

**Univerzita Hradec Králové**  
**Přírodovědecká fakulta**

**DIPLOMOVÁ PRÁCE**

**2021**

**Bc. Tomáš Musil**

**Univerzita Hradec Králové**  
**Přírodovědecká fakulta**

**Analýza letu jednoduché rakety za pomoci  
zabudovávaných senzorů**

**Diplomová práce**

Autor: Bc. Tomáš Musil  
Studijní program: N1701 Fyzika  
Studijní obor: Učitelství fyziky a informatiky pro střední školy

Vedoucí práce: Mgr. Petr Coufal  
Oponent práce: Ing. Petr Voborník, Ph.D.



## Zadání diplomové práce

<b>Autor:</b>	<b>Tomáš Musil</b>
Studium:	S18FY010NP
Studijní program:	N1701 Fyzika
Studijní obor:	Učitelství fyziky pro střední školy, Učitelství pro střední školy - informatika
<b>Název diplomové práce:</b>	<b>Analýza letu jednoduché rakety za pomoci zabudovaných senzorů</b>
Název diplomové práce AJ:	Simple rocket flight analysis using built-in sensors

### **Cíl, metody, literatura, předpoklady:**

Cílem diplomové práce je zkonstruovat model rakety osazené senzory a mikropočítači. V rámci teoretické práce budou popsány technologie, které budou použity pro získání letových hodnot. V rámci praktické práce bude popsána konstrukce rakety a hardwaru, tvorba softwaru a samotné zpracování získaných hodnot. Součástí práce je výukový materiál pro obohacení výuky informatiky a fyziky.

SVOBODA, Emanuel. Přehled středoškolské fyziky. 4., upr. vyd. Praha: Prometheus, 2005, 531 s. ISBN 80-7196-307-0. Air Command Water Rocket ? 2006-2018 [online]. Poslední revize 25. 5. 2011 [cit. 10. 7. 2018]. Dostupné z: <http://www.aircommandrockets.com> National Physical Laboratory. A guide to building and understanding the physics of Water Rockets [online]. Poslední revize 1. 7. 2007 [cit. 8. 7. 2018], verze 1.02, 50 s., Dostupné z: [http://www.npl.co.uk/upload/pdf/wr\\_booklet\\_print.pdf](http://www.npl.co.uk/upload/pdf/wr_booklet_print.pdf) VODA, Zbyšek. Průvodce světem Arduina. Vydání druhé. Bučovice: Martin Stříž, 2017. 239 stran. ISBN 978-80-87106-93-8.

Garantující pracoviště: Katedra aplikované kybernetiky,  
Přírodovědecká fakulta

Vedoucí práce: Mgr. Petr Coufal

Oponent: Ing. Petr Voborník, Ph.D.

Datum zadání závěrečné práce: 9.4.2019

## **Čestné prohlášení**

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a že jsem v seznamu použité literatury uvedl všechny prameny, z kterých jsem vycházel.

V Hradci Králové dne:

Podpis:

## **Poděkování**

Rád bych vyjádřil poděkování svému vedoucímu práce za vedení a pomoc při psaní diplomové práce.

## **Anotace**

MUSIL, Tomáš. *Analýza letu jednoduché rakety za pomoci zabudovaných senzorů*. Hradec Králové, 2021. Diplomová práce na Přírodovědecké fakultě Univerzity Hradec králové. Vedoucí diplomové práce Mgr. Petr Coufal. 94 s.

Diplomová práce popisuje práci s vodní raketou, která má v hlavici měřící zařízení se senzory ovládané Arduinem. Získaná letová data, jsou využita k analýze letu. Jednotlivé kroky tvorby a analýzy jsou vysvětleny, tak aby je bylo možné snadno napodobit. Využití práce je především vhodné v rámci rozšíření a ozvláštnění výuky informatiky a fyziky na základní a střední škole.

## **Klíčová slova**

Analýza, Arduino, fyzika, informatika, senzory, střední škola, vodní raketa, výuka, základní škola

## **Annotation**

MUSIL, T. *Simple rocket flight analysis using built-in sensors*. Hradec Králové, 2021. Diploma Thesis at Faculty of Science University of Hradec Králové. Thesis Supervisor Mgr. Petr Coufal. 94 s.

The Diploma Thesis describes the work with a water rocket, which has a measuring device with sensors controlled by Arduino in the nose cone. The obtained flight data are used for flight analysis. The individual steps of creation and analysis are explained so that they can be easily replicated. The use of the work is especially suitable in the extension and enrichment of the teaching of computer science and physics in primary and secondary school.

## **Keywords**

Analysis, Arduino, physics, computer science, sensors, high school, water rocket, teaching, elementary school

# Obsah

Úvod .....	10
Teoretická část .....	11
1 Jednoduchá raketa .....	12
1.1 Octové a vodní rakety .....	13
1.2 Vylepšení .....	13
1.2.1 Trup rakety .....	13
1.2.2 Hlavice .....	14
2 Arduino .....	15
2.1 Arduino Nano .....	16
2.2 Vývojové prostředí (IDE) .....	18
2.3 Přídavné moduly .....	19
2.4 Shieldy .....	20
2.4.1 Nepájivé pole .....	21
2.4.2 Vodiče .....	22
2.5 Senzory .....	23
2.5.1 Akcelerometr .....	23
2.5.2 Gyroskop .....	24
2.5.3 Barometr .....	26
2.5.4 I <sup>2</sup> C .....	27
2.6 Napájení .....	29
2.6.1 Baterie .....	29
2.7 Servomotory .....	32
2.8 Ukládání dat .....	34
3 Práce s daty .....	35
3.1 Nastavení .....	35
3.2 Kalibrace .....	36
3.3 Filtrace .....	36
4 Využití práce v rámci výuky .....	38
4.1 Základní škola .....	38
4.1.1 Informatika .....	38
4.1.2 Fyzika .....	39



4.2	Střední škola .....	39
4.2.1	Informatika .....	40
4.2.2	Fyzika .....	40
	Praktická část.....	41
5	Tvorba rakety.....	42
5.1	Trup.....	42
5.1.1	Spojování.....	43
5.2	Hlavice.....	45
5.2.1	Padák .....	46
5.2.2	Osazení pro měřící zařízení a padák.....	47
5.3	Letky – stabilizátory .....	50
5.4	Tryska.....	51
5.5	Kompletace rakety.....	52
5.5.1	Parametry rakety TOM II .....	55
5.6	Úpravy odpaliště.....	55
6	Měřící zařízení – hardware.....	58
6.1	Servomotor .....	58
6.2	Senzory .....	58
6.2.1	MPU-9250 (IMU) .....	58
6.2.2	BME280 (Barometr + Teploměr) .....	59
6.3	Napájení.....	59
6.4	Zapojení .....	60
6.4.1	Schéma .....	60
7	Měřící zařízení – software.....	61
7.1	Rozbor kódu.....	61
7.1.1	Inicializační část.....	61
7.1.2	Nastavovací část.....	62
7.1.3	Část datového sběru.....	65
8	Další možnosti (rozšíření) .....	68
9	Odpal a měření.....	69
9.1	Analýza dat letu .....	69
9.1.1	Čas .....	70
9.1.2	Kalibrace akcelerometrických dat .....	71

9.1.3	Zrychlení .....	72
9.1.4	Rychlost .....	73
9.1.5	Dráha .....	74
9.1.6	Gyroskopická data – rotace rakety.....	75
9.1.7	Barometrická data .....	79
9.1.8	Tah.....	81
9.2	Parametry letu .....	82
9.3	Cena .....	83
10	Využití ve výuce (metodická část).....	84
	Závěr .....	85
	Seznam použité literatury .....	86
	Seznam a zdroje obrázků .....	89
	Seznam grafů.....	94
	Příloha	

# Úvod

Diplomová práce navazuje na myšlenky z mé bakalářské práce [1]. V ní jsem se zaměřoval na fyzikální principy, které jsou spjaty s letem raket a následně je prakticky otestoval sestavením a měřením na modelu. Způsob měření probíhal natáčením letu modelu pomocí vysokorychlostní kamery, který se pak analyzoval ve specializovaném softwaru a výsledky se porovnávaly s teoretickými předpoklady. Získané výsledky nebyly příliš uspokojivé a zároveň jsem zjistil omezení, které daná metoda měření má.

Došel jsem tedy k závěru, že by bylo možné znatelně vylepšit přístup, jak získat letové hodnoty za pomoci senzorů, které by byly přímo zabudované v hlavici rakety. Samotný nápad jsem získal na předmětu robotiky, kde jsem se setkal s Arduinem. Rozhodl jsem se tedy více zaměřit na technickou část, která by mi pomohla získat lepší výsledky a případně mi pomohla uskutečnit funkce, které jsem bez ní provést nemohl.

Práce by měla sloužit i jako rozšiřující pohled na propojení předmětů informatiky a fyziky. Proces vlastního měření, za pomoci žáky vytvořené rakety a měřícího zařízení, by tak měl více umocnit získané znalosti a dovednosti než při klasické výuce, kde se hodnoty udávají primárně ze zadání úlohy (např. cyklista jel rychlostí 10 m/s, jakou dráhu urazil, pokud...).

V rámci teoretické části jsem se nejprve zaměřil na stěžejní prvek práce, což je právě samotná raketa. Dále pak na sounáležitosti spjaté s elektronikou, která má za úkol získat letová data. Jedná se o nejpodstatnější posun vůči mé bakalářské práci, a proto jsem mu věnoval zásadní pozornost. Též jsem se zaměřil na teorii práce s daty. Jako poslední jsou zmíněné možnosti využití tématu diplomové práce v předmětech informatika a fyzika s ohledem na rámec RVP.

V praktické části byl vytvořen ucelený popis tvorby rakety, použitých hardwarových komponentů a kódu programu. Hlavním obsahem je samotná ukázka analýzy dat, kde jsou jednotlivé kroky popsány, tak aby posloužily jako případný návod pro zájemce, kteří by chtěli dojít k vlastním výsledkům. Jsou zde doplněny parametry letu, odhad ceny komponentů a nápady na případné vylepšení. V rámci metodické části jsou poskytnuty materiály pro podporu při výuce.

# Teoretická část

Cíle teoretické části:

- A. Vymezení pojmu jednoduché rakety využité v této práci.
- B. Vytvoření uceleného popisu jednotlivých elektronických komponentů použitých při tvorbě měřicího zařízení využitého pro sběr dat při letu rakety.
- C. Popsat využití tématu diplomové práce v předmětech informatika a fyzika s ohledem na rámec RVP.

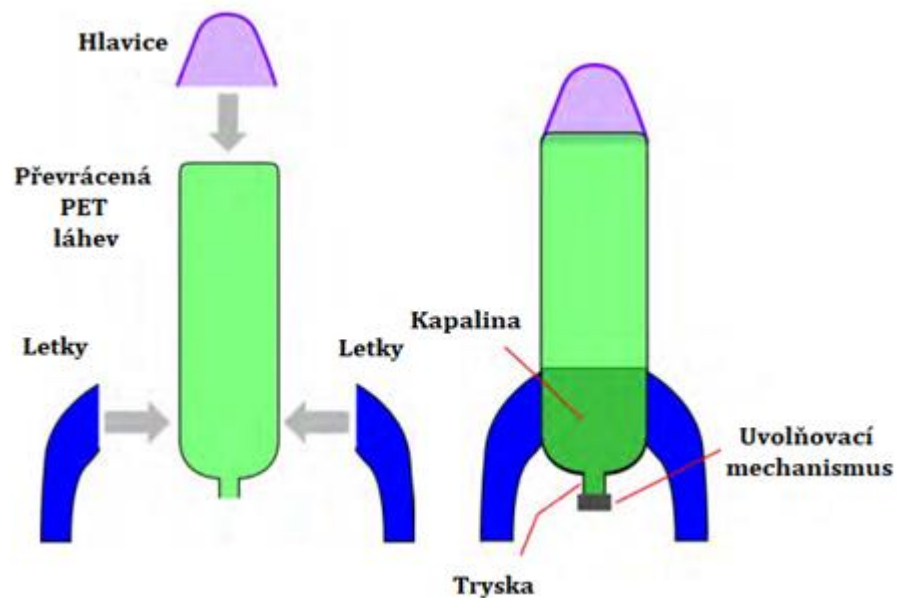
Metody a postupy dosažení cílů:

- Pro dosažení cíle A. se rozšíří množství poznatků popsanych v [1], tak aby více odpovídaly záměrům této práce.
- Pro dosažení cíle B. je důležité shromáždění ucelených informací o jednotlivých komponentech měřicího zařízení, tak aby mohly být následně v práci představeny a popsány.
- Pro dosažení cíle C. bude použita metoda analýzy dokumentů. V tomto případě se bude jednat o dokument Rámcově vzdělávací program pro základní vzdělávání a dokument Rámcově vzdělávací program pro gymnázia.

# 1 Jednoduchá raketa

Jednoduchá raketa je termín, který jsem osobně využil pro diferenci od běžné představy raket, jakožto nákladných a precizních strojů, které obdivujeme od dob dobývání vesmíru. Pro možnost užít si stavbu a let vlastních raket, aniž bychom si museli sáhnout hluboko do kapsy a nepotřebovali rozsáhlé stavební znalosti vznikly již před 50 lety právě například vodní rakety.

V nejjednodušší formě se jedná o převrácenou PET láhev, na kterou je přidělaná hlavice (jednoduše udělaná z jiné PET lahve) a stabilizátory (letky). V lahvi je obsažena kapalina (voda) a stlačený plyn (vzduch). Rychle unikající kapalina, která je tlačena roztahujícím se plynem urychluje raketu. K úniku kapaliny a plynu dochází skrze trysku, která proudící kapalinu směřuje pro maximální efektivitu. K tomu abychom mohli kontrolovat, kdy dojde k odpalu rakety využíváme uvolňovací mechanismy. [2]



Obrázek č. 1 – Vodní raketa jednoduše

Vodní rakety jsou součástí celosvětové zábavy již od roku 1973, kdy se šumivé nápoje začali dávat do PET lahví. Jedná se o opravdu úžasnou pomůcku pro výuku fyziky a informatiky. Pro žáky základních a středních škol (i vysokých) je tak možné si prakticky osvojit teoretické poznatky z fyziky jako jsou: setrvačnost, gravitace, odpor vzduchu, Newtonovy zákony pohybu, zrychlení, vztahy mezi prací a energií nebo impuls a hybnost, pohyb střely, výpočty volného pádu, vnitřní a vnější balistika. Či informatiky: sběr dat, zpracování velkého množství dat, tvorba měřícího zařízení, programování, pájení a mnoho dalších. [3]

Kromě výuky se dají vodní rakety využívat pro pořádání soutěží. Týmy a jedinci, tak vytvářejí rakety, které jsou nejen efektivní, ale i schopné určovat výšku vzletu, délku

doletu a mnoho dalších informací, tak aby bylo možné určit vítěze. [3] Příkladem je například [4], která je pořádaná již od roku 2003.

## 1.1 Octové a vodní rakety

Kromě vodních raket se můžeme setkat i s raketami octovými. Hlavním rozdílem octové rakety vůči vodní je právě v obsažené kapalině (x% ocet/voda) a stlačeným plynem (oxid uhličitý/vzduch). Kromě toho je rozdíl ve způsobu, jakým stlačený plyn vzniká. U octové rakety se musí smísit kyselina octová s hydrogenuhličitanem sodným (jedlou sodou), čímž vznikne chemická reakce, při které vzniká velké množství oxidu uhličitého. Zatímco u vodní rakety se využívá hustilka, či láhev se stlačeným vzduchem. U octové rakety se musí opatrně řešit předletová příprava. Po smíchání octa a jedlé sody začíná docházet k chemické reakci ihned a je potřeba PET láhev (trup rakety), co nejrychleji hermeticky (vzduchotěsně) uzavřít. To nemusí být úplně snadné, a navíc se může jednat o poměrně nebezpečnou operaci, kdy dochází k přímé manipulaci s postupně se natlakovaným objektem (tlaky kolem 400 kPa). U vodní rakety se vzduch do rakety může přivádět z větší vzdálenosti, navíc je hermeticky uzavřená ještě před začátkem tlakování. Je to tedy mnohem bezpečnější, pohodlnější, jednodušší a šetrnější k přírodě (kyselina octová škodí zeleni – bílé kruhy v trávnicích). Octová raketa je dobrá pro rychlou ukázkou letu rakety a vysvětlení chemické reakce, ale pro samotné praktické účely je výhodnější využít právě raketu vodní, kterou jsem si zvolil pro tuto práci.

## 1.2 Vylepšení

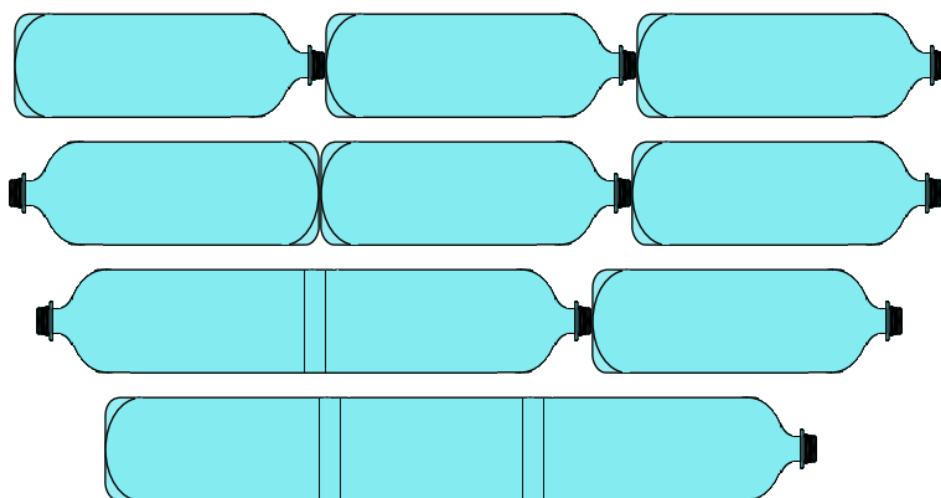
Nejjednodušší forma vodní rakety, jak byla popsána na Obrázek č. 1 má trup z jedné PET lahve a nemá v hlavici nic umístěno (měřicí zařízení, padák). Pokud chceme dosahovat větších výšek vzletu, stability, množství získaných dat, či větší bezpečnosti musí být provedena určitá vylepšení konstrukce.

### 1.2.1 Trup rakety

V základu se obecně předpokládá objem trupu rakety (objem jedné PET lahve) 2 l, či 2,25 l. To je často největší objem, ve kterém lze zakoupit šumivé nápoje. Lahve šumivých nápojů jsou konstrukčně nejpevnější varianty PET lahví, které jsou vyráběny, aby vydržely vysoké vnitřní tlaky (do 700 kPa i více v závislosti na podmínkách a výrobcí). Při vyšším tlaku dochází k permanentní změně tvaru lahve, až k následnému roztržení. Optimální by bylo zakoupit například 6 litrovou PET lahev úzkého podlouhlého profilu, bez tvarových úprav. Takovou, ale na trhu nenalezneme. Je tedy potřeba spojit více lahví, tak aby vytvořily celkově větší objem. V rámci práce se počítá se spojením tří PET lahví o jednotlivém objemu 2,25 l.

