB7, PE5 a PE6 připojíme na signálový bílý vodič regulátoru 1 až 4 v libovolném pořadí, protože jsou všechny tyto čtyři regulátory stejné. Mikroprocesor napájíme z libovolného regulátoru připojením na pin 5V mikroprocesoru a pin GND připojíme na černý zemnicí vodič regulátoru.

Regulátory mají vstupní napájení připojené z Li-Po baterie o hodnotě 14,8 V. Výstupní tři vodiče připojíme na vstupní napájecí tři vodiče BLDC Motorů v libovolné kombinaci, kterou po připojení quadcoptéry k napájení změníme, tak aby se vrtule otáčeli námi požadovaným směrem a to buď protisměru nebo po směru hodinových ručiček v závislosti na použitých vrtulích.

Motory jsou připojeny z regulátorů, avšak musíme zkontrolovat jejich umístění dle vstupních signálů regulátorů z mikroprocesoru. Motor 1, přední levý je napájen regulátorem se vstupním PWM signálem z pinu PB3 mikroprocesoru. Motor 2, přední pravý je napájen regulátorem který je připojen na pin PB7 mikroprocesoru. Motor 3 je pravý zadní a je připojen na regulátor se signálem z pinu PE5 mikroprocesoru. Poslední motor 4, levý zadní je připojen na regulátor se signálem pinu PE6 mikroprocesoru.

Teprve po zkontrolování zapojení všech modulů, můžeme zapojit akumulátor.

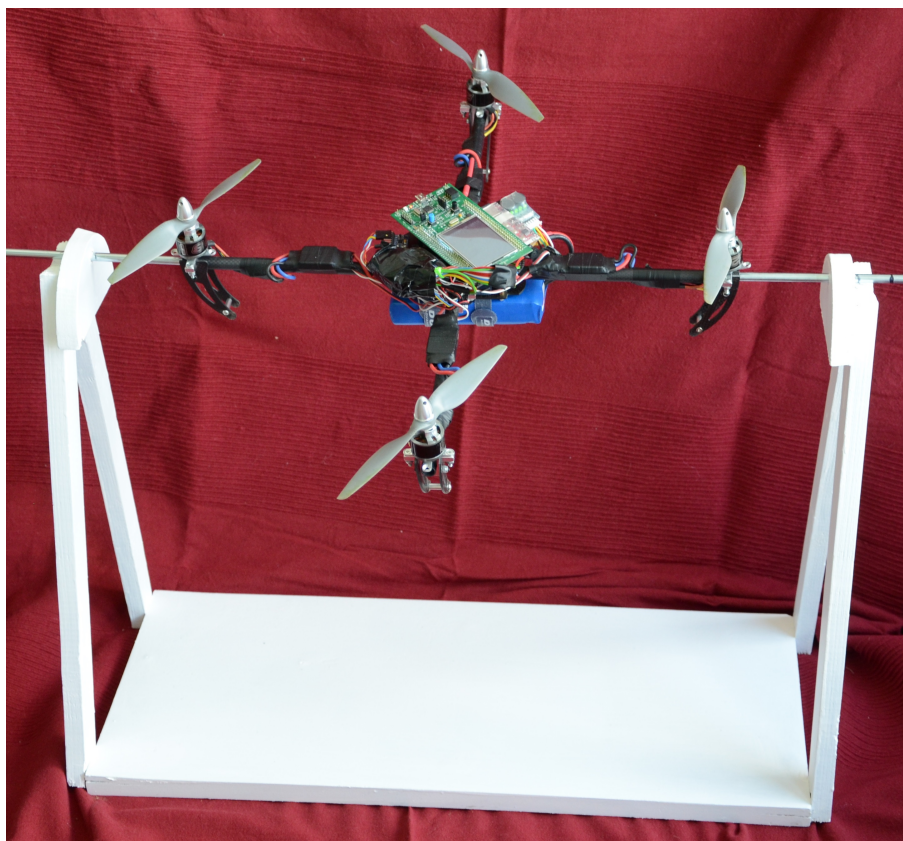
### 3.7.2 Testování motorů a ladění PID

Po úspěšné kontrole zapojení všech částí a jejich rozmístění se dostáváme k testování motorů. Pro testování máme možnost zkoušet quadcopter v místnosti, což důrazně nedoporučuji z důvodu možných vážných poranění od rotujících vrtulí které mají obvodovou rychlost 500-720km/h dle velikosti a použitých akumulátorů. Testování mimo uzavřený prostor také není vhodné pro první pokusy vyzkoušení stabilizace a rotace motorů, hlavně z důvodu možného nekontrolovatelného pohybu quadcoptéry. Proto jsem vytvořil testovací držák ze dřeva 3.10, do kterého se dá quadcoptéra připnout za 2 ramena, čímž docílíme pohyb pouze v ose Roll nebo Pitch dle uchycení.

V tomto mnou vyrobeném držáku již bez možných rizik testujeme směr otáčení vrtulí. V případě že se vrtule točí obráceně, než potřebnou rotací pro docílení tahu, který tuto quadcoptéru zvedá. Zaměníme 2 vodiče tohoto motoru, což způsobí změnu fáze a motor se bude otáčet obráceně námi požadovaným směrem otáček. Tento postup provedeme pro všechny čtyři motory.

Nyní máme nastavené motory pro vytvoření tahu správným směrem. Můžeme nastavit PID pro jednotlivé osy Roll a Pitch. Připevníme quadcoptéru tak aby se umožnil pohyb pouze v námi zvolené ose. A nastavíme konstanty PID na  $P=1$   $I=0$   $D=0$ , takto upravený program nahrajeme do mikroprocesoru a vysílačkou budeme ovládat požadovaný tah motorů při ponechání ovládacího prvku pro nastavení náklonu v ose Pitch a Roll na 0 stupňů. Quadcoptéra by se měla snažit docílit vyrovnaní vodorovné pozice vůči zemi. Avšak s těmito konstantami nám regulační smyčka PID regulátoru pracuje pouze jako P regulátor. Tuto konstantu upravujeme do té doby, než quadcoptéra dokáže docílit námi požadovaný náklon. Následně zjistíme že v této konfiguraci nám nastává oscilace quadcoptéry v této ose. Proto nyní nastavíme konstantu D která nám začne brzdít motory při blížení se k požadované hodnotě náklonu. Po úspěšném nastavení konstant P a D quadcoptéra dokáže docílit požadovaný náklon. Avšak nyní musíme přidat integrační konstantu I, z důvodu





**Obr. 3.10:** Držák pro testování quadcoptéry

aby mohla quadcoptéra docílit přesné hodnoty v případě že se nedokáže stabilizovat na námi požadovaný náklon. Tato konstanta dorovná potřebný rozdíl, který nedokážeme stabilizovat pomocí P a D členu. Tento proces je při použití klasického nahrávání konstant do mikroprocesoru velmi zdlouhavý, proto jsem po několika testech udělal úpravu programu, kde jsem nastavil vstupní 2 kanály vysílačky na potenciometry. Těmito dvěma potenciometry jsem nastavoval mnou požadované hodnoty konstant a zároveň je zobrazoval na display mikroprocesorového kitu. Po docílení stabilního náklonu s rychlou odezvou jsem tyto konstanty zadal do mikroprocesoru přímo. Tímto postupem pro všechny osy a každou konstantu.