Spojování tří PET lahví může být provedeno vícero způsoby viz Obrázek č. 2, kdy každá z metod poskytuje různé výhody, či nevýhody a představuje různé překážky při tvorbě.

V praktické části se zaměřím pouze na způsob, jakým jsem se rozhodl spojování provést, společně s problémy, postřehy a myšlenkami pro případné další vylepšení.



Obrázek č. 2 – Různé způsoby spojování PET lahví (odshora: jenom Robinson coupling, jenom Robinson coupling druhý způsob, splicing + Robinson coupling, jenom splicing)

Zvětšování objemu trupu rakety má kromě možnosti vyletět výše navíc další sounáležitosti. Zvětšujeme zároveň výšku, hmotnost a plochu trupu rakety, a tudíž se mění i místo těžiště a působíště tlaku. To má obecně za následek lepší stabilitu (popsáno v [1]).

Větší hmotnost má však za následek menší zrychlení (tah je stejný, ale tíhová síla je větší), to je kompenzováno možností mít větší počáteční množství kapaliny a stlačeného plynu, což má za následek delší časový úsek, po který je raketa urychlována (větší impuls síly tahu rakety).

### 1.2.2 Hlavice

Zásadní vylepšení, které jsou podstatné pro tuto práci, jsou právě v hlavici. Tato část rakety má v praktickém hledisku tvorby vodních raket 4 významy:

1. Aerodynamický
2. Návratový
3. Nákladový
4. Stabilizační

Aerodynamický význam vychází z potřeby snížení odporu vzduchu a zefektivnit letové schopnosti rakety. Tento význam jsem dále rozebíral v mé bakalářské práci v sekci 2.1.4.

Návratový význam byl v bakalářské práci zmíněn, ale neproveden. To také vedlo k tomu, že raketa byla po dokončení letu velmi poškozena a zároveň její dopad nebyl moc bezpečný. Proto jsem se v tomto ohledu rozhodl vytvořit systém, který by měl zaručit vypuštění padáku v momentě přetočení rakety (ideálně, když raketa dosáhne nejvyšší

vzletové výšky). Takto by se mělo nejenom zaručit bezpečí osob, ale i samotné rakety a jejího nákladu.

Rakety by samy o sobě neměly moc význam, kdyby nebyly schopné přenášet hmotný náklad (kromě samotné rakety). Právě ten se nachází v hlavicích a podle potřeb daných raket se jejich obsah různí (živá posádka, družice, výbušný materiál, senzory a počítače, ...). Pro zájmy této práce se bavíme o Arduinu společně se senzory, médiiem na ukládání dat, servomotorem a bateriemi. Právě tento význam z hlavice na raketě činí její zásadní podstatu i její "vysokou" cenu.

Jak již bylo zmíněno, tak delší profil rakety může tvořit lepší stabilizaci. Je tedy i v případě hlavice tento význam částečně relevantní. Zvláště pak pokud se například rozhodneme pro raketu pouze z jedné PET lahve.

## 2 Arduino

Arduino je jednoduchá a cenově dostupná počítačová deska s mikrokontrolerem (malý počítač), která dokáže získávat vstupní data z různých senzorů a snímačů (např. akcelerometr, gyroskop, barometr, ...) a zpracovat získané údaje na výstupu (např. otočit servomotorem, zapsat data na SD kartu, ...). Skrze roky, co již existuje vzniklo nespočet projektů, jímž je právě Arduino mozkiem. [5][6]

Počátky vzniku Arduina začínají rokem 2005 v Itálii. Skupina nadšenců do elektroniky z Institutu Interakčního Designu ve městě Ivrea se rozhodla vytvořit levnou a jednoduchou variantu tehdy ještě obecně drahých vývojových setů. [7]

Hlavním důvodem celosvětové úspěchu však nebyla možnost koupě samotné desky, ale právě takzvaného open source [7], který umožňuje volné šíření softwarových a hardwarových prostředků, uživatelům. To dalo za vznik jednak velkému množství knihoven, které rozšiřují možnosti softwarové a takzvaných klonů, které díky volné dostupnosti konstrukčních plánů pod licencí Creative Commons umožňuje vytvářet upravené kopie. [5]

Díky tomu může člověk koupit náhražky, které se zdánlivě jeví jako originál. Ty mohou lákat ještě nižší cenou, přičemž mohou být vadné, či znatelně méně výkonné. Je proto potřeba si dávat pozor při koupi a objednávat pouze od prověřených dodavatelů, či oficiálního výrobce.





Obrázek č. 3 – Logo Arduino

Základním stavebním prvkem Arduinu je 8bitový mikrokontrolér ATmega vyráběný firmou Atmel. [8] Jeho často používanou variantou je model ATmega328P, jehož specifikace jsou:

#### **ATmega328P**

<b>Architektura:</b>	AVR (Harvardská architektura) [8]
<b>Pracovní napětí:</b>	5 V
<b>Frekvence:</b>	16 MHz
<b>Paměť:</b>	32 KB (2 KB zabere Bootloader)
<b>SRAM:</b>	2 KB
<b>EEPROM:</b>	1 KB
<b>Proudový odběr:</b>	20-30 mA
<b>Max. proud pro I/O na 1 pin:</b>	40 mA

Tabulka č. 1 - specifikace ATmega328P [9]

Jedná se o model, který využívá deska Arduino Nano, jenž je využita v této diplomové práci. Je zároveň součástí současně nejčastěji používané desky Arduino UNO. [7]

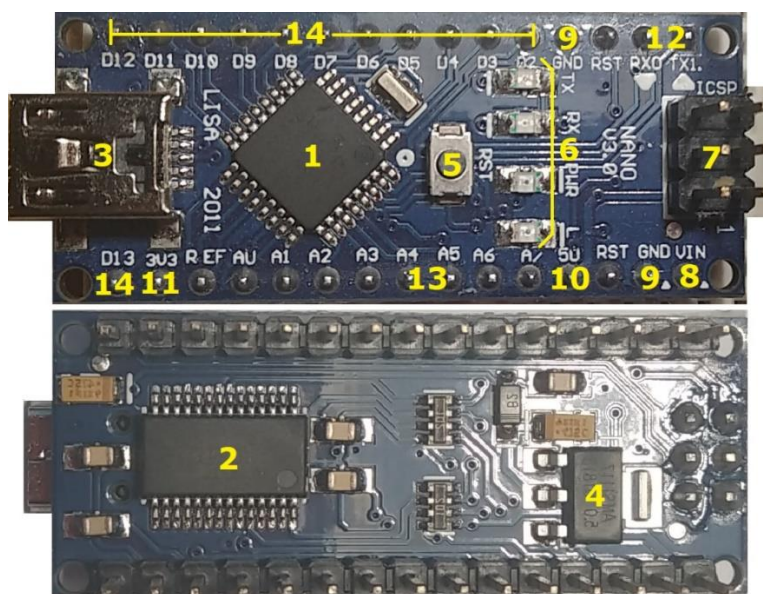
Další důležitou součástí jsou vstupně výstupní porty (I/O), díky kterým je možné posílat data ze vstupních zařízení na zpracování nebo zadávat příkazy na provedení do výstupních zařízení. Tím vzniká nepřeberné množství kombinací, ke kterému lze Arduino využít.

Aby bylo možné provádět komunikaci s počítačem, mají desky na sobě často ještě převodník [7] „USB-to-RS232“ [8], díky kterému může probíhat komunikace přes USB konektor mezi počítačem a deskou Arduina. Ten se u všech modelů objevovat nemusí, což může být způsobeno snahou uspořít místo, nebo je již zabudovaný v čipech. [7]

## 2.1 Arduino Nano

Pro účely projektů, kde je důležitá, či výhodná kompaktnost se hodí využít právě desku Arduino Nano. Vzhledem k rozměrům hlavice rakety, do které bude deska umístěna a požadavkům na vstupně výstupní zařízení jsem zvolil právě tento typ.

V příloze č. 1 je přiložen obrázek s detailním popisem pinů, kterými deska disponuje. Společně s omezeními na maximální velikosti proudů. Zároveň jsem vytvořil vlastní Obrázek č. 4, na kterém krátce popíšu většinu zásadních částí desky Arduino Nano.



Obrázek č. 4 – Arduino Nano v3.0 (vrchní a spodní pohled)

Popis Obrázek č. 4:

- 1 Mikrokontrolér ATmega328P – mozek desky
- 2 Převodník FT232R – pomocí něj může probíhat komunikace s PC přes USB
- 3 USB mini typ B (5 pinů)
- 4 Stabilizátor napětí 5 V – zásadní při připojení externího napětí přes VIN pin
- 5 RESET – jeho zmáčknutí se nahraný program spustí znovu od začátku
- 6 Indikační LED – od spodu nahoru L, PWR, RX, TX
  - L – blikat může přes programový příkaz pro testování, ale zároveň je spjata s pinem 13 [7]
  - PWR – svítí při probíhající napájení desky
  - RX – rozsvítí se, když dojde přijímání signálů do desky
  - TX – rozsvítí se, když dojde k odeslání signálů do výstupních zařízení
- 7 ICSP piny se zdají být důležité, ale často se nevyužívají. Jsou podstatné, když chceme předávat data z jednoho Arduina do jiného např. přehrát bootloader. V této práci využité nejsou.
- 8 VIN pin, který slouží pro připojení externího napájení v doporučeném rozsahu 7-12 V.
- 9 GND (ground – zem) pin, který slouží jako uzemnění. Doporučuji vždy před přivedením napětí si kontrolovat správnost zapojení, hrozí zničení desky a vstupně výstupních zařízení!
- 10 5 V pin, kterým rozvádíme napájení do vstupně výstupních zařízení. Navíc na něj můžeme napojit externí zdroj, ale musí se jednat o spolehlivý regulovaný zdroj o napětí 5 V!

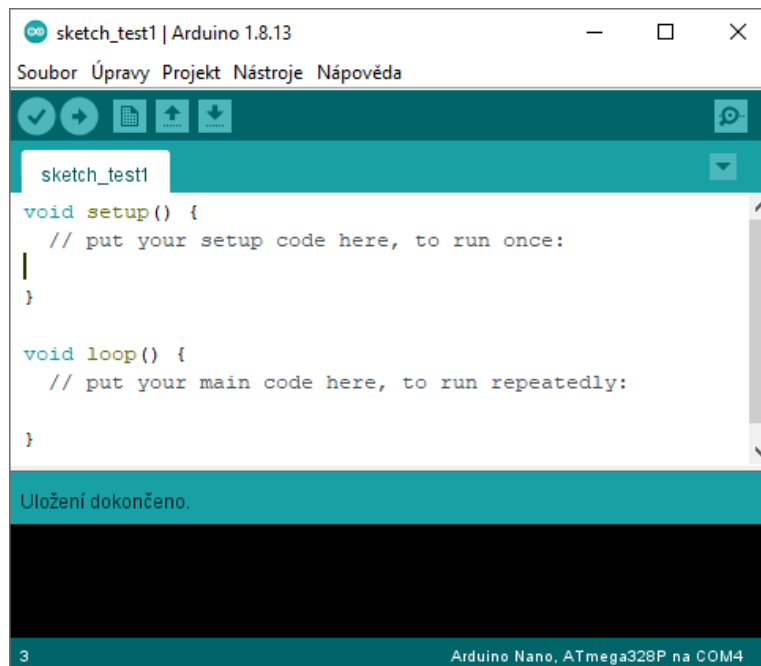
- 11 3V3 pin, který dodává regulované napětí 3,3-3,4 V na vstupně výstupní zařízení, které operují na daném napětí.
- 12 RX0 a TX1 pin – nejčastěji používaný pro využití Bluetooth modulu, pro předávání dat.
- 13 A4 (SDA) a A5 (SCL) piny – pomocí nich dokáží fungovat senzory připojené k Arduinu Nano
  - SDA (Serial Data) – datová linka
  - SCL (Serial Clock) – linka hodinového signálu
- 14 Piny D2-D13 – piny používané pro práci s moduly, které fungují na digitální logice signálů (0 V, 5 V). Některé z nich dokáží využít pulzně šířkovou modulaci (PWM). [10]

## 2.2 Vývojové prostředí (IDE)

Vývojové prostředí slouží programátorům k usnadnění a zpříjemnění práce při vytváření algoritmů a programů. Často nabízí možnosti nápovědy, kontroly, formátování a mnoho dalšího s čím se při tvorbě často uživatel setká. Můžeme jej nalézt pod zkratkou IDE neboli *Integrated Development Environment*.

IDE pro práci s Arduinem je vytvořené v jazyce Java. [7] Kdy samotné grafické prostředí vyšlo z prostředí Wiring [11], což je „podobný projekt jako Arduino“ [8] a Processing [12], což je „prostředí pro výuku programování“ [8].

Programovací jazyk tohoto prostředí je C nebo C++, kdy se klasicky využívá knihovna jazyka C++ zvaná Wiring. Jelikož je za podstatně dlouhou dobu používání značně rozšířená, dá se o ní mluvit, jako o samostatném programovacím jazyku. [7]



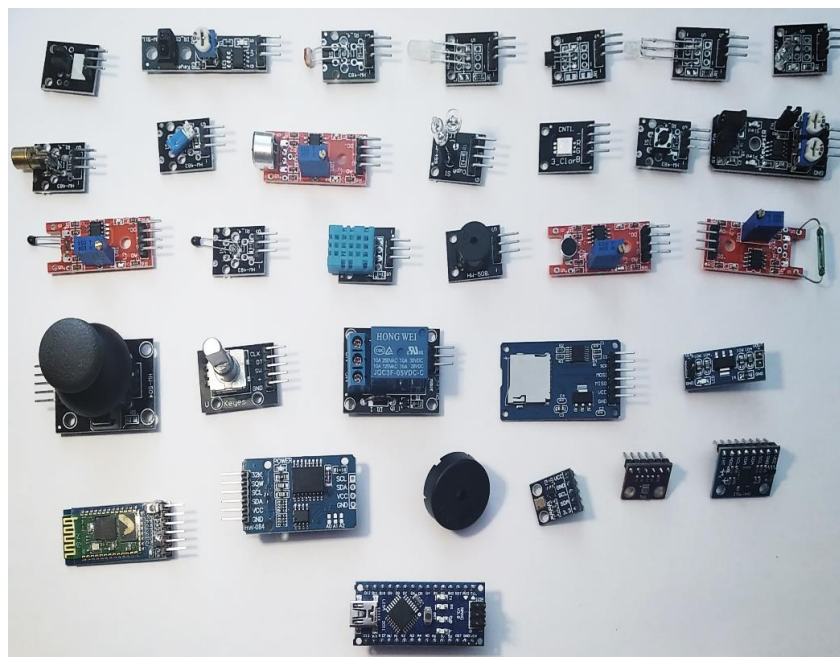
Obrázek č. 5 – IDE Arduino

IDE Arduina nabízí přívětivý a jednoduchý design, který nováčka nezaskočí spoustou možností, což je právě pro výuku v něm dost výhodné. Postupné aktualizace dodávají funkce, které zkušený uživatel rád ocení, jako je například dopsání proměnných přes zkratku Ctrl + Enter atp.

Základní princip, na kterém probíhá tvorba kódu, je založen na funkcích *void setup()* a *void loop()*. V první z nich se provádí úvodní nastavení a provedení všech potřebných příkazů, které mají často za účel spuštění přídatných modulů a kontrola jejich funkčnosti. V druhé dochází ke smyčce, která se opakuje, dokud je Arduino napájeno. Před funkcí *setup()* se importují dodatečné knihovny a deklarují proměnné. [10]

## 2.3 Přídavné moduly

Arduino Nano je samo o sobě dost “holé” zařízení. Pro získání možnosti využití jeho dispozic je zásadní připojit externí moduly, díky kterým mohou vznikat účelné projekty. Je také důležité si ze začátku uvědomit, že práce s Arduinem je práce s elektronikou, a proto je dobré mít základní elektrotechnické znalosti jako je například Ohmův zákon, způsob zapojování ampérmetru, měření napětí, měření odporů (čtení odporů) a další. Samozřejmě i bez nich se dá ledasco zajímavého programovat a tvořit, avšak dojdeme do bodu, kdy budou tyto znalosti nutné. Zvláště pokud nebude něco fungovat, jak má.



Obrázek č. 6 – Ukázka přídavných (externích) modulů, dole Arduino Nano 3.0

Moduly jsou často jednoúčelné, proto je jejich velikost velmi malá. Pro jejich použití je vhodné nastudovat dokumentaci, která lze jednoduše najít na internetu. Díky open-source platformě lze získat zadarmo již vytvořené knihovny, které v sobě mají připravené funkce od jiných uživatelů a nám odpadne nutnost komplikované práce s registry.

Na trhu lze najít moduly i jiných značek (Sparkfun, Adafruit), jejich využití je oficiálně podporováno. [13] „Jejich prodej je často zprostředkován i z oficiálních stránek projektu Arduino.“ [13]

## 2.4 Shieldy

Jiným způsobem, jak rozšířit možnosti desek Arduina je připojení takzvaných shieldů [šildů]. Může se jednat o rozšíření jako je: Wifi, ethernet, motor, či úprava pro jiný druh připojování periférií. [7]



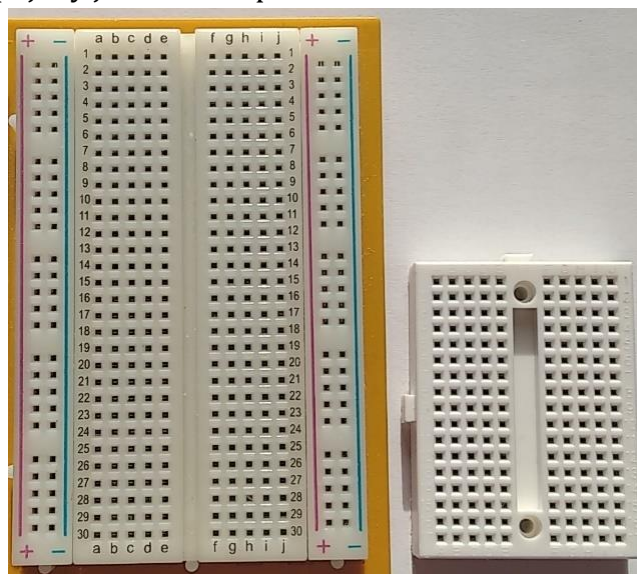
Obrázek č. 7 – Ukázka shieldu pro Arduino Nano

Je však třeba si dávat pozor, aby zakoupený shield byl kompatibilní s typem desky Arduina. [7]

### 2.4.1 Nepájivé pole

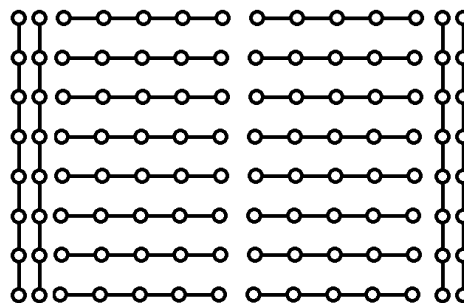
„Nepájivé pole (věc, co vypadá jako bílá čokoláda se spoustou dírek, do nichž můžete elektronické prvky strkat a různě propojovat, aniž byste se museli obtěžovat pájením)“ [14]. Tak by se dalo nepájivé pole jednoduše popsat.