### 3.7.3 Možnosti realizované quadcoptéry

Po implementaci kompletního algoritmu a naladění PID konstant s testováním na držáku, přišel na řadu testovací let, odkaz tohoto letu je na obrázku 3.11 . Při testování quadcoptéry na držáku zanedbáváme Yaw složku rotace modelu. Proto se může stát, že bude quadcoptéra rotovat. Rotace je způsobeno především nepřesným náklonem motorů. V případě, že je motor nakloněn směrem doleva nebo doprava, bude tento model tímto směrem rotovat. Tohoto problému se zbavíme mechanicky srovnáním motorů tak, aby byli kolmo k zemi s použitím vodováhy. Mnou realizovaná quadcoptéra dokáže létat dle požadavků z řídicího rozhraní, avšak je nutné upravit citlivost nastavených hodnot na řídicím rozhraní. Zejména z důvodu, že i naklopení modelu o několik málo stupňů způsobí pohyb tímto směrem značnou rychlostí.



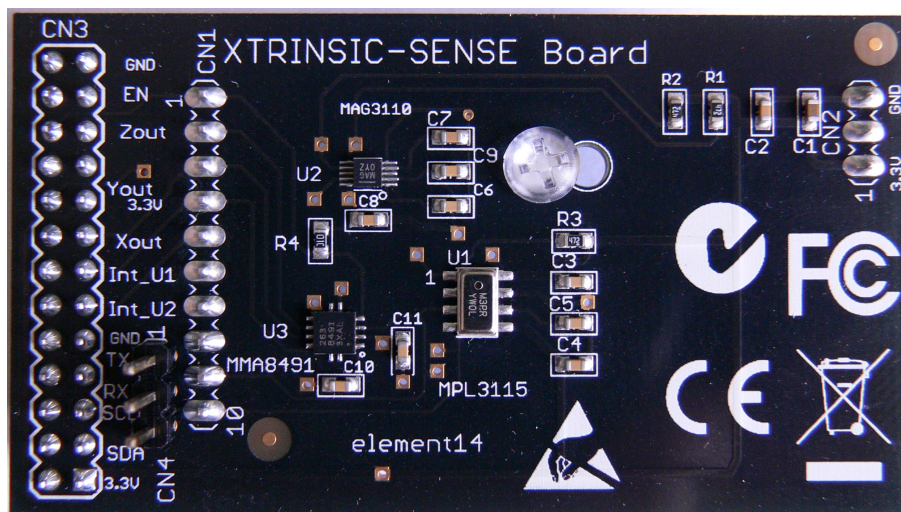
**Obr. 3.11:** Testování realizované quadcoptéry

Tato mnou realizovaná quadcoptéra dokáže létat v závislosti na požadovaném směru a výšce z řídicího rozhraní. S použitím vrtulí 8045 dokáže unést navíc přibližně 400 gramů zátěže bez ovlivnění manévrovacích schopností. Proto je možno tuto quadcoptéru vybavit větší kamerou či fotoaparátem. Já používám na této quadcoptéře miniaturní HD kameru o hmotnosti 30 gramů, která postačuje mým účelům. Doba letu tohoto realizovaného modelu závisí spíše na schopnostech pilota a to především z důvodu, že realizovaná quadcoptéra nemá implementované ultrazvukové detektory, které by zabránily možným kolizím s předměty. Vypočtená délka letu za běžných podmínek je přibližně 10 minut. Při nízkém napětí začne quadcoptéra pomocí připojeného modulu vydávat hlasitý tón, který nám signalizuje vybití akumulátoru. Bezprostředně po zaznění tohoto tónu je nutno s quadcoptérou přistát, jinak dojde k nekontrolovatelnému pádu. Myslím si, že po snížení citlivosti řídicího rozhraní je možné tuto quadcoptéru v tomto stavu běžně používat pro focení z výšek, natáčení videí krajiny nebo pouze létání za účelem zábavy. Případně při přidání dalších modulů, které popisuji v poslední části této práce, je možné použít tuto quadcoptéru pro dopravu menších předmětů ze startovací lokace do cílové lokace stanovené GPS souřadnicemi. S ohledem na velikost této quadcoptéry však není možné použití v uzavřených prostorech, a to především z důvodu velmi rychlého pohybu při požadavku na přemístění se a absence detektorů, které by quadcoptéře pomohly k orientaci se v místnostech.