Zásadní věcí při vytváření prototypu elektrického obvodu je možnost jeho jednoduché, přehledné a rychlé možnosti propojení, které bude snadno upravitelné. K tomu slouží právě nepájivé pole. Pomocí mnoha otvorů, které jsou navzájem určitým způsobem z výroby vodivě propojeny je následné práce s vodiči velmi snadná.



Obrázek č. 8 – Ukázka nepájivých polí

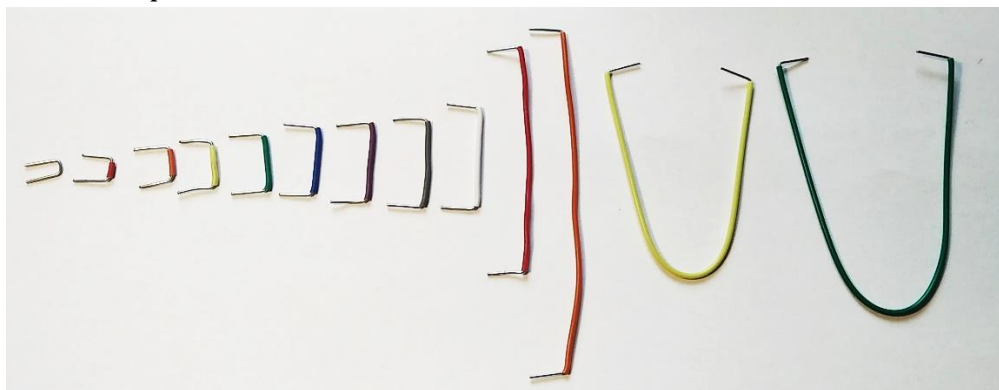
Z Obrázek č. 8 lze vidět, že jsou nepájivá pole různých rozměrů, počtu otvorů a způsobů propojení. Vlevo je verze, která má postranní vodivé pásy pro rozvedení napájení a uzemnění, což je při práci s větším množstvím součástek velmi užitečné. Obě nepájivé pole mají levou a pravou soustavu horizontálně vodivých řad, tak jak lze vidět na Obrázek č. 9.



Obrázek č. 9 – Schéma nepájivého pole

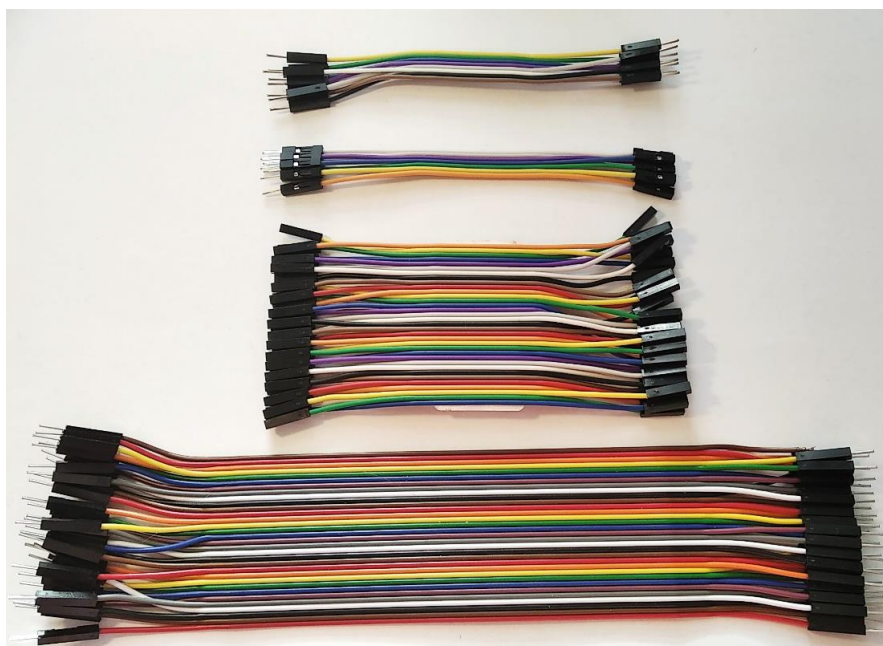
## 2.4.2 Vodiče

K propojování elektronických komponentů je nutné použití vodičů, které jsou napojené přímo z vývodních pinů desky Arduina na ostatní komponenty, či nepřímo přes nepájivé pole. Vodiče bývají součástí vývojových kitů Arduina, či se dají poměrně levně ve větších množstvích zakoupit.



Obrázek č. 10 – Vodiče neosazené

Jak lze vidět na Obrázek č. 10 a Obrázek č. 11 jsou vodiče vytvářené v různých velikostech a barevných provedeních. Zvolení vhodných barev (barvy izolační vrstvy) a délek pomáhá k udržení přehlednosti, navíc je vhodné si stanovit určité barvy pro přívod napětí a uzemnění. Typicky to je červená pro napětí a černá pro uzemnění. Nezáleží však na barvách, spíše na tom barevná pravidla dodržovat, což nám může ušetřit spoustu problémů do budoucna. Jak si lze také všimnout z obrázků, je při nedostatku černých a červených vodičů, což se stává často, dobré stanovit barvy přidružené. Rozšířit tak červený kabel například hnědým a černý modrým. To už samozřejmě závisí na každém zvlášť, jak si barvy rozvrhne.



Obrázek č. 11 – Vodiče osazené (typ od shora: M-M, M-F, F-F, M-M)

Vodiče lze zakoupit neosazené, či osazené. Osazené vodiče se dělí na 3 typy. M-M, M-F a F-F (M = Male => samec, F = Female => samice). Právě podle toho se liší koncové osazení. Před započítím práce s Arduinem doporučuji zakoupit všechny typy, které lze zároveň koupit v několika délkových variantách.

## 2.5 Senzory

Zásadním prvkem této práce je využití senzorů, nebo též snímačů. Právě pomocí nich je například v dnešním technologicky rozvinutém průmyslu možné předávat informace procesorům. Dalo by se říct, „že počítače by byly zcela bezmocné bez snímačů“ [15]. Pomocí nich jsme schopni získávat stav prostředí, se kterým jsou v přímém styku a následně tak zjistit hodnoty tlaku, teploty, zrychlení a další. [15] Kromě průmyslu jsou nedílnou součástí mnoha projektů při práci s Arduinem, kde se s nimi lze poměrně snadno naučit pracovat.

Senzor je zdrojem fyzikálních, chemických, či biologických hodnot, které se nejčastěji převádí na různé druhy elektrických signálů, jako je napětí, proud, kapacita atp. Část, která je citlivá právě na dané hodnoty se nazývá čidlo a je právě jedním z rozhodujících parametrů, které mají dopad na kvalitu senzoru. Signál se převádí analogově, kde například určité hodnotě tlaku je přiřazena určitá hodnota proudu, nebo binárně (dvoustavově), kde například může senzor fungovat jako spínač, který kontroluje určitou vzdálenost, která když je překonána vyše napěťový výstupní signál. [15]

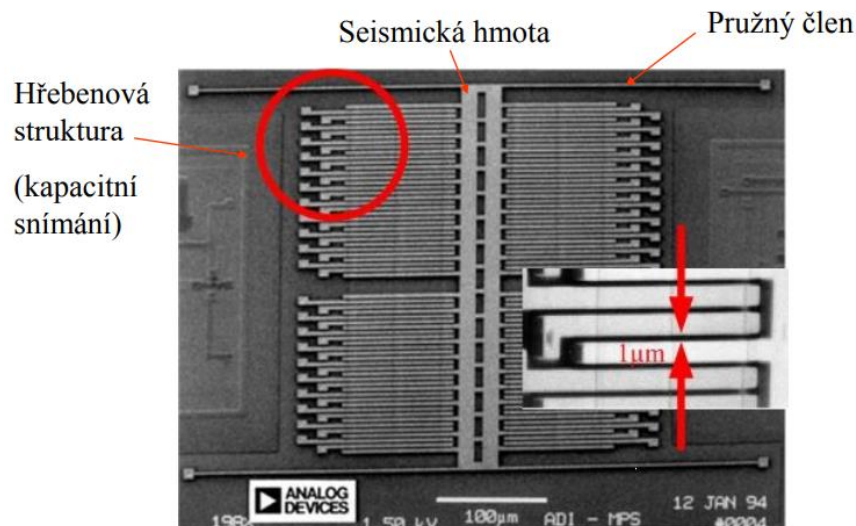
V rámci tvorby s Arduinem je na trhu obrovské množství senzorů, které jdou pořídit za velmi přívětivé ceny. Co do samotných kvalit se nemusí úplně rovnat průmyslové kvalitě, avšak tam se zase cena a nepřívětivé programování může projevit, jako překážka pro začínající a hobby uživatele.

### 2.5.1 Akcelerometr

Jak název napovídá jedná se o zařízení, které měří akceleraci neboli též zrychlení. Přesněji řečeno měří síly působící na senzor, ve směrových osách x, y a z. Ty jsou na sebe navzájem kolmé a dávají nám tak možnost úhlové orientace a pohybových změn senzoru. Data jsou z analogového signálu převedeny na digitální a odeslány na výstupu. [16][17]

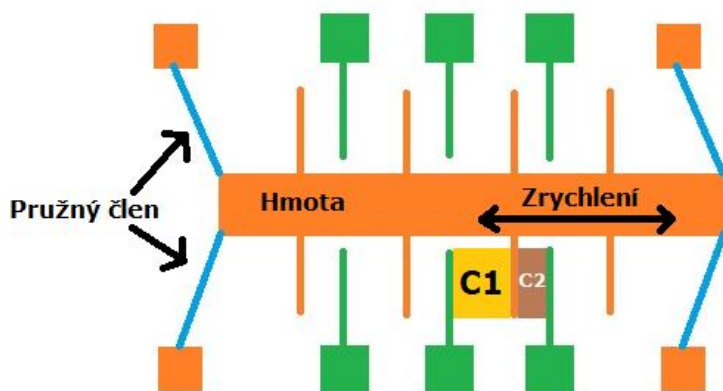
Technologicky jde o využití MEMS (Micro Electro Mechanical Systems). Jedná se o velmi malé systémy, nebo zařízení, které se skládají z komponentů v řádu velikosti desetin až tisícín milimetru. Komponenty jsou vytvořeny ze silikonu, polymeru, kovu nebo keramiky. Zároveň jsou většinou skombinovány s vlastní řídicí jednotkou. [18]





Obrázek č. 12 – MEMS – akcelerometr

K měření zrychlení dochází měřením změny kapacity. Na Obrázek č. 12 vidíme seismickou hmotu, která je připevněna k pružnému členu. Ten funguje, jak název napovídá, jako pružina. Pružný člen je pohyblivý pouze v jednom směru, takže když dojde právě v něm k udělení silových účinků, hmota se začne pohybovat (oscilovat). Hmota má hřebenové výběžky, které jsou umístěny mezi napevno osazenými deskami kondenzátorů, kde lze následně měřit změnu v poměru kapacit (viz Obrázek č. 13) a přes výpočet, tak zjistit zrychlení v ose pohybu hmoty. [18] Jsou tak vytvářené akcelerometry různých stupňů volnosti, kde při nutnosti 3D orientace musíme takto měřit právě 3 navzájem kolmé osy.



Obrázek č. 13 – MEMS – akcelerometr – nákres

Při práci s Arduinem a akcelerometry se lze setkat s celou řadou modulů, které na sobě mají jejich různě kvalitní provedení.

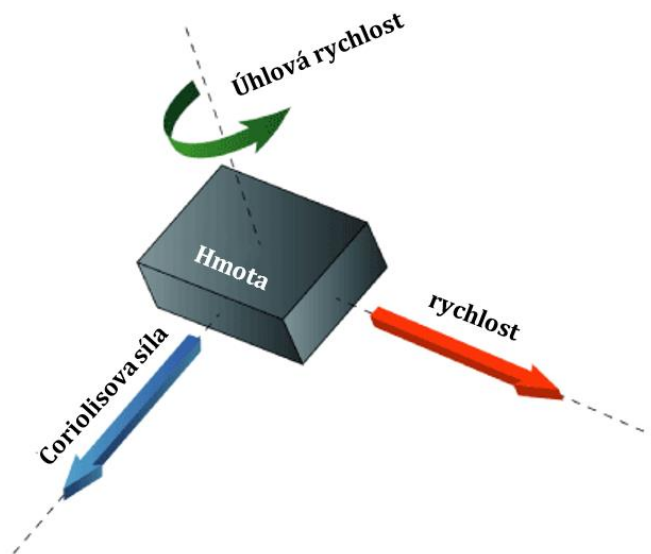
## 2.5.2 Gyroskop

Často se na modulu akcelerometru též vyskytuje gyroskop a v některých případech i magnetometr. Společně tak vzniká modul, který se označuje jako IMU (Inertial Measurement Unit), neboli jednotka, která získává a odevzdává data nejenom

o působících silách, ale i změnách úhlů ve směrových osách a horizontální orientaci za pomoci magnetometru, který funguje jako kompas. Následně pak lze za pomoci matematických operací kombinovat tyto data a získat možnost popsat kompletní letový stav objektů. [16][19]

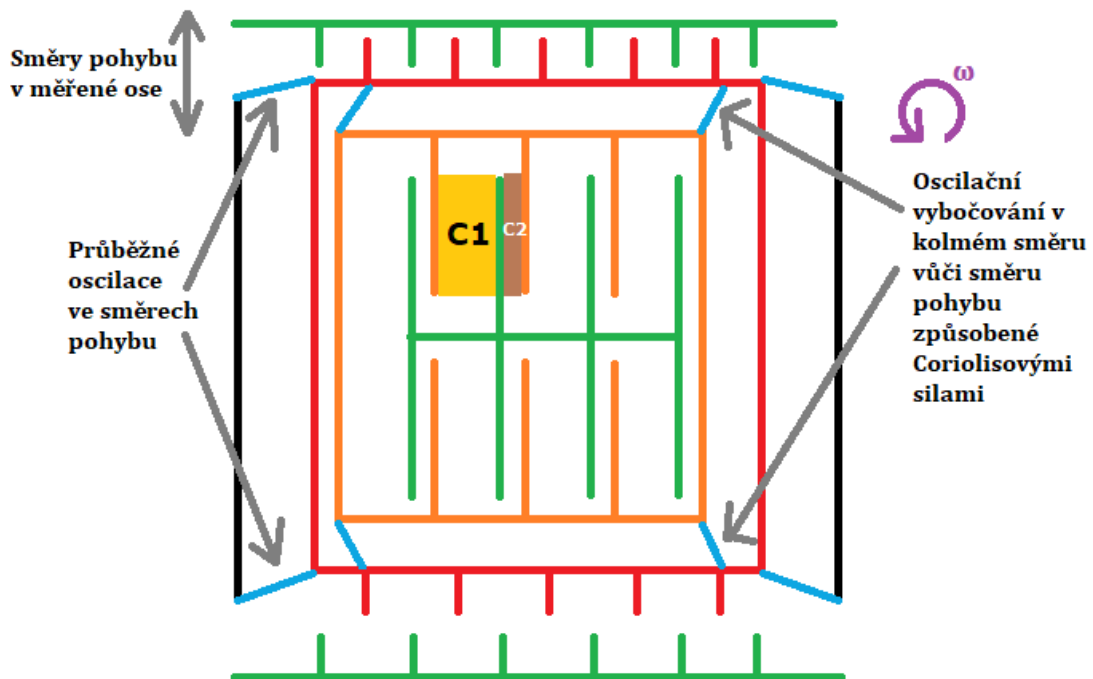
Důvod, proč je obecně potřeba použít právě všech těchto senzorů vychází z problémů dlouhodobého měření, kdy dochází k akumulaci chyb v rámci různých šumů. Následně se pak používají různé filtračních metody jako například Kálmánův filtr, který je má potlačovat. [19] To je poměrně za hranicemi komplexnosti, kterou se snažím touto prací docílit, avšak věřím, že mi gyroskopická data poskytnout zajímavá zjištění.

Gyroskop v MEMS provedení funguje na principu Coriolisovy síly. Ta říká, že pokud se hmota pohybuje v určitém směru s určitou rychlostí a ocitá se v rotující vztažné soustavě vznikají zdánlivé silové účinky působící právě na danou hmotu, jak lze vidět na Obrázek č. 14. Takže podobně jako v případě akcelerometru dojde k vychýlení hmoty, což způsobí změnu naměřených kapacit, čemuž odpovídá určitá úhlová rychlost. [18] Úhlová rychlost se získá v radiánech za sekundu, což lze jednoduše převést na úhly a ve výsledku například zjistit kolikrát a jak rychle se raketa otáčela ve svých osách, kde ta nejzásadnější bude osa z.



Obrázek č. 14 – Coriolisův efekt

Stejně jako v případě akcelerometru jsem vytvořil náčrt pro gyroskop v MEMS provedení viz Obrázek č. 15. Samotné oscilace ve dvou směrech jsou již trochu složitější na představu, proto je dobré zapátrat po internetu, kde jsou i případné animace, na kterých je to lépe vidět.



Obrázek č. 15 – MEMS – gyroskop – nákres

### 2.5.3 Barometr

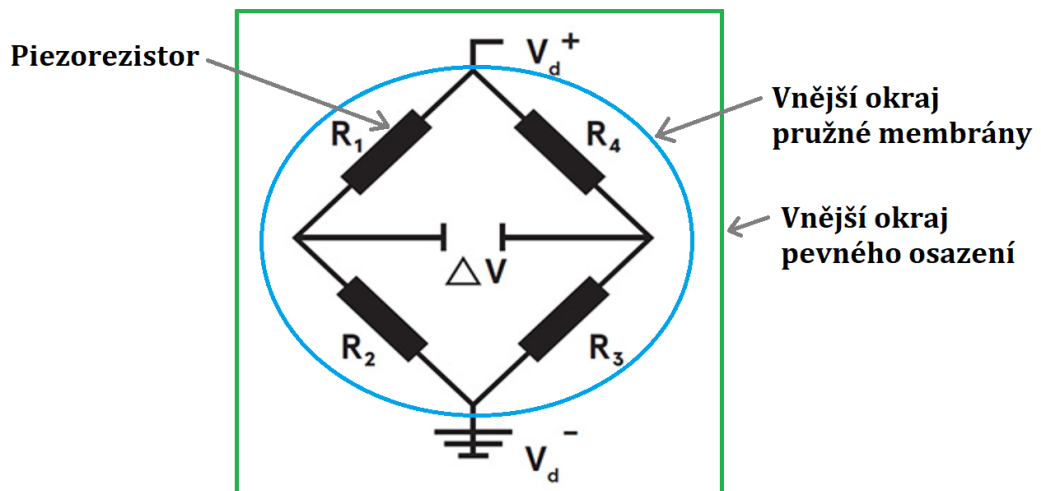
Barometr je zařízení, které primárně slouží k měření atmosférického tlaku, jenž se obecně zmenšuje s rostoucí nadmořskou výškou v důsledku úbytku tíhy atmosféry. Díky tomu je pak možné pomocí vzorce vypočítat nadmořskou výšku, která se v průběhu letu mění a stanovit časovou změnu výšky. [20]

Pro měření vzletové výšky vodních raket se může použít hned několik metod. Důvod použití právě MEMS barometru v této práci je výhodné právě pro jeho jednoduché využití a velkou přesnost. Další možnosti měření vzletové výšky lze najít zde [21].

V rámci tvorby MEMS barometrů se většina tvoří na principu piezoelektrickém a již popsaném kapacitním. V obou případech je vytvořena pružná membrána, která se působením vnějšího tlaku deformuje. Zaměřím se zde na nepopsanou piezoelektrickou metodu. [22]

V rámci piezoelektrického jevu se uvažuje o přímé a nepřímé variantě, kdy u práce se senzory se využívá právě přímá. Ta v rámci namáhání dielektrika (běžně křemen, či křišťál) tlakem, tahem, ohybem, či krutem dává za vznik elektrických nábojů, které se dají následně zaregistrovat a zpracovat jako změna napětí. [23]

Vodivé snímací prvky, které slouží jako proměnné odpory, fungující právě na piezoelektrickém jevu, ty jsou pak osazené přímo na membráně. Zapojené jsou do Wheatstoneova můstku, což je zapojení (viz Obrázek č. 16) uzpůsobené pro přesné měření změn odporů, či v tomto případě napětí. [22][24]



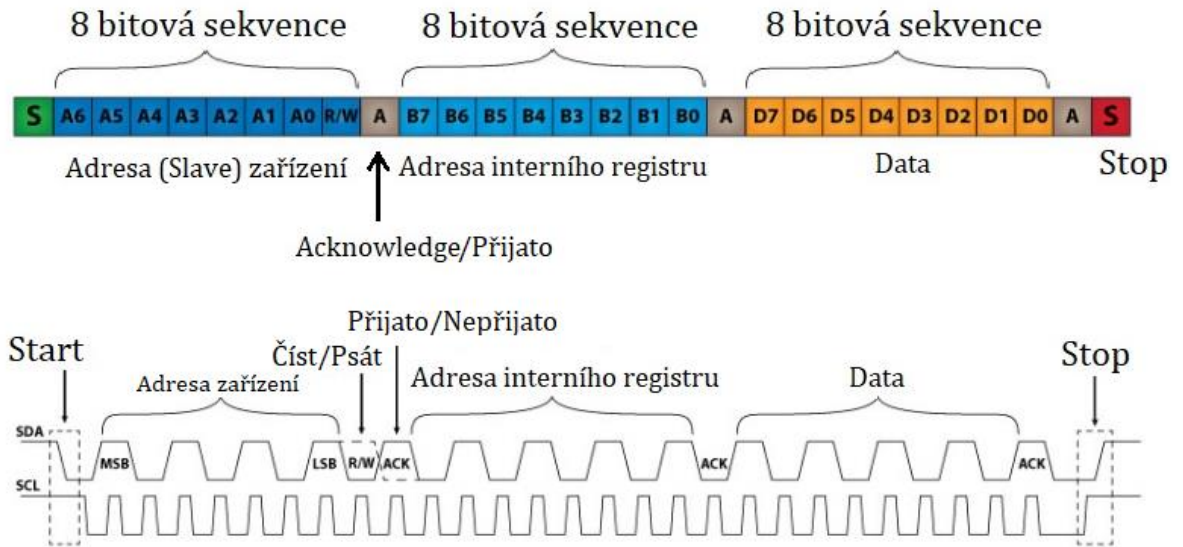
Obrázek č. 16 – MEMS – barometr – schéma

Tento typ senzorů, je hojně využíván v automobilovém a lékařském průmyslu, či u domácích spotřebičů. [22] Při použití pro Arduino se dá opět setkat s různými moduly obsahující různě kvalitní barometry. Parametry a odchylky se dají jednoduše najít v dokumentacích, avšak pro každý vyrobený kus se budou pravděpodobně lišit a je vhodné si případně odchylky změřit.

#### 2.5.4 I<sup>2</sup>C

Přijímání dat ze senzorů u Arduino Nano probíhá přes zmíněnou linku SDA na pinu A4. Při přijímání dat z více senzorů, však nastane problém, protože Arduino samo od sebe nerozezná, která data patří určitému senzoru. K tomu se využívá I<sup>2</sup>C protokol. Každé zařízení (senzor) má vlastní přednastavené ID, kde se následně za pomoci 7bitové adresace dá rozpoznat až 128 (2<sup>7</sup>) zařízení. [25] To je z praktického hlediska pro práci s Arduinem nadměrně vysoké množství, se kterým si uživatel dobře vystačí.

Kromě SDA linky se využívá i linka SCL, která je na pinu A5 a její funkcí je synchronizace dat. Na této lince se vysílá hodinový signál. Celkově je I<sup>2</sup>C protokol fungující na modelu Master/Slave (Pán/Otrok), kdy je vždy alespoň jeden řídicí člen Master (může být i více) a vícero členů "sloužících" Slave. Arduino Nano tak funguje jako Master a senzory jako Slave. Hodinový signál je generovaný Masterem. [25]



Obrázek č. 17 – I<sup>2</sup>C protokol

Dle Obrázek č. 17 jsou data převáděná v sekvencích po 8 bitech. Po startovní podmínce je prvních 8 bitů označení ID daného Slave zařízení (7 bitů), se kterým se bude pracovat a poslední rozhodne o tom, jestli Master provede čtení (logická 1) ze Slave, či zápis (logická 0) do Slave. Lze si všimnout, že adresace začíná MSB (nejvýznamnějším bitem) a končí LSB (Nejméně významným bitem). [25]

Následuje bit ne/přijato, kterým dává Slave na vědomí, že předchozí data ne/přijmul a je ne/schopen další aktivity na lince SDA. V případě, že Slave není schopen další aktivity, tak Master rozhodne, co se bude dít dál. Po úspěšném přijetí následuje adresace interního registru. To je lokace v paměti Slave zařízení, která obsahuje určitá data a informace. Například se může jednat o registr uvnitř paměti akcelerometru obsahující zrychlení v ose x, kterou my budeme chtít číst. Dané interní registry lze nalézt v dokumentaci registrů daného zařízení. Následuje opět bit ne/přijato a samotná datová operace, která závisí na tom, jestli jsme chtěli data číst, nebo zapsat. Při čtení dojde k uložení dat z interního registru a při zápisu se data v registru přepíší, což je například potřeba, když chceme operovat s registry obsahující informace o nastavení zařízení Slave. Opět je nakonec bit ne/přijato a ukončovací podmínka uvolňující linku SDA. [25]

Z praktického hlediska se při práci s Arduinem a senzory potkáme pouze s nutností znát adresu Slave zařízení, kterou lze zjistit stažitelnými knihovnami například *i2cdetect*. Často pak knihovny vytvořené pro moduly mají interní registry a operace s nimi přepsané do funkcí, které se pak dají snadno volat a využívat, bez nutnosti znalosti adres interních registrů.

## 2.6 Napájení

Arduino Nano pracuje na napětí 5 V. Díky tomu je výhodné využití USB napájení, které 5 V standardně poskytuje a může sloužit nejen jako stabilní zdroj napětí, ale zároveň jako způsob, jak přenášet data (pokud je připojeno k PC). V rámci koupě prvního Arduina je dobré zkontrolovat jaký typ USB připojení deska podporuje a pokud není součástí balení rozhodně jej zakoupit, neboť bez něj není možné nahrát programy do paměti Arduina. Kromě napájení přes USB port, jsou dále dvě další možnosti, jak napájet desku Arduino Nano. A to přes VIN pin a 5 V pin.

Přes VIN pin je doporučeno připojovat zdroje o napětí 7-12 V [9], které jsou následně převedeny na 5 V stabilizátorem napětí zabudovaným na desce. Je dobré si uvědomit, že stabilizátor prakticky přeměňuje nadbytečné napětí na teplo, kdy při dodávání vyšších hodnot napětí a nadstandardního množství proudu dochází k velmi značnému zahřívání stabilizátoru, což může vést v lepších případech k sepnutí pojistky a vypnutí přívodu napájení. Tuto metodu je dobré využít, když nemáme externí stabilizátor a odběrové proudy připojených součástek k Arduinu nemají vysoké nároky.

5 V pin slouží klasicky jako metoda výstupního napájení elektronických součástek připojených k Arduinu. Je však možné přes tento pin i samotné Arduino napájet, má to však své rizika. Externí zdroj musí sám stabilně dodávat napětí 5 V. Stabilně se rozumí, že i v případě zátěže by se napětí nemělo zásadně měnit. K tomu abychom stabilizovali zdrojové napětí se využívá již zmíněného stabilizátoru. Technologicky lze využít takzvaných lineárních stabilizátorů, jejichž účinnost je kolem 40-60 %, či step-down/up měničů, které mají účinnost vyšší a to 80-90 %. Kromě rozdílu v účinnosti mají své specifické výhody i nevýhody, ale jejich funkce je stejná. [26]

### 2.6.1 Baterie

Při tvorbě některých projektů je nutné napájet Arduino a ostatní součástky, tak aby s nimi byla zároveň možná volná manipulace. To vyžaduje externí přenosné napájení, což můžou být právě baterie, které „mění chemickou energii na elektrickou“ [20].

Zde přijde na řadu právě chvíle, kdy je dobré znát některé poznatky související s elektřinou. Mnohé problémy a zklamání, které začínající uživatel s Arduinem má, jsou právě z nedostatku pochopení, jak správně napájet projekt pomocí baterií. Je proto dobré znát teorii a zásady s nimi související jako je:

- Zapojování baterií sériově a paralelně a vliv na napětí a kapacitu článků.
- Umět zacházet s multimetrem a vědět, jak měřit napětí a proudy.
- Zjistit si velikosti maximálního vybíjecího proudu baterie a porovnat ho s potřebným odběrem pro všechny součástky.

Co do typů baterií je velké množství, které se na trhu nabízí. Základní rozřazení baterií by mohlo vypadat následovně:

- a) Sekundární články – akumulátory (dobíjecí baterie)
  - Olověné
  - Lithium-iontové
  - Lithium-polymerové
  - Nikl-kadmiové
  - Nikl-metal hydridové
- b) Primární články – galvanické články (jednorázová nedobíjecí baterie)
  - Alkalické
  - Zinko-uhlíkové
  - Lithiové
  - Stříbro-oxidové
  - Zinko-vzdušné
- c) Průmyslové baterie – existují pro napájení těžkých aplikací, jako jsou stroje, železnice a záložní napájení pro datová centra, veřejné služby nebo telekomunikace.
- d) Automobilové baterie – baterie pohánějí naše automobily, motocykly, čluny a další motorová vozidla. [27]

Samozřejmě každý typ a varianta baterií má své výhody a nevýhody. Zaměřím se na ty, které jsem osobně použil, či bych zvažoval použít do budoucna.

V rámci prvních kroků s Arduinem jsem využíval tuto [10] knížku. V ní je zmíněna právě 9 V baterie (též označována jako PP3), kterou lze zakoupit jak v dobíjecí, tak jednorázové verzi. Dobíjecí baterie jsou zpravidla dražší, mají menší kapacitu a v některých případech neposkytují očekávané hodnoty napětí vůči jednorázovým. Mají však vratnou hodnotu právě v možnosti dobíjení. Co do jednorázových se klasicky setkáme s alkalickými články, které jsou cenově přívětivé a ve verzi 9 V se obvykle pohybuje kolem kapacity >500 mAh (záleží na výrobcí). Pokud nám nevadí si připlatit je možné zakoupit lithiové baterie, které mají větší kapacitu, trvanlivost (každá baterie se v průběhu času různou rychlostí i bez zátěže vybíjí) a poskytují i větší odběrové proudy. Právě s odběrovými proudy mohou nastat problémy. Pokud máme v projektu zahrnuté například servomotory, o kterých se ještě zmíním, bude potřeba dodávat pro jejich funkčnost proudy v řádu stovek miliampérů. S tím se právě PP3 9 V baterie nedokáží moc poradit, neboť se maximální odběrové proudy pohybují optimálně do 50 mA, a proto je vhodné v takovém případě vhodné zvolit jinou variantu napájení. [28][29]



Obrázek č. 18 – 9 V alkalická baterie v boxu se zabudovaným vypínačem

Známou formou baterií v dnešní době je baterie označována jako AA. Opět se jde setkat s dobíjecími variantami a jednorázovými. Typicky se jedná o 1,5 V články pro jednorázové a 1,2 V pro nabíjecí typy (existují výjimky). Beru-li čistě jednorázové, tak má cenu přemýšlet pouze o alkalických a lithiových. Alkalické AA články jsou schopné dodávat odběry okolo 500 mA za cenu menší výsledné kapacity. Jejich cena a parametry se dost liší dle výrobce a prodejce, ale jsou obecně znatelně výhodnější než zinkové. Samozřejmě je možnost si připlatit za lithiové, které mají skvělé výsledky při vysokých odběrech. Pro využití při práci s Arduinem se tedy dojde k tomu, že baterie spojíme sériově, čímž zvýšíme celkové napětí. To lze jednoduše udělat zakoupením boxů, nebo shieldů do kterých námi zamýšlený počet baterií vložíme. [26][29][30]



Obrázek č. 19 – 4 lithiové AA baterie v boxu (sériově) s přibudovaným vypínačem

Technologicky pokročilou a hojně využívanou variantou jsou lithium-polymerové akumulátory, které vznikly na bázi lithium-iontových akumulátorů a vylepšují některé jejich nevýhody, což ale neznamená, že by měly být jasně lepší, neboť každý typ má své výhody a nevýhody. V dnešní době se objevují hlavně jako baterie v mobilech, či při sestavování RC modelů, nicméně jejich využití je výhodné i právě pro Arduino. Jsou malé, lehké, mají velkou kapacitu, jsou vytvářené v různých velikostech napětí, dokáží dodávat velké výstupní proudy a jsou dobíjecí. Mají však svá rizika, neboť při špatné manipulaci dokáží být nebezpečné. Je proto potřeba se před jejich využitím dostatečně seznámit se způsobem manipulace a pravidel pro jejich dlouhou životnost. Samozřejmě se v jejich



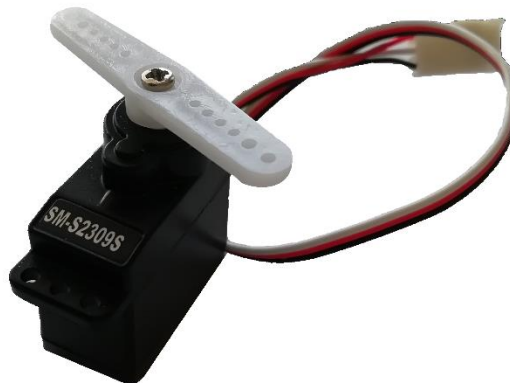
ceně odráží zmíněné výhody, které jsou způsobené náročností výroby, tudíž se nemusí jednat o úplně levnou záležitost.



*Obrázek č. 20 – Li-Pol akumulátor*

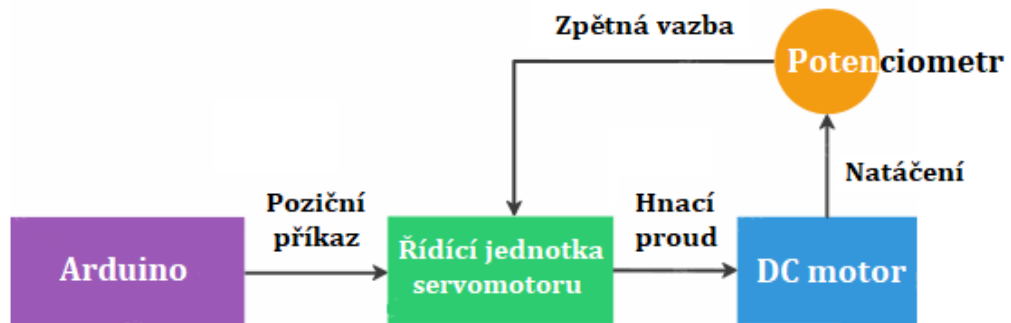
## 2.7 Servomotory

Ať už se jedná o pohyb nohy robota, či zatáčení kol autíčka nebo směrovky letadýlka mají všechny společné to, co jim to umožňuje. Tím jsou servomotory či zkráceně servo. Na rozdíl od samotného elektrického stejnosměrného motoru (DC motor), je možné servomotory precizně ovládat v jejich natočení.



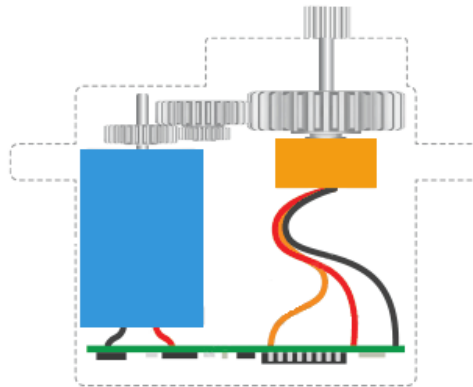
*Obrázek č. 21 – Ukázka servomotoru*

Servomotory jsou uzavřeným řídicím systémem viz Obrázek č. 22, který využívá zpětné vazby k získání požadovaných nastavení. [31]



Obrázek č. 22 – Řídicí systém servomotoru

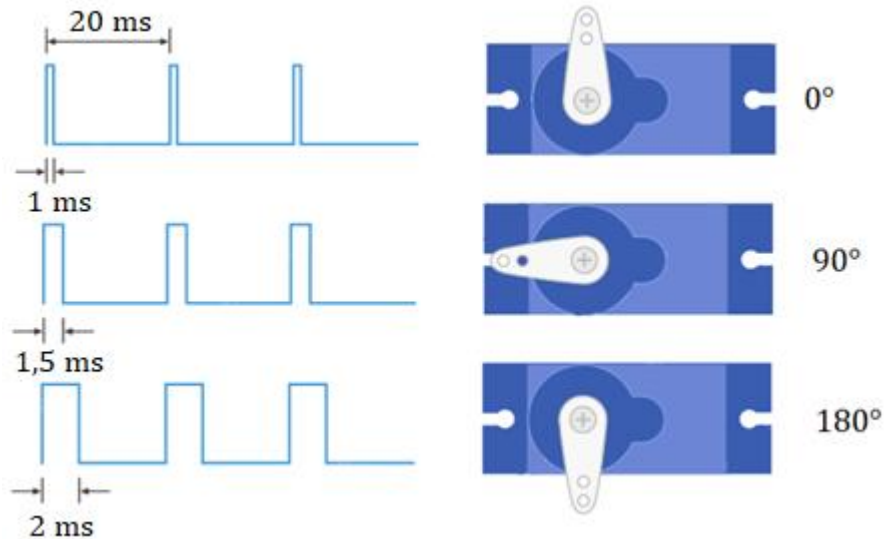
Ve vnitřku servomotoru nalezneme soustavu ozubených kol, které převodují výstup motoru, tak aby bylo docíleno velkého momentu sil a přesnějšího úhlového natočení viz Obrázek č. 23.



Obrázek č. 23 – Průřez servomotoru

Ze servomotoru obecně vychází tři vývodní vodiče (napájení, uzemnění a signál) viz Obrázek č. 21. Jak již bylo zmíněno v předchozí části o napájení v sekci o 9 V baterii, tak servomotory mají poměrně vysoké proudové nároky. Právě kvůli tomu je velmi důležité napájet servomotory z externích zdrojů a nikdy k tomu nevyužívat přívodní 5 V pin od Arduino Nano! Ten není konstruovaný na přívod stovek miliampér, které servomotory při otáčení vyžadují. Uzemnění se přivádí k jednomu hromadnému bodu, tak aby fungovala signálová logika.