## 3.8 Možnosti rozšíření realizované quadcoptéry

### 3.8.1 Modul kompasu a altimetru

Pro měření aktuální výšky quadcoptéry a přesného směru natočení jsem zvolil modul sensorů XTRINSIC-SENSE Board 3.12, který je k zakoupení na [www.farnell.com](http://www.farnell.com). Tento modul obsahuje vysoce přesné MEMS senzory nové generace. Například MPL3115 Altimetr, který dle datasheetu umí určit výšku s přesností 6cm. Tento altimetr použijeme k výpočtu autonomního nastavení letové výšky v závislosti na aktuální předvolbě z řídicího rozhraní. Dále MAG3110 magnetometr s měřením geomagnetického pole ve třech osách. Tento magnetometr využijeme k získání větší přesnosti určení rotace Yaw. Dále tento modul obsahuje také MMA8491Q akcelerometr, jenž nevyužijeme z důvodu již implementovaného MPU6050. Tento modul komunikuje pomocí I2C sběrnice, a proto jej připojíme k již používané I2C3 sběrnici na mikroprocesoru.



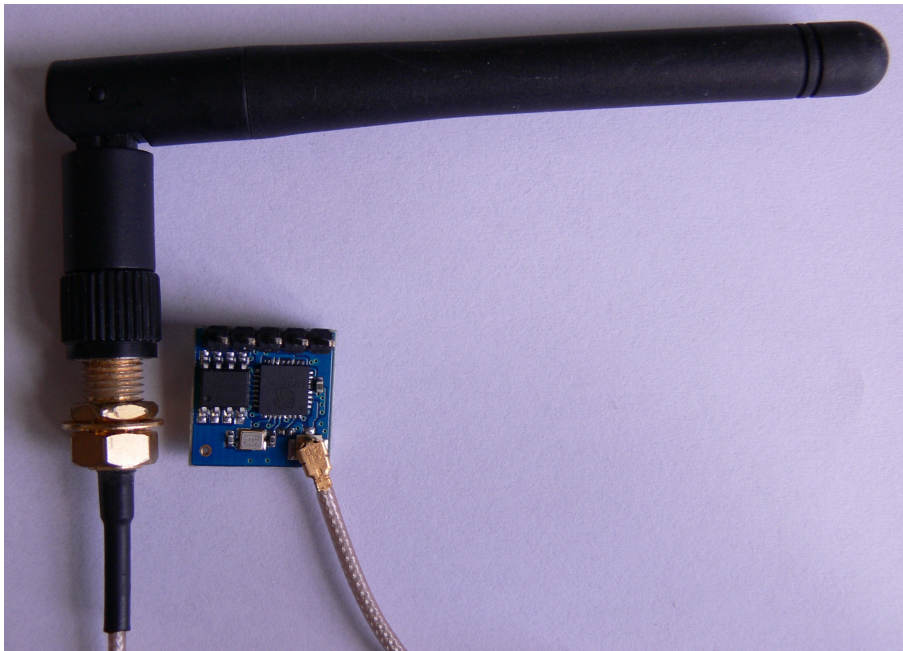
Obr. 3.12: XTRINSIC-SENSE Board

### 3.8.2 WiFi modul

Pro komunikaci s PC, případně s mobilním telefonem, nebo jiným zařízením využívající 2.4 Ghz WiFi použijeme modul ESP8266 3.13. Tento modul zpracovává data zaslaná po UART periférii a odešle na námi určené zařízení připojené do wifi sítě vytvořené tímto modulem. Případně se pomocí AT příkazů zadaných tomuto modulu připojíme na dostupnou WiFi síť a můžeme získávat informace z internetu nebo je tam odesílat.

Po připojení zařízení k inicializovanému UART mikroprocesoru stačí zaslat tyto AT příkazy dle datasheetu [6] .

- AT - vrátí OK v případě, že zařízení komunikuje
- AT+RST - restartuje modul



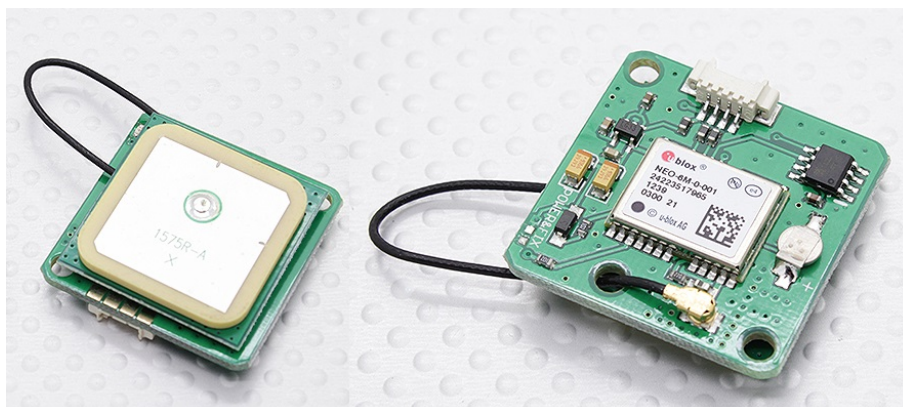
**Obr. 3.13:** WiFi modul ESP8266

- AT+CWMODE=3 - přepne modul do režimu client a zároveň Access Point
- AT+CWJAP - vrátí nalezené WiFi sítě v dosahu
- AT+CWJAL="jméno\_sítě", "heslo\_sítě" - připojí se k síti
- AT+CIFSR - vrátí přiřazenou IP tomuto zařízení

Po úspěšném odeslání těchto AT příkazů, jsme se připojili k WiFi síti. A již je pouze na nás, jak tuto možnost využijeme. Můžeme zasílat data, nebo je přijímat od jiných zařízení v této síti, případně na internetu. Tímto řešením se nám otevřeli nové možnosti použitelnosti quadcoptéry.

### 3.8.3 GPS modul

Přidáním modulu GPS na stávající realizovanou quadcoptéru by jsme získali přesnou aktuální pozici modelu. Následně se dá navigovat quadcoptéra na určené cílové souřadnice. A to za použití například WiFi modulu pro snadné definování cílových souřadnic. Jako vhodný modul lze doporučit "NEO-6M GPS Module"<sup>3.14</sup>, který je běžně dostupný. Tento modul komunikuje pomocí UART. Má integrovanou anténu a proto není nutné řešit další součástky pro implementaci tohoto modulu. Navíc tento modul má velmi malé rozměry 22x30x13 mm a hmotnost pouze 12 gramů. Při těchto miniaturních rozměrech dokáže pracovat až s 50-ti kanály a po prvním zaměření, které dokáže již za 1 vteřinu, odesílá při frekvenci 5hz aktuální souřadnice. Tyto souřadnice jsou s přesností až 2 metry pro horizontální určování polohy.



Obr. 3.14: NEO-6M GPS Modul [2]

### 3.8.4 Xbee modul

Další možností komunikace quadcoptéry a řídicího rozhraní je modul Xbee 3.15. Tento modul pracuje na frekvenci 2.4Ghz a komunikuje pomocí UART rozhraní. S tímto modulem můžeme spojit quadcoptéru s řídicím rozhraním, nejlépe mikroprocesorem s připojeným Joystickem na vzdálenost 90 metrů v exteriéru a při použití Xbee PRO verze až 1600 metrů.