Poslední kabel pro signál je přiveden na piny digitálního signálu, kde je následně pomocí pulzně šířkové modulace provedeno nastavení pozice servomotoru. Dle Obrázek č. 24 je využita například frekvence 50 Hz, která odpovídá množství pulzů za sekundu (perioda je tedy 20 ms). Samotná informace o velikosti úhlů je ukryta v šířce pulzu (času, kdy je signál ve stavu logická 1). Pro úhel 145° odpovídá pulzní šířka ~1,806 ms  $((1 / 180) * 145)$ . Samotné frekvence a šířky pulzů se různí dle výrobců, Obrázek č. 24 tedy slouží čistě jako ukázkový příklad. [31]



Obrázek č. 24 – Ukázka pulzně šířkové modulace pro servomotory

## 2.8 Ukládání dat

Samotné Arduino Nano má již zmíněný čip ATmega328P, který má paměť EEPROM o velikosti 1 kB. Šikovný programátor dokáže do této paměti ukládat data, která na ni zůstanou i po vypojení napájení. [32] To však, ale nemusí stačit, když chceme získávat větší množství dat ze senzorů, které pak chceme jednoduše zpracovávat například na stolním počítači. K tomu je výhodné připojení modulu, do kterého lze zapojit SD kartu a mít možnost do ní zapisovat i číst data.

Velkou výhodou je, že IDE Arduina již v základu obsahuje knihovnu, která je právě určena k obsluze modulů pro SD karty. Jediné, na co je však potřeba si dát pozor, je formátování SD karet. Arduino podporuje souborové systémy FAT16 a FAT32. Navíc soubor, se kterým se bude pracovat nesmí mít název delší než osm znaků a samotná přípona má mít znaky tři. Moduly pro SD karty jsou velmi levné, kdy jediné cena samotné SD karty se může být poměrně dražší v závislosti na výrobci na datové kapacitě. [2]



Obrázek č. 25 – Modul pro micro SD kartu

Samozřejmě kromě oficiální knihovny pro práci s moduly SD karet a paměti EEPROM lze najít spoustu knihoven vytvořených komunitou uživatelů.

## 3 Práce s daty

V této části bych se chtěl zaměřit na dílčí problematiku, které by měly následovat před samotným získáním dat a po něm.

### 3.1 Nastavení

K samotnému získání a analýze smysluplných dat vede mnoho jednotlivých kroků. Smysluplných právě proto, že často nestačí pouze spustit měřící přístroj a začít získávat data. V rámci získávání dat ze senzorů, přes jejich interní registry lze poměrně jednoduše získat "surová" data. Úspěšné získání těchto dat je dobrým indikátorem, že jsme navázali spojení se senzorem, což by pro praktické účely mělo sloužit jenom jako začátek.

Kupříkladu u senzoru akcelerometru a gyroskopu lze po nahlédnutí do dokumentace zjistit, že mají nastavitelné rozsahy měření. Pro akcelerometr lze například nastavit možnost měření v rozsahu  $\pm 2$  g ( $1 \text{ g} \sim 9,81 \text{ m}\cdot\text{s}^{-2}$ ),  $\pm 4$  g,  $\pm 8$  g a  $\pm 16$  g. Pro gyroskop zase maximální úhlová frekvence (či změna úhlu za jednotku času). Otázkou tedy je, proč nenastavit senzory vždy na maximální rozsah? Pokud pomineme situace, kdy opravdu chceme měřit děje, které potřebují největší rozsahy, pak odpovědí je citlivost. Data získaná čidlem senzoru jsou nejprve analogová a následně jsou vnitřně zabudovaným převodníkem převedena na digitální. [17] Porovnáme-li tedy citlivost rozsahů  $\pm 2$  g a  $\pm 16$  g pro 16bitový A/D převodník ( $2^{16} = 65\,536$  hodnot), pak vychází hodnoty následovně:

- Rozsah  $\pm 2$  g neboli intervalově  $\langle -2 \text{ g}, 2 \text{ g} \rangle$ 
  - $65\,536 / 4 = 16\,384$  hodnot na jedno g
  - Teoretická přesnost (bez šumu)  $\Rightarrow 1 / 16\,384 \doteq 0,00006 \text{ g}$
- Rozsah  $\pm 16$  g neboli intervalově  $\langle -16 \text{ g}, 16 \text{ g} \rangle$ 
  - $65\,536 / 32 = 2\,048$  hodnot na jedno g
  - Teoretická přesnost (bez šumu)  $\Rightarrow 1 / 2\,048 \doteq 0,005 \text{ g}$

Teoreticky tak dojdeme k 8násobné přesnosti. To však, ale jenom za předpokladu, že hodnoty šumu (chybové odchylky od "přesné" hodnoty) nebudou překračovat přesnosti. Na to bych osobně u "levných" senzorů nespolehal. Ne vždy to, co je v dokumentaci bude platit pro všechny výrobky. Proto je dobré si jednotlivé senzory vždy prozkoušet. To by se tedy týkalo zčásti důležitosti nastavení měřících zařízení, samozřejmě je i více možných optimalizací, jako je například úspornost napájení nebo vzorkovací frekvence. To už je vždy na uživateli, jak moc chce využít možnosti, které senzory nabízí.

## 3.2 Kalibrace

Měřící přístroje i přes všechnu úžasnou technologii, která je tvoří mají své chyby při měření. Kalibrace slouží k tomu, aby tyto chyby byly, co nejvíce zredukovány a naměřená data byla, co nejvíce přesná. [33]

V rámci kalibrace dochází k porovnávání hodnot mezi kalibrovaným zařízením a etalonem neboli kalibrátorem. Etalon je v podstatě mnohem přesnější měřící zařízení, které slouží jako reference pro správnou hodnotu (co nejpřesnější). [33] Problém však může být v tom, mít samotný etalon. Nemusí se jednat o levnou záležitost, a proto je potřeba někdy trochu improvizovat.

Pro barometrické údaje je například možné využít referenční data z nejbližších profesionálních meteorologických středisek například Český hydrometeorologický ústav, či přes aplikace (Google Earth) zjistit bodovou nadmořskou výšku a výpočtem ji porovnat.

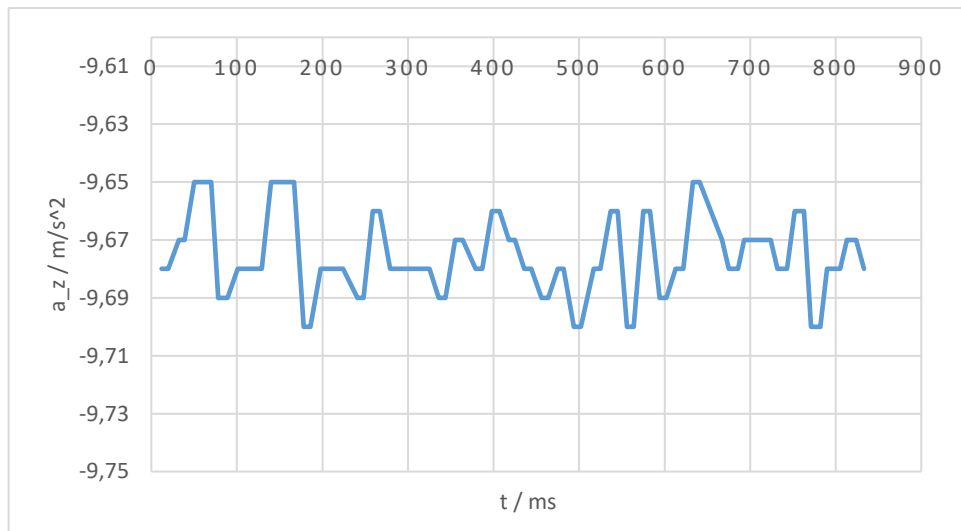
U akcelerometru je nutné si uvědomit, že v klidu funguje na bázi tíhového zrychlení, které je ovlivněno nejenom gravitační silou Země (ovlivněna vzdáleností těžištěm Země a tělesa), ale i rotační odstředivou silou (ovlivněna zeměpisnou polohou tělesa). Je tedy možné si referenční hodnotu vypočítat, nicméně pro jednoduchost lze použít hodnotu  $9,81 \text{ m}\cdot\text{s}^{-2}$ .

Měření teploty pomocí teplotního čidla může být stejně jako v případě barometru porovnáno s meteorologickými hodnotami z blízkých středisek. Jde však poměrně jednoduše dojít k nepřesným údajům při měření. Jakékoliv dopadající sluneční světlo, či tepelně izolovaný prostor a další vlivy mohou narušit přesnost měření. Je proto vhodné se dobře zamyslet nad umístěním senzoru, pro jeho správnou funkčnost.

Gyroskop by měl při převedení do klidu udržovat nulové hodnoty, je tedy vhodné sledovat jeho dlouhodobější stálost.

## 3.3 Filtrace

Při pohledu na Graf č. 1, si lze všimnout náhodně oscilujících dat. Tyto data jsou získaná z akcelerometru, který byl v klidu. Dala by se tedy očekávat rovná horizontální čára, která je však narušena šumem. Šum je v jakékoliv elektronice a jedná se o rušivou informaci, která interferuje s chtěnými daty. Prakticky se ho snažíme co nejvíce potlačit (vyfiltrovat), ale kompletně ho odstranit nelze. [34]



Graf č. 1 – Ukázka šumu v signálu

Důvodů pro vznik šumu může být více a vždy je potřeba rozlišovat, jestli se jedná o šum při práci s hudbou, obrazem, senzory atd. Podle toho se pak mohou různit projevy šumu, či způsob jeho potlačování. Je možné ho potlačovat různými způsoby, kdy příkladem může být stínění kroucené dvojlinky. Jinak se také dá využít softwarová redukce šumu, kdy jsou opět různé variace, jak filtrace docílit. Softwarová filtrace různým způsobem zpracovává průběžně data, což lze provádět jak po měření, tak i během něj. Často, ale takto může dojít k “zmizení” důležitých krátkodobých dat. Je proto dobré vždy mít ke zpracování i nefiltrovaná data. [34]

## 4 Využití práce v rámci výuky

Téma raket vytváří zajímavé propojení mezipředmětových vztahů, kde se kromě informatiky a fyziky dokáže zapojit i dějepis (historie raketového inženýrství), praktické činnosti (dílny), chemie, či výtvarná výchova. Samotná práce má posloužit svým obsahem pro rozšíření výuky. Ať už jako snaha napodobit celý proces tvorby, nebo její jednotlivé části, které mohou vybízet k novým nápadům a inspiracím do výuky. Největším lákadlem je atypičnost. Téma raket není ve školách běžné. Žáky a studenty má velkou šanci zaujmout. Velký potenciál spatřuji především v projektové výuce, kde lze nechat žáky volněji poznávat, experimentovat a případně i soutěžit. Dokáží si představit, že by mohly vzniknout na školách zájmové kroužky a případně i meziškolní soutěže. V rámci internetu lze najít mnoho skupin zabývajících se vodními raketami z Ameriky a Německa, kde se akce pořádají již mnoho let. Největší překážku vidím v přidaném čase (mimo pracovní dobu), ve kterém učitel potřebuje získat dodatečné znalosti, zajistit materiální prostředky, finanční podporu, prostory a vyřešit další nečekané nástrahy, které doprovází nově vznikající odvětví.

### 4.1 Základní škola

Základní pilíř, který dává směr, jakým se má přistupovat k výuce na základních školách je Rámcový vzdělávací program pro základní vzdělávání. V době psaní této práce byla vytvořena úprava z lednu 2021 [35], která bude platná od 1. 9. 2021. Dle této úpravy jsem se řídil při stanovování využití práce v rámci výuky na základní škole.

#### 4.1.1 Informatika

Propojení učiva vzdělávací oblasti informatiky v rámci RVP pro základní vzdělávání s tématem diplomové práce:

Data a informace získané z letu jsou zásadní částí této práce. Proces jejich získávání a ukládání je v této práci dostatečně popsán. Samotná práce s daty se též nabízí, buď jako využití letových dat získaných v této práci (budou v příloze), či získání vlastních, což je také dostatečně vysvětleno, ač v rámci úrovně základní školy nemusí být dosažitelné (vhodné pro nadané žáky).

Modelování jakožto tvorba schémat, myšlenkových map, či vývojových diagramů se nabízí, neboť před započatím tvorby větších celků je vždy dobré si kroky a možnosti graficky znázornit, porovnávat, vyhodnocovat a optimalizovat. Žáci si tak lépe představí a ucelí kroky vedoucí k dokončení komplexnějších úloh.

Algoritmizace, programování, kontrola a samotná tvorba digitálního obsahu je klíčovým krokem k možnostem, které tato práce nabízí. Úrovně vidím tuto část spíše na střední škole, avšak věřím, že pro nadané žáky základních škol by se zde jednalo o možnost

seberealizace. Dá se též očekávat, že s následujícími roky by měla stoupat infromatická znalost žáků, která bude nabízet možnosti dřívější realizace látky, která by byla v předešlých letech za hranicemi možností.

Hromadné zpracování dat vede k možnosti analýzy dat. Ta je v této práci popsána a provedena v praktické části.

Hardware a software samotného Arduina je rozsáhlou možností, jak rozšířit výuku informatiky. Teoretické poznatky popsané v teoretické části a samotná vlastní tvorba v praktické je obsahově nejrozšířenější částí této práce.

#### 4.1.2 Fyzika

Propojení učiva vzdělávací oblasti fyziky v rámci RVP pro základní vzdělávání s tématem diplomové práce:

Měření fyzikálních veličin je výstupem analýzy dat. Jsou zde popsány způsoby jejich měření i jejich principy (z části v [1]). Jsou zde získávány hodnoty metodou přímou i nepřímou, dále i typy statické, dynamické. Pro jejich měření je popsána přípravy měření, vlastní měření a jejich samotné zpracování.

Pohyb tělesa, gravitační síla, tlak, třecí síla, výslednice sil, (Newtonovy zákony), jsou učiva, která se zásadním způsobem projevují při letu raket. Jejich znalost, využití a samotné pochopení může být obohacem informací této práce vylepšeno.

Projevuje se zde dále Pascalův zákon jako důsledek urychlení výtlačku kapaliny z rakety. Atmosférický tlak a jeho pokles s vyšší nadmořskou výškou. Formy energie, jako přeměna tlakové potenciální energie na kinetickou. Přeměna skupenství a probíhající adiabatické ochlazování v trupu při odpalu rakety.

Elektrický obvod obsahující netypické spotřebiče (Arduino, senzory, akcelerometr), zdroj napětí, spínač a LED.

Sluneční soustava a poznání vesmíru jako důvod vesmírného raketového inženýrství (historie, překážky, úspěchy).

### 4.2 Střední škola

Využití na středních školách hodně závisí na samotném typu střední školy a časové dotaci pro dané předměty. Dle typu střední školy se liší právě Rámcové vzdělávací programy. Pro tuto práci jsem se řídil dle RVP pro gymnázia se zněním k datu 1. 9. 2007. [36] V mnoha ohledech se využití v rámci středních škol shoduje se základním vzděláváním, kde je tentokrát však záměrem zajít hlouběji do problematiky. Vzhledem k ucelenějším



znalostem a dovednostem je možné studenty nechat více experimentovat a bádát, tak aby učitel sloužil více jako organizátor a pomocník.

#### 4.2.1 Informatika

Propojení vzdělávacího obsahu v rámci vzdělávací oblasti informatiky a informačních a komunikačních technologií v RVP pro gymnázia s tématem diplomové práce:

Digitální technologie jsou studentům čtyřletých gymnázií již mnohem bližší než na základní škole. Prohloubením hardwarových a softwarových znalostí jsou schopni lépe pochopit principy elektronických zařízení, které jsou součástí této práce. Lepším pochopením mohou začít úspěšně využívat prostředků a znalostí k tvorbě vlastních projektů, díky kterým budou moci získávat vlastní data a využít je v praxi.

Zdroje a vyhledávání informací s komunikací byly pro tuto práci důležitou součástí, bez které by nebyla plnohodnotnou. Stejně tak je zásadní pro studenty, aby jejich tvorba byla podložena ucelenými dokumenty, které splňují náležité parametry (např. autorská práva).

Zpracování a prezentace informací jsou primárně spojeny s prohloubením dovedností a znalostí algoritmických a programovacích. Vzhledem k lepšímu abstraktnímu myšlení jsou studenti schopni lépe pochopit textovou formu programování, kterou dokáží sami vytvořit a upravit, dle zadání. Složitost problematiky této práce bych v tomto ohledu spatřoval na 3. až 4. ročník středních škol.

#### 4.2.2 Fyzika

Propojení vzdělávacího obsahu v rámci vzdělávací fyziky v RVP pro gymnázia s tématem diplomové práce:

Fyzikální veličiny a jejich měření je v rámci možností podrobena větší přesnosti, kde nyní zároveň dochází ke stanovení odchylek měření, které vychází jednak z neshody teorie s praxí (a diskuzí nad důvody), či různým náhodným vlivům. Je tedy obecně ve snaze dojít k přesnějšimu popisu reality, která je zde získaná, právě pomocí senzorů, tak aby mohla být využita k analýze.

Pohyb tělesa (rakety) je spjat s mnohými silami, které mají za důsledek polohové změny. Práce staví na jevech týkajících se učiva kinematiky a dynamiky pohybu, kdy je tak možné z naměřených hodnot získávat důsledky působících sil. Mělo by tedy být možné tyto síly zjistit výpočtem z naměřených hodnot i teoreticky podložených předpokladů a následně je porovnávat.

# Praktická část

Hlavní cíl diplomové práce:

- Analýza letu jednoduché rakety.

Metoda a postup dosažení hlavního cíle:

- Pro dosažení hlavního cíle bude využita metoda sběru dat. Získaná data budou analyzována, tak aby poskytla co nejširší množství hodnot fyzikálních veličin (např. zrychlení, rychlost, ...) a informací popisující let jednoduché rakety.

Vedlejší cíle praktické části diplomové práce:

- A. Sestavení konstrukce vodní rakety, úprava odpaliště, vytvoření měřicího zařízení a jeho naprogramování.
- B. Vyčíslení finanční stránky sestavené rakety, odpaliště a měřicího zařízení použitého v diplomové práci.
- C. Vytvoření pracovních listů pro využití diplomové práce ve výuce na základní, či střední škole.

Metoda a postup dosažení vedlejších cílů:

- Pro dosažení vedlejšího cíle A. bude zásadní si jednotlivé kroky při sestavování rakety, úpravy odpaliště, tvorby měřicího zařízení a jeho programování detailně zaznamenávat, jak pomocí fotodokumentace, tak videozáznamu. Kroky tvorby budou chronologicky zapsány, tak aby je bylo možné napodobit.
- Pro dosažení vedlejšího cíle B. poslouží účtenky a faktury k získání přehledu o ceně rakety, odpaliště a měřicího zařízení.
- Pro dosažení vedlejšího cíle C. bude využit program pro tvorbu výukových materiálů, tak aby pracovní listy byly přehledné, graficky přívětivé a obsahovaly všechny potřebné odpovídající části spjaté s diplomovou prací.