Obr. 3.15: Xbee modul (vlevo) a Xbee PRO modul [3]

## 4. Závěr

Cílem práce byla realizace létajícího prostředku typu "quadcopter". Pro dosažení tohoto cíle bylo nutné porozumění možných mechanických řešení létajících prostředků tohoto typu. A především vhodnosti užití jednotlivých dílů jako jsou motory a vrtule pro různé hmotnosti výsledného výrobku. V dnešní době se tyto quadcoptéry prodávají jako hotové funkční výrobky a až na ojedinělé typy také bez možných zásahů do řídicích algoritmů. Na možných otevřených řešeních pro úpravy typu OpenSource se následně podílí desítky vývojářů a bohužel bez kvalitní dokumentace. Proto tyto projekty nelze použít jako odrazový můstek pro vývoj bez kompletního porozumění veškerým implementovaným algoritmům, jenž jsou většinou psané pro 8 bitové mikrokontroléry v jazyce C++ s využitím velkého množství knihoven různého původu.

Proto jsem v mém řešení realizované quadcoptéry vycházel především ze základních principů, výslednou implementaci řešení jsem začal stavět od jednoduchých částí a upravoval do dosažení mých požadavků. Z tohoto důvodu jsem musel vyzkoušet mnoho konfigurací a najít tu nejvhodnější empirickým vývojem. Vzhledem k tomu, že jsem všechny součástky použité ve všech testovacích prototypch nakupoval výhradně ze zahraničí, měl jsem na výběr z opravdu velmi širokého spektra možných konfigurací. Základní hranici pro výběr z možných komponent jsem si proto určil cenu kompletního řešení a to deset tisíc korun. Avšak nejen z důvodu průběžných modifikací původně zvolené konfigurace, ale především z důvodu poškození jednotlivých komponent při testování stabilizačních algoritmů se výsledná cena vývoje vyšplhala na nejméně trojnásobek původní předpokládané ceny kompletního řešení. Tímto jsem si ověřil, že vývoj finálního výrobku od základu je nejen časově, ale také finančně náročný.

V této práci jsem proto popsal jakým způsobem je vhodné vybírat jednotlivé komponenty pro výsledné řešení. A také mnou použité komponenty, které již mám otestované pomocí finální verze algoritmů v realizované funkční quadcoptéře. Dále jsem vyzkoušel různé možnosti algoritmů zpracovávající data ze senzorů pro výsledné určení aktuálních náklonů vůči zemi ve všech osách. Například jsem ověřil implementaci použití komplementárního filtru pro výpočet náklonu v jednotlivých osách z dat akcelerometru a gyroskopu. Toto řešení v mnou určeném cíli bylo nedostačující, a to především z důvodu narůstající integrační chyby měření, která by při delším provozu quadcoptéry způsobila nestabilitu a případně až selhání možnosti řízení tohoto modelu. Následně jsem prostudoval použití Kalmanovo filtru pro určení přesného aktuálního náklonu, toto řešení je výrazně přesnější, avšak také náročnější na implementaci. Avšak nejlepším možným řešením získáváním dat o aktuálním náklonu byla implementace nového typu senzoru MPU6050 od společnosti InvenSense který obsahuje DMP jednotku. Tato DMP jednotka s použitím vnitřních algoritmů dokáže

zpracovávat data z akcelerometrů gyroskopů v tomto senzoru a také automaticky kompenzovat chyby a automaticky se sama kalibrovat. Výstupem z této DMP jednotky jsou již přesná kalibrovaná data ve formátu quaternionů.

Použití této jednotky DMP bylo však komplikované, a to nejen z důvodu že se jedná novou funkci těchto senzorů o které se nedá jednoduše dohledat příklad jejího použití. Ale také z důvodu, že samotný výrobce InvenSense zdrojové kódy pro aktivaci DMP uvolňoval pouze jeho odběratelům s licenčními omezeními použití. Nicméně se výrobce tohoto senzoru v nedávné době rozhodl změnit licenční podmínky použití jeho knihoven a uvolnil tyto knihovny s názvem "Motion Driver 5" jako OpenSource pro mikroprocesory Texas Instruments MSP430. Ovšem jejich uvolněné zdrojové kódy pro použití sebou přinesly další problém, a to kromě použití jiného než mně dostupného překladače také to, že jejich uvolněné řešení se snaží ukázat maximální potenciál tohoto senzoru bez ohledu na optimalizaci pro zpracování dat v mikrokontroléru. Data se totiž odesílají pomocí USART do PC, kde dochází k jejich následnému zpracování a vykreslení. Proto jsem musel kompletně tyto knihovny přeportovat pro použití ve vývojovém prostředí Keil uVision a zároveň upravit pro možnost použití i zpracování v mnou zvoleném mikroprocesoru STM32F429. Po několika měsících vývoje pro získávání co nejpřesnějších dat o aktuálním náklonu quadcoptéry vůči zemi se mi povedlo získávat přesná data s použitelnou přesností na setinu stupně náklonu v ose Pitch a Roll. Pro získání této přesnosti i v ose Yaw by bylo možné použití magnetometru a tím rozšíření ze šesti stupňů volnosti na devět stupňů volnosti. Pro tento účel jsem zakoupil MPU9250, který je nástupce MPU6050 rozšířený o magnetometr a výstupem jsou i přesnější data ve všech devíti osách. Nicméně bohužel mi tento senzor dorazil ze zahraničí vadný a již jsem další kus nezakupoval, s tím že pro můj cíl bude postačovat funkčních šest stupňů volnosti.

Po úspěšném získávání přesných dat o aktuální poloze quadcoptéry 200 krát do vteřiny jsem začal implementovat řídicí algoritmy pomocí PID regulátorů. Funkčnost těchto algoritmů jsem testoval na mém testovacím stojanu, do dosažení rychlé odezvy a stabilního udržení požadovaného náklonu v ose Roll a Pitch. Při testování těchto algoritmů bylo zjištěno, že je nevhodné předimenzovávat maximální tah quadcoptéry. A to především z důvodu, že se tímto sníží rozsah možného nastavení potřebného výkonu nutného pro stabilizaci požadovaného náklonu. Proto jsem opět změnil konfiguraci, a zvolil menší průměr vrtulí. Tímto krokem jsem snížil maximální tah celkové soustavy, avšak mohu jemněji nastavovat tah jednotlivých motorů, protože vypočtený tah nutný pro udržení konstantní výšky je nyní na přibližně 50-ti procentech výkonu motorů. V předchozím testovaném řešení byl již přibližně na 20-ti procentech výkonu motorů, proto bylo toto původní řešení nestabilní.

S finálním algoritmem jsem provedl několik testovacích letů, a pro mnou stanovený cíl je toto řešení dostačující. Avšak realizované řešení quadcoptéry má potenciál na možné další rozšíření pomocí přídatných modulů které jsou v práci také popsány.

# Literatura

- [1] STMicroelectronics. *STM32F4 Series*. 2015 [cit. 2015-05-11]. Dostupné z:  
<http://www.st.com/web/en/catalog/mmc/FM141/SC1169/SS1577?sc=stm32f4>
- [2] Hobbyking.com *Products reference manuals*. 2015 [cit. 2015-05-11]. Dostupné z:  
<http://www.hobbyking.com>
- [3] Digi International® Inc. *Product reference manual*. 2015 [cit. 2015-05-11]. Dostupné z:  
<http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-series1-module>
- [4] eCalc *xcopterCalc*. 2015 [cit. 2015-05-11]. Dostupné z:  
<http://www.ecalc.ch/xcoptercalc.php>
- [5] Wikipedia *Quadcopter*. 2015 [cit. 2015-05-11]. Dostupné z:  
<http://en.wikipedia.org/wiki/Quadcopter>
- [6] Nurds *ESP8266*. 2015 [cit. 2015-05-11]. Dostupné z: <https://nurdspace.nl/ESP8266>
- [7] Invensense *MPU6050 Datasheet*. 2015 [cit. 2015-05-11]. Dostupné z:  
<http://www.invensense.com/mems/gyro/mpu6000.html>