## 5 Tvorba rakety

Prvním krokem, bylo vytvoření nové vylepšené rakety oproti předchozímu modelu TOM I viz Obrázek č. 26, tak aby se mohla zlepšit analýza letu, ale i letové schopnosti. Novou raketu jsem tedy příhodně pojmenoval TOM II.



Obrázek č. 26 – Raketa TOM I

### 5.1 Trup

První věc, kterou při tvorbě vodní rakety musíme brát v potaz je vytvoření trupu neboli té části, která v sobě obsahuje kapalinu a stlačený plyn. Rozhodl jsem se jít cestou celkového objemu 6,75 litrů, který vznikne spojením tří 2,25 litrových PET lahví. Trup tak bude možné naplnit větším množstvím kapaliny a plynu než v předešlém modelu.

Dalším důvodem, proč jsou tři PET láhve vhodné pro konstrukci souvisí s poměrem kapaliny a plynu. Dle pokusů z [37] lze dojít k rozmanitým výsledkům optimálního množství kapaliny pro neoptimálnější letové schopnosti. Dle množství přiložených grafů lze usoudit, že ke stanovení procentuálního naplnění obsahu kapalinou má vliv suchá hmotnost rakety, průměr trysky, objem trupu rakety, hustota kapaliny, vnitřní tlak a mnoho dalších. Základním odhadem bývá zhruba 1/3 objemu trupu rakety, což je v tomto případě právě jedna 2,25 litrová PET lahev. Analýza letu rakety při naplnění různým množstvím kapaliny by tak mohla případně sloužit k optimalizaci letových schopností.

Pro výběr vhodné PET lahve pro vodní rakety, je důležité vybírat takové, které svým tvarem co nejvíce napodobují válec, tak aby nenastávaly turbulence kolem trupu rakety

při letu, které by mohly narušit její stabilitu a zpomalovali ji. Mnohé lahve šumivých nápojů mají designové tvarování, což právě není moc vhodné. Navíc budeme láhve spojovat, takže je vhodné, aby průměr dolní a horní části byl stejný.

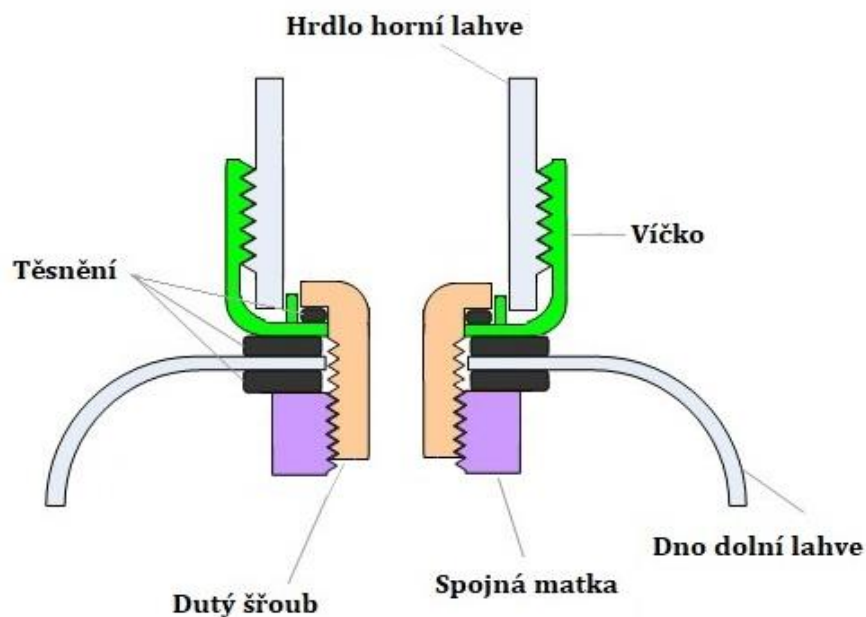
Osobně jsem s tímto měl z počátku problém a spokojil jsem se 2,25 litrovou lahví od Pepsi a vytvořil z ní celý trup. Nicméně jsem následně objevil mnohem lépe válcovitě tvarovanou láhev od Kofoly a rozhodl jsem se trup předělat z ní viz Obrázek č. 27, kde můžeme vidět rozdíl v tvaru PET lahví.



Obrázek č. 27 – Konstrukční PET lahve 2,25 l

### 5.1.1 Spojování

Moje inspirace pro způsob spojování vyházely z videí na youtube, které mě skrze komentáře zavedly k zjištění, že metoda spojování, kterou jsem se rozhodl použít se nazývá Robinson coupling. Jedná se o využití dutého šroubu, kterým prochází plyn a případně kapalina mezi spojenými PET lahvemi viz Obrázek č. 28.



Obrázek č. 28 – Robinson coupling

Na středu dna láhve je proveden otvor o průměru dříku šroubu (velikost šroubu je omezena rozměry hlavy šroubu a vnitřnímu průměru hrdla PET lahve). Stejně velký otvor je proveden do středu víčka lahve. Oba tyto otvory je potřeba opatrně začistit, neboť plast při vrtání má tendenci se horkem z vrtání deformovat. Plochy dna láhve a víčka by měly být, co nejrovnější, tak aby při vkládání jednotlivých těsnění došlo, co k největšímu kontaktu.

Pro jakýkoliv druh spojování je zásadní otázkou pevnost hermetického těsnění, tak aby nedocházelo při vysokém tlaku k úniku kapaliny a vzduchu z rakety, či jejímu roztržení. Právě tato metoda je závislá na kvalitě použitých součástí a preciznosti provedení otvorů. Její hlavní výhodou je možnost rychlého sestavení. [38] Opravy poničeného trupu rakety jsou tak pouze otázkou získání nové PET lahve a chvilky úprav. Pokud je matka na šroubu dobře dotažená a těsnění správně doléhá, je nejslabším článkem na trupu rakety víčko, u kterého je také možné pozorovat různě kvalitní provedení v závislosti na značce výrobce. Opět doporučuji víčka brát pouze z lahví šumivých nápojů.

Tvorba dutých šroubů byla poměrně komplikovaná. Hlavním problémem byla neznalost pevnostních tříd. Jádra kalených šroubů, která jsou klasicky ke koupi, jsou velmi tvrdá a není snadné je provrtat. Je tedy vhodné použít šrouby nekalené do kterých se dá vrtat pomocí běžných vrtáků na kov (či si nechat duté šrouby vytvořit na soustruhu). Použil jsem šestihranné šrouby M12x30. Průměr duté části se odvíjí od přesnosti vrtání. Osobně jsem docílil průměru 7 mm. S lepším náčiním by průměr mohl dosáhnout 8,5 mm.

Následovalo experimentování s různými rozměry těsnění, kdy jsem z počátku využíval i kovové podložky, které jsem později přestal používat, neboť přes ně docházelo k ucházení tlaku. Výslednou kombinací, která se nejlépe osvědčila souhlasí s předchozím

Obrázek č. 28, tudíž byl použit o-kroužek o vnitřním průměru šroubu ve víčku a dvě pryžové podložky, kde je jedna mezi víčkem a spodkem druhé PET lahve a druhá mezi matkou a z vnitřku druhé PET lahve.

Propojené PET lahve je vhodné opatrně natlakovat (doporučuji přes ně přehodit peřiny a mezi raketu a sebe dát ochranou bariéru), tak aby se zjistilo, že nikde neuniká vzduch. Testovací velikost tlaku by měla být minimálně o půl atmosféry více, než v ní bude při odpalu. Pokud vše těsní, tak se na raketu přidá tzv. kapotáž, která kryje mezery mezi PET lahvemi. Kapotáž je vytvořena z válcovitého výstřižku stejného typu PET lahve, která je použita na tvorbu trupu. Následně je válcovitý tvar podélně rozstřižen, tak aby se dal na trup nasadit, bez nutnosti povolení šroubů. Po nasazení kapotáže na trup je nutné její zajištění pomocí lepící pásky.



*Obrázek č. 29 – Trup rakety TOM II s přilepenou kapotáží*

Kapotáž je důležitá nejen pro dobrou aerodynamiku, ale i pevnost a rovnost konstrukce. Samotné propojení PET lahví dutými šrouby je poměrně flexibilní a nemusí být zcela rovné. Precizním přilepením kapotáže na trup, tak vyřešíme dané nepřesnosti a nedostatky.



*Obrázek č. 30 – Detail kapotáže*

## 5.2 Hlavice

Pro tvorbu hlavice jsem se rozhodnul využít PET lahev z 2,25 litrové Pepsi, jenž má lépe tvarovanou špici než láhev od Kofoly. Tím by se měla zlepšit aerodynamika a vzhled

rakety. Ve výsledku byla potřeba 2 lahví k tomu, aby bylo možné vytvořit místo pro padák a osazení pro měřící zařízení. Hrdlo lahve je potřeba opatrně odstranit a dotvarovat, na co nejaerodynamičtější tvar. V mém případě jsem využil polovinu pingpongové míčky, který jsem připevnil lepidlem. Později se podle rozměrů osazení pro padák a míst, kde je potřeba se dostat do vnitřku hlavice, vytvoří potřebné otvory.



Obrázek č. 31 – Dva výstřižky PET lahve Pepsi na hlavici

### 5.2.1 Padák

Padák slouží ke zpomalení rychlosti dopadu rakety. Optimálně by se měl rozevřít krátce po dosažení vrcholu letu a raketu tak začít zpomalovat dříve, než získá velkou rychlost způsobenou volným pádem. Vysoká rychlost při otevření padáku by mohla způsobit poničení struktury rakety, odtržení padáku, či jiné nežádoucí problémy.

Tvorba padáku je poměrně jednoduchá. Využil jsem starého deštníku. Stačilo opatrně odstříhnout jednotlivá připevnění ke konstrukci. Z režné nitě jsem zhotovil osm metrových odstřížků, které jsem připevnil k cípům. Zhruba v půlce a ke konci jsem vytvořil pevné uzly, tak aby se padák optimálně rozevřel a bylo ho možné později nastavit.



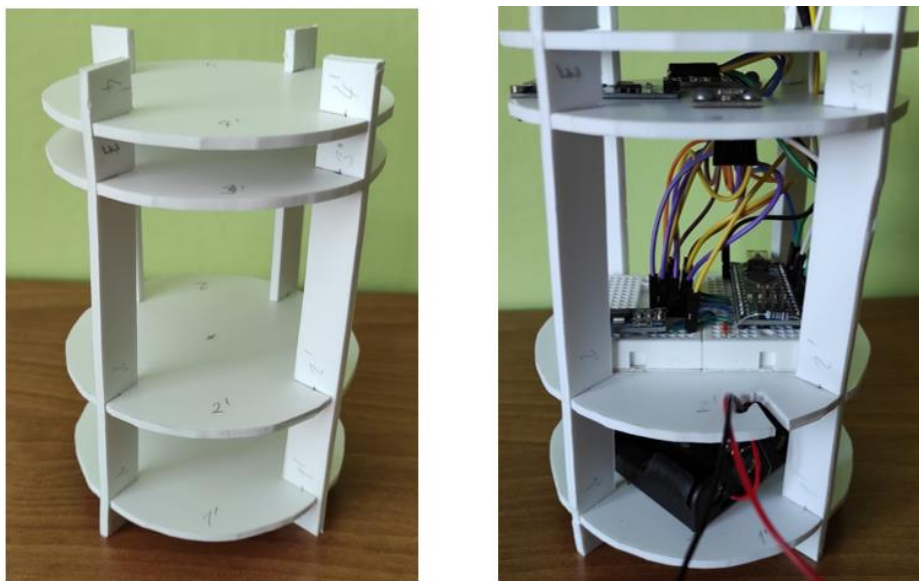
Obrázek č. 32 – Padák

Důležitou součástí při práci s padákem je jeho zabalení, tak aby jeho otevření nastalo rychle a bez komplikací. Proces balení padáku je srozumitelně popsán zde: [39].

### 5.2.2 Osazení pro měřící zařízení a padák

Jako materiál na tvorbu osazení pro měřící zařízení, moduly, servomotor a padák, jsem využil plastové desky GUTTA Hobbycolor z tvrdé pěny. Je poměrně pevná, lehce flexibilní, levná, dá se dobře řezat odlamovacím nožem, při použití tavné pistole se z ní dá sestavit pevná konstrukce a je k dostání v několika barvách. Jelikož se jedná o materiál, který lze horkovzdušnou pistolí tvarovat, zvolil jsem provedení v bílé barvě. Uvědomil jsem si, že v letních měsících, kdy se vodní rakety často vypouštějí, mohou být poměrně horké dny, a právě Sluncem ozařované části raket je dobré vytvářet, tak aby energii pohlcovaly, co nejméně a byla, zde malá šance jejich deformace. V praxi si toho pak lze všimnout na skutečných vesmírných raketách (jejich povrch má bílou barvu).

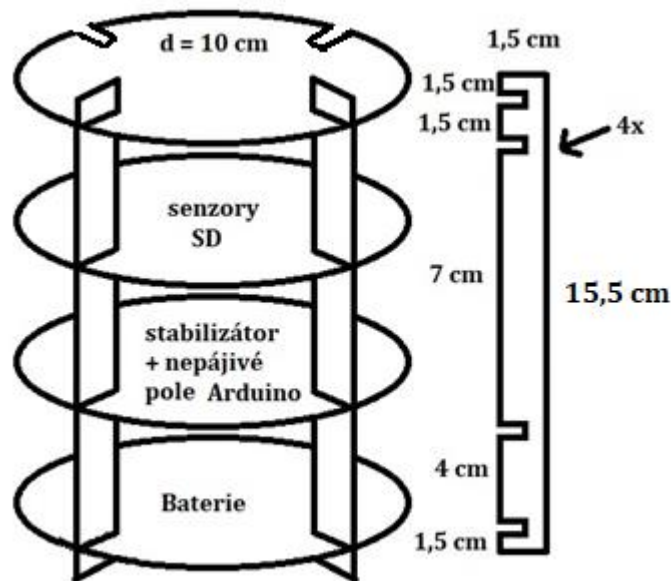
Konstrukce osazení vycházela čistě z vlastního nápadu. Vytvořil jsem 4 disky o vnitřním průměru hlavice, do kterých jsou zasunuty 4 pilíře (nebylo zde použito lepidlo). Celkově mám tak 4 patra, kam mohu vložit potřebné komponenty měřícího zařízení viz Obrázek č. 33



Obrázek č. 33 – Osazení pro měřící zařízení před a po vložení komponentů



Vytvořil jsem též náčrt s rozměry viz Obrázek č. 34.



Obrázek č. 34 – Náčrt pro osazení měřícího zařízení

V třetím patře, kde jsou senzory a modul na SD kartu jsem nechal málo místa, tak aby byly, co nejvíce kryté od Slunce a byly co nejvíce v bezpečí. Modul na SD kartu jsem nasměroval a připevnil na okraj disku, tak aby bylo možné později do hlavice zhotovit otvor pro vyjmutí a zasunutí SD karty, bez nutnosti demontáže.

V druhém patře, je na rozdíl od třetího potřeba místa více, neboť bude potřeba vytvořit dostatek prostoru pro práci s vodiči, tak aby se dalo vše jednoduše propojit.

Pro osazení na padák a servomotor jsem vytvořil další dva disky a dva obdélníky, které jsem k sobě připevnil tavnou pistolí, tak aby vytvořily kapsu. Stačilo nasadit gumičky a připevnit k nim kus obdélníku pro rovnoměrnější vytlačení padáku viz Obrázek č. 35.



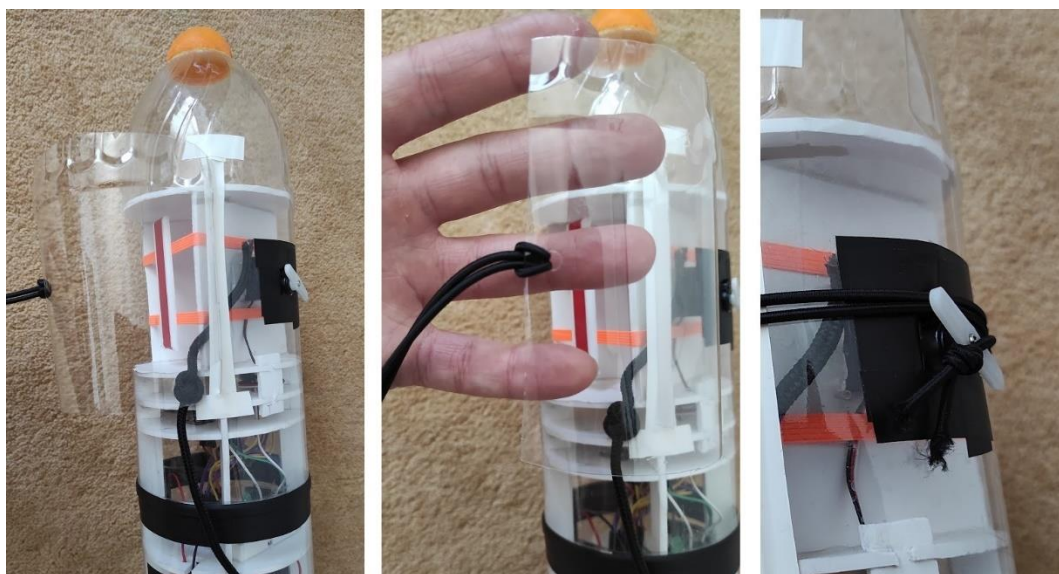
Obrázek č. 35 – Osazení pro padák a servomotor

Následuje zkontrolování zapojení elektroniky, připevnění servomotoru k osazení pro padák pomocí tavné pistole, vložení baterií do boxu a osazení tak lze vložit do hlavičky.



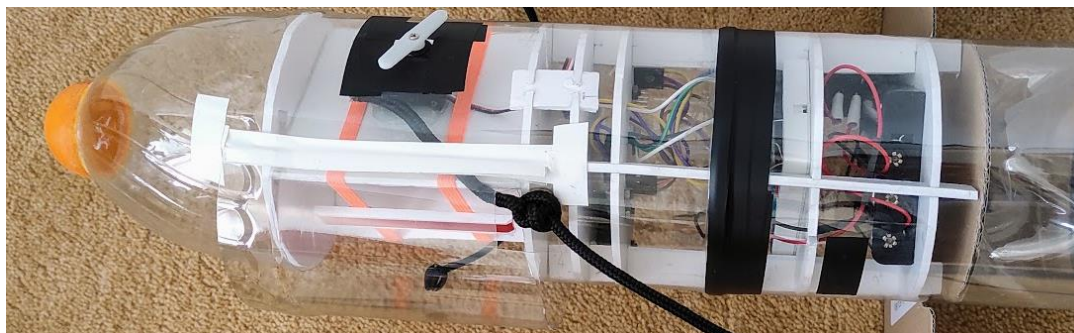
Obrázek č. 36 – Osazení pro měřící zařízení a padák s hlavičkou před sestavením

Po vložení osazení do hlavičky je potřeba provést několik úprav na hlavičce. Malý otvor pro SD kartu, kruhový otvor pro spínač a dírky pro uchycení lanka pro padák v pravém dolním rohu pod osazením pro padák. Dalším důležitým prvkem na hlavičce jsou dvířka hlavičky pro padák. Ta jsou vystřížena z nové Pepsi lahve, tak aby co nejvíce lícovala a byla zhruba o centimetr větší ze všech stran než samotný otvor na hlavičce. Z jedné strany jsou přidržované lepicí páskou a z druhé mají malý otvor pro gumičku, která je bude držet zavřené, přičemž bude zaháknutá za servomotor viz Obrázek č. 37.



Obrázek č. 37 – Dvířka hlavičky pro padák

Sestavená hlavičce s osazením a funkční elektronikou uvnitř může být takto připevněna na různé trupy raket, mající stejný průměr.

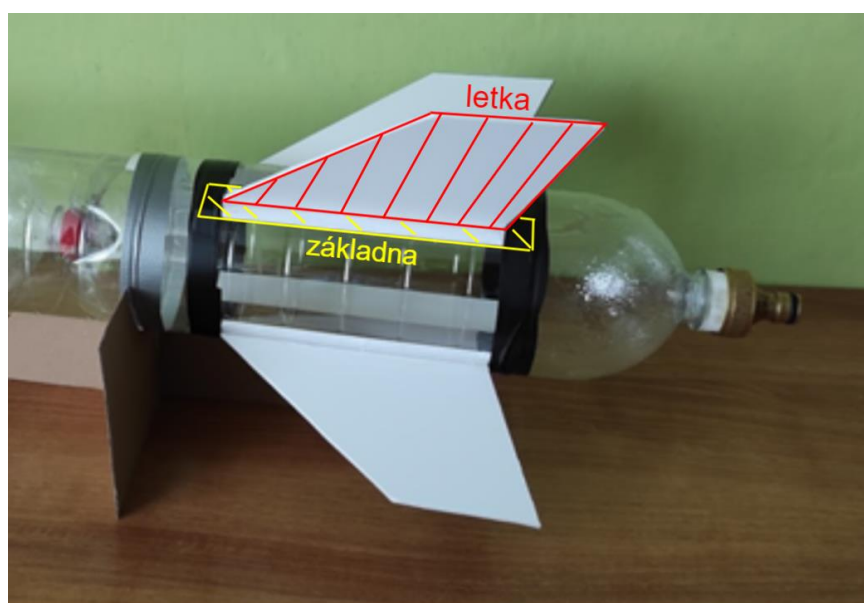


Obrázek č. 38 – Kompletní hlavičce

### 5.3 Letky – stabilizátory

Menší úpravou si prošly letky rakety TOM II. Rozhodl jsem se vytvořit raketu, která by měla letky 4. Jejich rozmístění tedy bude vždy po  $90^\circ$  kolem trupu. Pro co největší efektivitu se umístí, co nejnižže na trupu rakety. Zde se právě vyplatí mít válcovitý trup, ke kterému je možné letky dobře připevnit. Kromě většího množství letek, jsem se také rozhodl vylepšit materiál. Staré letky byly vytvořeny z kartonu přelepeného lepící páskou, což rozhodně nemusí být špatné, neboť byly velmi lehké a levné. Byly však hodně ohebné a pokud se pod lepící pásku dostala vlhkost, což je při práci s octovou a vodní raketou poměrně snadné, byly prakticky nepoužitelné.

Využil jsem tedy stejného materiálu jako pro osazení, se kterým se velmi dobře pracuje. Zkonstruoval jsem tedy 4 letky viz Obrázek č. 39, kde si lze všimnout základny, na kterou je tavnou pistolí připevněna samotná letka.



Obrázek č. 39 – Upevněné stabilizátory

Takto je tedy možné základnu s letkou lépe a jednodušeji připevnit k trupu rakety, ideálně lepidlem určeným na lepení plastů, či omotáním lepící pásky. Rozhodně není vhodné použít tavnou pistoli, neboť by horké lepidlo zdeformovalo, případně zničilo trup rakety.

## 5.4 Tryska

Tryska z rychlopojky se na předchozí raketě TOM I osvědčila, tudíž v tomto ohledu jsem se rozhodl ji použít znovu. Změna však musela nastat v držícím nástavci, který musí být nyní průchozí, tak aby šlo raketu natlakovat. To je popsáno v části o úpravách odpaliště.



Obrázek č. 40 – Tryska z rychlopojky

Pro správné těsnění trysky, po jejím nasazení na hrdlo trupu, je zásadní její pevné dotažení (doporučuji pomocí síkových kleští) a využití pryžového o-kroužku na hrdlo láhve, pro ještě lepší těsnění.

## 5.5 Kompletace rakety

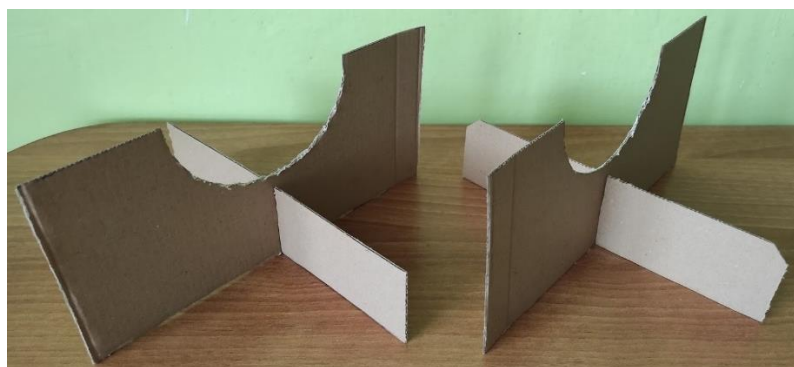
Nezbývá tedy než zkompletovat všechny části rakety v jeden funkční celek. To zahrnuje připevnění hlavičky k trupu, připevnění padáku a držáků stojných tyčí.

Spodní část pilířů osazení uvnitř hlavičky má poměrně hranaté tvary, které by mohly poničit trup, proto jsem se rozhodl udělat ještě jeden disk z kartonu, který je rozdělí.



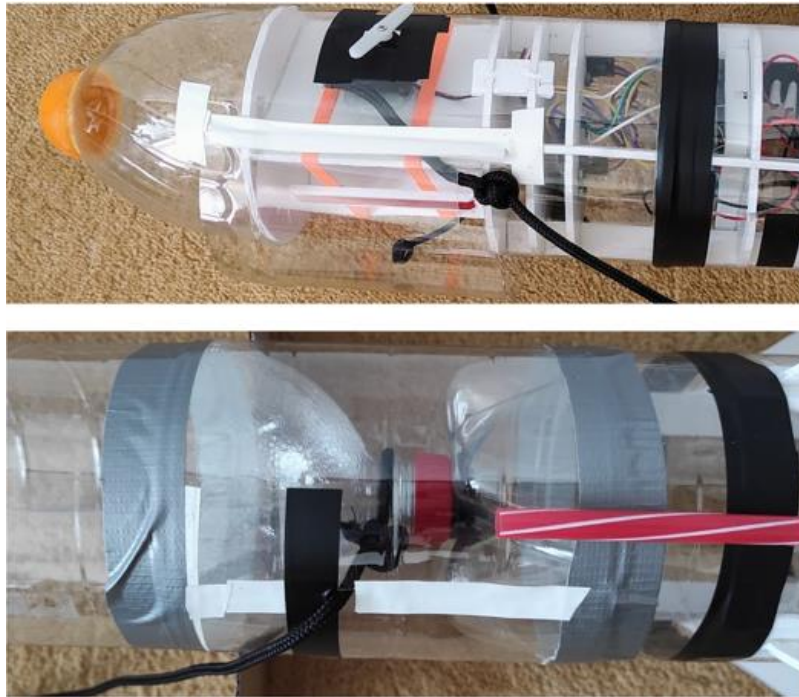
Obrázek č. 41 – Kartonový disk mezi osazením a trupem

Pro pohodlnější manipulaci s raketou a zamezení poničení stabilizátorů jsem vytvořil dva podstavce z kartonu.



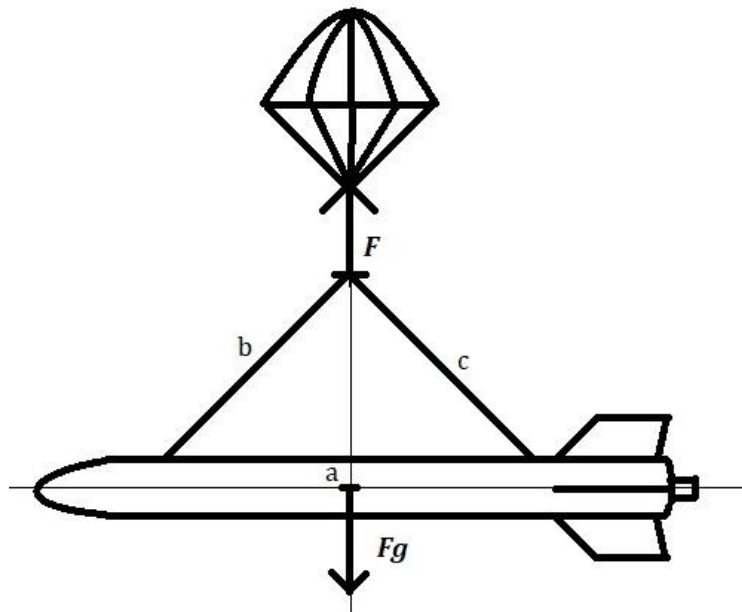
Obrázek č. 42 – Podstavce na raketu

Pro připevnění padáku jsem se rozhodl pro dvoubodové uchycení, tak aby raketa padala horizontálně k zemi. Tak by se mělo využít poměrně velké plochy rakety k dodatečnému zpomalení a poměrně bezpečnému dopadu. Jeden bod úchyty je centimetr pod osazením pro padák, kde jsem vytvořil dvě dírký do hlavičky. Pro druhý bod úchyty jsem využil hrdla střední lahve z trupu.



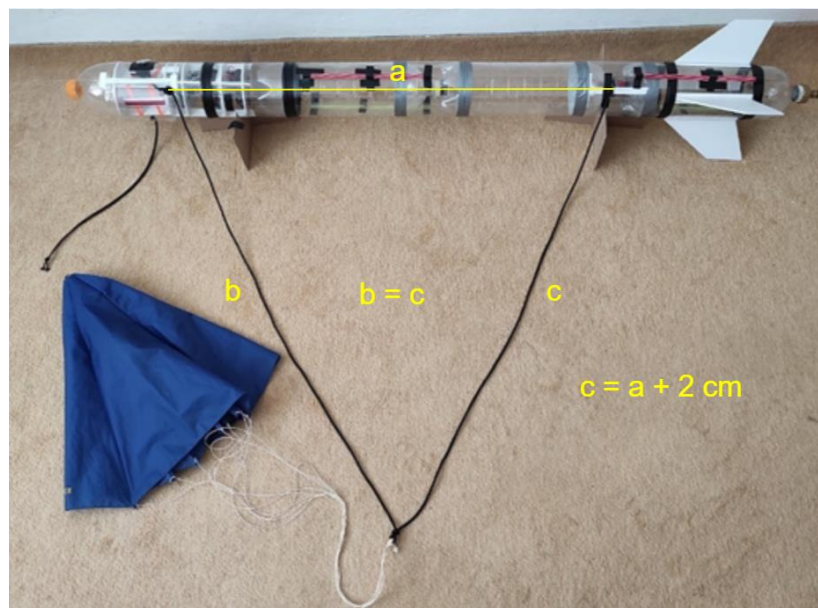
Obrázek č. 43 – Body úchytu pro lanko padáku

Využil jsem paracordové lanko, které je velmi pevné a nepruží. Padák jsem k lanku navázal, tak aby podle Obrázek č. 44 byla strana c zhruba o 2 centimetry delší, než strana a (obrázek tomu úplně neodpovídá), proto aby při složení padáku bylo lanko strany c, co nejvíce napnuté podél rakety.



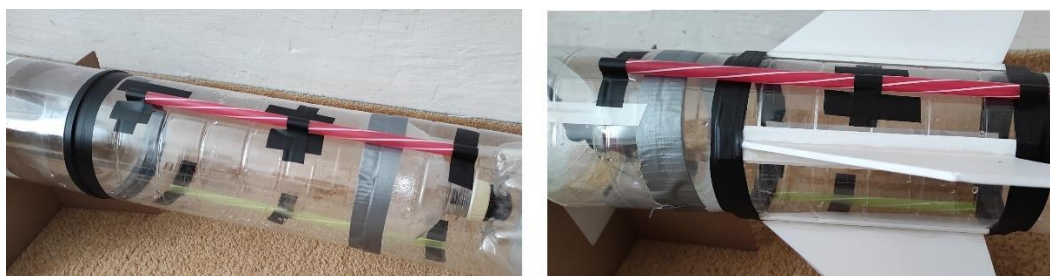
Obrázek č. 44 – Schéma uchycení padáku

Celkově tak raketa bez složeného padáku vypadá následovně viz Obrázek č. 45



Obrázek č. 45 – Raketa TOM II bez zabaleného padáku

Posledním krokem před kompletací je připevnění držáků stojných tyčí. Ty slouží k přidržení rakety, před a krátce po startu, stejně jako to bylo provedeno na raketě TOM I. Tentokrát jsem pouze přilepil k raketě dva páry místo jednoho viz Obrázek č. 46. Jedná se v podstatě o rovná brčka, jenž mají vnitřní průměr o 1 mm větší, než je průměr stojných tyčí, tak aby šly tyče do nich volně zasunout. Pro jejich připevnění jsem raketu připevnil k odpališti a zasunul stojné tyče. Na tyče jsem postupně nasunul jednotlivá brčka a v místech, kde se tyče dotýkaly trupu rakety bylo vhodné brčka přilepit lepící páskou, nejlépe na třech bodech nahoře, uprostřed a dole.



Obrázek č. 46 – Držáky stojných tyčí

Tím je tedy raketa kompletní a připravená k odpalu.

### 5.5.1 Parametry rakety TOM II

NÁZEV RAKETY	TOM II
SUCHÁ HMOTNOST	780 g
VÝŠKA	1,45 m
PRŮMĚR TRUPU	10 cm
VZDÁLENOST TĚŽIŠTĚ OD TRYSKY (SUCHÁ HMOTNOST)	0,8 m
VNITŘNÍ PRŮMĚR TRYSKY	1 cm
POČET LETEK	4
ODZKOUŠENÁ HERMETIČNOST DO	5 atm

### 5.6 Úpravy odpaliště

Odpaliště je nedílnou součástí vodních raket. Tato část slouží jako základna, se kterou se celá raketa před startem spojuje. Tvoří raketě její oporu a splňuje další zásadní funkce. Na odpališti je umístěn uvolňovací mechanismus, který zaručuje, že raketa samovolně nevzlétne, dokud tak není námi provedeno. Skrze uvolňovací mechanismus musí být možné raketu natlakovat a ideálně i naplnit vodu do trupu.

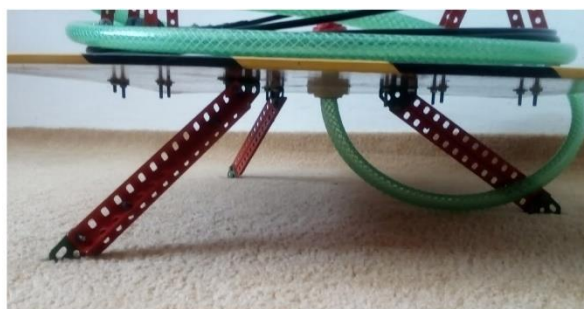
Využil jsem původní odpaliště z [1] a provedl jsem 5 zásadních změn. První byla nutnost zakoupení nové spojky. Ta pro raketu TOM I byla zaslepená, neboť by jinak nešlo natlakovat octovou raketu. Následně bylo potřeba vyřešit její připevnění ke konstrukci, což jsem nakonec po různých pokusech provedl pomocí šroubovacích spon. Nakonec jsem jen připevnil povolovací mechanismus, který byl již vytvořený.



Obrázek č. 47 – Připevnění nezaslepené spojky

Zadruhé bylo potřeba vytvořit způsob, jak natlakovat raketu. K tomu jsem využil zahradní hadici a bezdušový auto ventilek. Zároveň jsem si zakoupil kvalitní hustilku se zabudovaným manometrem, pro lehkou kontrolu tlaku.





Obrázek č. 48 – Přívodní hadice pro tlakování rakety



Obrázek č. 49 – Hustilka s manometrem

Třetí změnou, byly podpůrné tyče, které drží raketu zpříma a pomáhají jí tak udržet kolmý start, dokud si raketa nezíská vlastní stabilitu. V minulém provedení byla tyč pouze jedna o délce 1,5 m. Pro raketu TOM II jsem zakoupil sloupky dva o délce 2 m (průměr 6 mm). Raketa by tam měla mít podporu po delší dobu, a navíc by neměla získat hned od startu rotaci, která je pro let raket nežádoucí.



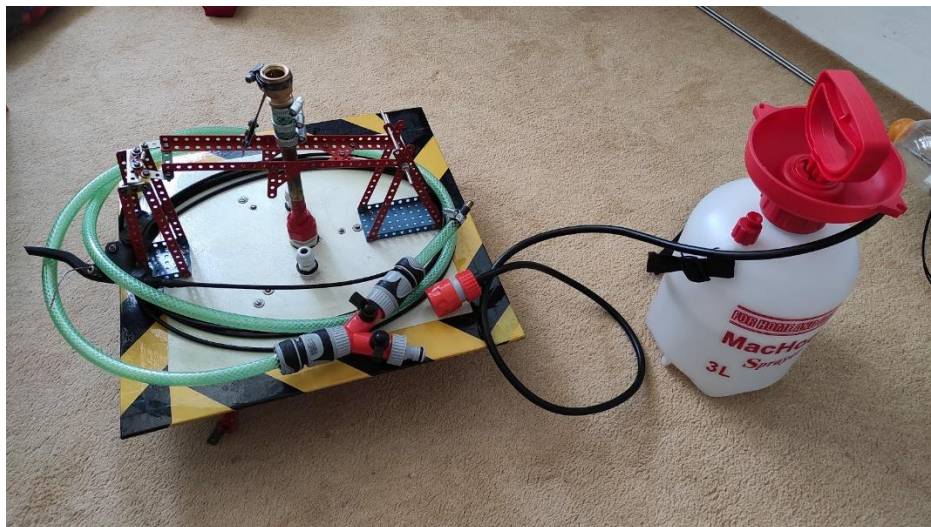
Obrázek č. 50 – Dva přidržovače podpůrných tyčí

Čtvrtou změnou, byl pohotovostní rozbočovač, který slouží jako možnost vypustit a odtlakovat raketu v případě, že bych chtěl bezpečně raketu vrátit do nenatlakovaného stavu.



Obrázek č. 51 – Rozbočovač

Poslední pátá změna byla využití rozbočovače k možnosti naplnění rakety vodou pomocí připojené tlakovací nádoby. Zakoupil jsem zahradní postřikovač s objemem 3 l a rychlospojku viz Obrázek č. 52.



Obrázek č. 52 – Postřikovač jako tlaková nádoba pro přívod vody do rakety

## 6 Měřicí zařízení – hardware

### 6.1 Servomotor

Když jsem přemýšlel, jak budu vystřelovat padák sestrojil jsem nejprve řešení mechanické, kde byl natahovací strojek (z hračky), který by po několika sekundách padák uvolnil. Poté jsem si uvědomil, že by mohlo dojít k uvolnění padáku brzy, či pozdě, neboť vlastně nevím, jak raketa poletí. Došel jsem tedy k závěru, že bude výhodné použít data ze senzorů k tomu, abych padák vypustil ve správný okamžik. Nabízela se možnost použít data z výškoměru (barometru) a akcelerometru, jenž by daly vědět servomotoru, kdy se má natočit a vypustit padák.

Využil jsem servomotoru, který byl součástí začátečnického kitu pro Arduino, přesněji řečeno se jedná model z Obrázek č. 21. Nejprve jsem se s ním naučil pracovat samostatně přes příklady přiložené v IDE Arduina. Zjistil jsem, že ač by měl být schopen se natáčet v rozsahu  $0^\circ - 180^\circ$ , tak se jeho skutečný rozsah pohyboval spíše v  $0^\circ - 160^\circ$ . Toho si pak lze všimnout v kódu.

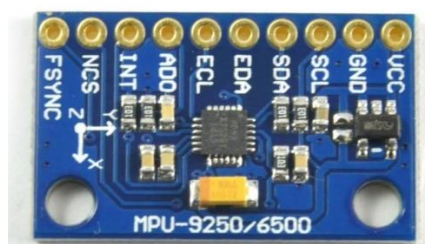
Kromě rozsahu jsem později došel ke zjištění pomocí měření, že servomotor při pohybu odebírá poměrně značný proud (až 200 mA). Jak tedy bude později ukázáno na schématu zapojení, bude servomotor přímo napájen skrze baterie, čímž tak nehrozí desce Arduina nebezpečí překonání maximálního odběru proudu (Příloha č. 1 – maximální odběr pro celé zařízení je 200 mA).

### 6.2 Senzory

V mém původním plánu bylo pouze použití IMU modulu. Došel jsem však k uvědomění, že by bylo v rámci analýzy zajímavé zjistit data z více zdrojů, tak aby se daly porovnávat a šlo z toho odvodit více závěrů.

#### 6.2.1 MPU-9250 (IMU)

Samotný modul je kombinací modulu MPU-6500 (akcelerometr + gyroskop) a magnetometru (kompasu) v jednom osazení. Pro tento modul jsem se rozhodl, jako záchranné hledisko, které nakonec posloužilo jako vylepšení.



Obrázek č. 53 – MPU-9250

V původním zpracování jsem používal modul MPU-6050 (akcelerometr + gyroskop), jenž je sám o sobě také velmi schopným modulem, nicméně jsem s ním měl dlouhodobé komplikace. Projevily se při externím napájení, kdy všechny hodnoty, které jsem z něj získával byly pouze nuly. Snažil jsem se problém řešit měřením napětí, proudů, úpravami kódu atd. Nic nepomáhalo, a tak jsem nakonec rozhodl zakoupit právě nový modul MPU-9250, který by podle dokumentace měl být dokonce výkonnější (vyšší možná frekvence sběru dat) v rámci gyroskopických měření.

### 6.2.2 BME280 (Barometr + Teploměr)

Pro měření tlaku a teploty jsem se rozhodl zakoupit senzor BME280 vyráběný firmou Bosch. Kromě toho má možnost měřit i vlhkost, což mi přišlo pro tuto práci zbytečné, ale do budoucna by se mohl využít i jinak, například jako domácí meteorologická stanice.



Obrázek č. 54 – BME280

Dle [40] lze za pomoci teploty, atmosférického tlaku a tlaku vzduchu na hladině moře dojít k výpočtu výšky za pomoci hypsometrického vzorce.

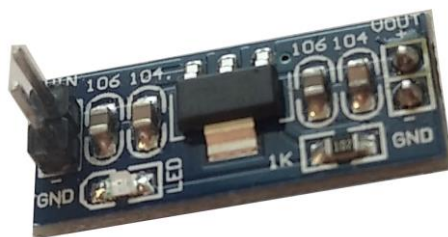
$$h = \frac{\left( \left( \frac{P_0}{P} \right)^{\frac{1}{5,257}} - 1 \right) \cdot (T + 273,15)}{0,0065} \quad (1)$$

kde  $T$  je teplota v °C,  $P_0$  je tlak vzduchu přepočtený na hladinu moře v hPa a  $P$  je místní tlak vzduchu v hPa.

## 6.3 Napájení

Napájení je vyřešeno pomocí zapojení čtyř AA lithiových baterií do boxu, v němž jsou propojeny do série pro získání dostatečného napětí (>6 V). Box na baterie v základu neměl vypínač, tudíž by elektrický proud začal procházet obvodem hned jakmile bych připojil poslední baterii. Byl proto připájen vypínač viz Obrázek č. 19.

Stabilizaci napětí je vyřešena pomocí externího 5 V lineární stabilizátoru viz Obrázek č. 55.

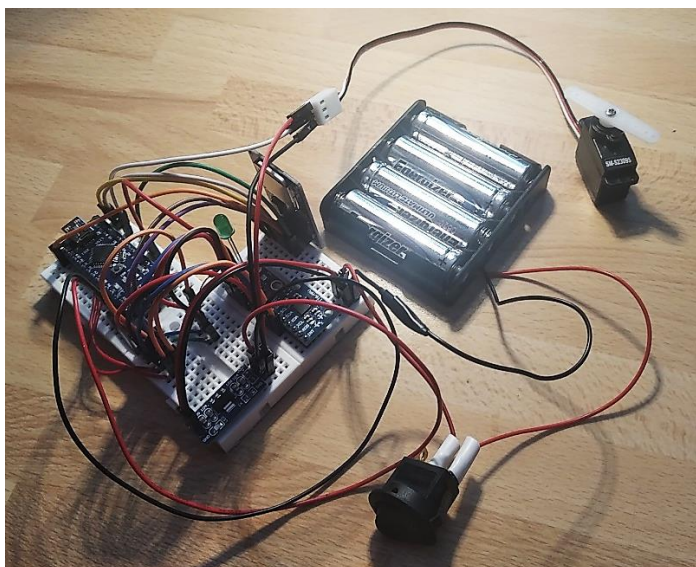


Obrázek č. 55 – 5 V lineární stabilizátor napětí

Vstupní napětí 6 V–12 V převádí na 5 V. Maximální výstupní proud je 800 mA.

## 6.4 Zapojení

Pro pracovní zapojení všech součástí jsem využíval malých nepájivých polí a mnoha M-M propojovacích vodičů, jak lze vidět na Obrázek č. 56.



Obrázek č. 56 – Pracovní zapojení

Zásadní otázkou pro mě bylo, jak neoptimálněji toto všechno dát do hlavičky rakety, tak aby bylo vše pevně a bezpečně umístěné. K tomu jsem využil konstrukci, která je zobrazena v části 5.2.2. Výsledek tak vypadá následovně viz Obrázek č. 33.

### 6.4.1 Schéma

Pro tvorbu schématu zapojení jsem využil programu Fritzing. Schéma je obsaženo v datové příloze, tak i jako Příloha č. 2. Program Fritzing je placeným programem, který obsahuje velké množství dat o součástkách a dovoluje tvorbu schémat pomocí drag

and drop. Některé součástky, které jsem fyzicky využil, však nemá v daném provedení, avšak z praktického hlediska to na funkci vliv nemá.

## 7 Měřicí zařízení – software

V této části se zaměřím na způsob, jakým jsem přistupoval k tvorbě programu, jenž byl nahrán do paměti Arduina. Zároveň jednotlivé části kódu popíši, tak aby zájemcům dávaly smysl a případně se s nich něco naučili.

### 7.1 Rozbor kódu

Část programovací byla pro mě poměrně trnitá, neboť jsem se vše učil jako samouk. V rámci výuky jsem se učil nejčastěji skrze videa na youtube a fóra (Stack Overflow, GitHub, Arduino), kdy jsem se nejprve zaměřil na plné pochopení jednotlivých modulů a práce s nimi. Potom jsem jednotlivé funkční programy sjednocoval v jeden celek. Kromě digitální přílohy jsem kód vložil i jako Příloha č. 3–7.

Kód jsem pro smysluplnost a přehlednost rozdělil na tři části: inicializační, nastavovací a část datového sběru.

#### 7.1.1 Inicializační část

Nejpodstatnější úlohou inicializační části je importace knihoven do paměti Arduina. Knihovny v sobě obsahují množství funkcí, které jsou podstatné pro efektivní práci s jednotlivými moduly. V rámci práce v IDE Arduina je možné najít jednoduše mnoho knihoven s odkazy na jejich bližší popis. Velmi výhodné je využít vytvořených příkladů, které jsou často součástí knihoven, čímž lze lépe pochopit využití jednotlivých funkcí, které knihovny nabízí. Obecně by se dalo říci, že knihovny práci usnadňují, ale je důležité si uvědomit, že mohou zabírat poměrně mnoho paměti, které v případě Arduino Nano tolik není.

```
14 //Inicializace senzoru BME280 (barometr - výškometr + teploměr)
15 #include <Arduino.h>
16 #include "BMx280I2C.h"
```

Pro importování knihoven se používá syntaxe `#include <název_knihovny.h>`, či `#include "název_knihovny.h"`. Jednoduše řečeno se pro importaci již systémových (nestažených) knihoven použijí `<>` a pro stažené a lokálně uložené knihovny `"`. Za touto syntaxí se napíše středník.

Kromě knihoven zde dochází k přiřazení pracovních názvů jednotlivých modulů.

```
11 MPU9250 IMU
18 BMx280I2C bmx280
23 File myFile;
27 Servo myservo;
```

Později se tak k jednotlivým modulům budou volat funkce dle jejich pracovních názvů.

Vzhledem k tomu, že získávám data ze dvou senzorů (zařízení) je potřeba znát jejich adresy (podtržené) pro práci přes I<sup>2</sup>C protokol.

```
11 MPU9250 IMU(Wire, 0x68);  
17 #define I2C_ADDRESS 0x76  
18 BMx280I2C bmx280(I2C_ADDRESS);
```

Ty se dají zjistit pomocí knihovny i2cdetect, nebo přes dokumentaci, či internet. Zapisují se klasicky v hexadecimálním zápisu, který lze poznat podle předpony 0x. V kódu se musí vložit do argumentu pracovního názvu senzoru, tak aby Arduino rozpoznalo, z kterého senzoru přichází data.

Poslední úlohou inicializační části je deklarace globálních proměnných, které jsou později využity v kódu.

```
29 //Inicializace proměnných  
30 long timer, d_t, timer_helper;  
31 int const pinCS = 4;  
32 byte buff = 0;  
33 boolean servo_stop = true;  
34 byte servo_help = 0;
```

Občas se může stát, že bude potřeba co nejvíce minimalizovat datovou náročnost kódu. Zde je právě jedna z možností, kde je dobré se zamyslet, které datové typy, chceme jednotlivým proměnným přiřadit.

### 7.1.2 Nastavovací část

Při pojmenování této části jsem vycházel z názvu funkce, která je základním stavebním kamenem programů psaných pro Arduino. Jedná se o funkci void setup ().

```
38 void setup() {
```

Celá tato část je její součástí.

Prvním krokem je nastavení přenosové rychlosti dat neboli tzv. baud rate, která udává množství přenesených bitů za jednotku času (bps = bit per second). Arduino obecně podporuje řadu určitých hodnot, kdy klasickou hodnotou, se kterou se setkáme v rámci mnoha návodů je 9 600 bps. Lze se setkat s moduly, které mohou mít firmware připravený na různé hodnoty, čemuž se dá podřídit, nebo je lze i přeprogramovat. V rámci této práce jsem použil přenosovou rychlost 38 400 bps.

```
40 Serial.begin(38400);
```

Pro práci se servomotorem je potřeba mikrokontroleru Arduina deklarovat, na kterém pinu je zapojený.

```
43 myservo.attach(9);
```

V rámci toho voláme funkci `attach()` na pracovní název servomotoru, jenž jsme vytvořili v inicializační části. V argumentu je tedy číslo pinu (9), přes který bude probíhat komunikace. Dále nastavuji, aby se servo nastavilo na jednu z krajních hodnot.

```
44 | myservo.write(0);
```

V argumenty funkce `write()` se zapisuje hodnota úhlu, kdy 0 je krajní pozice, při které se zajišťuje padák a čeká se na přetočení rakety.

Vhodnou testovací a ověřovací metodou, kterou jsem začal používat pro debugging je využívání LED, které jsem nechával rozsvěcet v různých fázích tvorby kódu, když mi něco zrovna nešlo, jak mělo, abych našel, kde už mě kód nefunguje. Nakonec jsem se rozhodl jednu zelenou LED ponechat pro indikaci, že probíhá sběr dat a kontrolu funkčnosti akcelerometru (bude rozebráno v poslední části). V nastavovací části se musí uvést LED, tak aby Arduino vědělo, že se přes daný digitální pin bude jednat o výstupní zařízení, do kterého se bude dodávat napětí.

```
47 | pinMode(3, OUTPUT);
```

Abych mohl zapisovat data do micro SD karty je potřeba modul pro práci s SD kartou spustit (zkontrolovat, zda funguje).

```
50 | if (!SD.begin(pinCS)) {  
51 |     Serial.println(F("Inicializace SD karty selhala!"));  
52 |     while (1) {}  
53 | }
```

Funkce `SD.begin()` vrací hodnotu Pravda, pokud spuštění proběhlo v pořádku, proto je před ní vykřičník, který její logiku neguje, aby mi podmínka sdělila případný problém (zprávu), pokud při spuštění modulu problém nastane. Následně pak na řádce č. 52 (později i na řádce č. 66 a 75) dochází k nekonečnému zacyklení, které slouží jako brzda. Pokud tedy nastane problém a bude Arduino připojeno k počítači, vypíše se problém do sériového monitoru (a SD karty, zdali funguje), pokud však nastane problém mimo připojení přes počítač zjistím tak problém právě přes nerozsvícenou LED, či přečtením dat z SD karty.

Při kompilaci kódu se v dolní části IDE Arduina objeví hlášení o datové velikosti programu. Celkem rychle mi došlo, že je potřeba vytvořit úsporné řešení, neboť se mi začalo ukazovat hlášení o malé dostupné paměti s varováním na problémy se stabilitou. Zmírnění nároků na dynamickou paměť mi pomohlo použití makra `F()`, kterým jsem zaobalil všechny texty pro výpis textu.

```
51 |     Serial.println(F("Inicializace SD karty selhala!"));
```

Obecně doporučuji jakýkoliv holý text do tohoto makra ihned vložit.

V případě úspěšného spuštění modulu SD karty je provedeno nastavení módu zápisu viz řádek č. 55.



```

54 else {
55     myFile = SD.open("test.txt", O_CREAT | O_WRITE | O_APPEND);
56     myFile.println(F("Inicializace SD karty uspesna!"));
57     Serial.println(F("Inicializace SD karty uspesna!"));
58 }

```

Testováním rychlostí zápisů dat na SD kartu různými modifikacemi řádku č. 55, jsem došel k metodě, která jej razantně urychluje, právě za pomoci daného zápisu v druhém argumentu funkce SD.open(první argument, druhý argument). Dle poznatků k tomuto zápisu, které jsou zde [41], může i nemusí k urychlení zápisu vždy dojít, avšak zde se zlepšení projevilo.

Nastavení senzoru MPU-9250 je za pomoci knihovny MPU9250.h velmi snadné. V základu má tato knihovna nastavena rozsah akcelerometru  $\pm 16$  g a rozsah gyroskopu  $\pm 2000$  °/s. To jsou maxima, která bych nastavil, kdyby to bylo jinak. Stačí tedy zjistit, že došlo k navázání komunikace, což je provedeno následovně.

```

61 status = IMU.begin();
62 if (status < 0) {
63     Serial.println(F("MPU fail!"));
64     myFile.println(F("MPU fail!"));
65     myFile.println(status);
66     while (1) {}
67 }
68 myFile.println(F("Senzor MPU-9250 pripraven"));
69 Serial.println(F("Senzor MPU-9250 pripraven"));

```

Další možnosti knihovny a její celkový popis lze najít zde [42].

Pro senzor BME280 jsem využil knihovnu BMx280I2C.h, která umí pracovat se senzory BME280 a BMP280 [43]. Senzor BME280 má oproti BMP280 navíc čidlo pro měření vlhkosti, ale to zde nevyužívám. Knihovna podporuje 64bitovou kalkulaci tlaku a má poměrně jednoduchý způsob pro získání dat a nastavení. Spuštění a kontrola navázání spojení je provedeno následovně.

```

72 if (!bmx280.begin()) {
73     Serial.println(F("BME280 fail!"));
74     myFile.println(F("BME280 fail!"));
75     while (1) {}
76 }
77 if (bmx280.isBME280()) {
78     Serial.println(F("Senzor BME280 pripraven"));
79     myFile.println(F("Senzor BME280 pripraven"));
80 }

```

Tento senzor jsem po zhodnocení nastavil na 4násobné převzorkování tlaku, tak aby docházelo k určitému vyhlazení, ale zároveň aby data reagovala poměrně rychle na změny. Převzorkování teplotní čidla jsem podle rad v dokumentaci zanechal na 1násobku.

```

81 //Překonfigurování BME280
82 bmx280.resetToDefaults();
83 //Převzorkování senzoru tlaku
84 bmx280.writeOversamplingPressure(BMx280MI::OSRS_P_x04);
85 //Převzorkování senzoru teploty
86 bmx280.writeOversamplingTemperature(BMx280MI::OSRS_T_x01);

```

Při práci se servomotorem, který jsem měl k dispozici, jsem si všiml, že se neustále snaží vyrovnávat poslední zadanou pozici (lehce se klepal). Tím dochází ke zbytečnému odběru proudu, což jsem zjistil pomocí ampérmetru. Provedl jsem proto softwarové odpojení servomotoru od řízení.

```

89 myservo.detach();

```

Tím by se teoreticky mohla volně otočit osa servomotoru (to nechceme), pokud by na ni zapůsobila dostatečně velká síla, která by překonala sílu převodů uvnitř servomotoru, ale dle pokusů jsem došel k závěru, že k tomu nedojde.

Posledním krokem v nastavovací části, jsem provedl zápis sdělení startu měření a zároveň pro jistotu provedl kontrolu zápisu dat na SD kartu.

```

92 Serial.println(F("Start měření"));
93 myFile.println(F("Start měření"));
94 myFile.flush();
95 myFile.close();
96 myFile = SD.open("test.txt", O_CREAT | O_WRITE | O_APPEND);

```

Opět jsem zde vycházel z [41], kde je tento postup zmíněn.

### 7.1.3 Část datového sběru

Celá tato část je součástí funkce void loop(), která se provádí, jak už název napovídá v “nekonečné” smyčce, jenž je prováděna po provedení funkce void setup(). Konec této smyčky je dán odepnutím přívodu napětí.

Základním pilířem pro práci s daty ze senzorů je zjistit čas, jenž od sebe dělí jednotlivá měření. Tento čas jsem pojmenoval v kódu d\_t neboli  $\Delta t$  (delta t).

```

103 timer = millis();
104 d_t = timer - timer_helper;
105 timer_helper = timer;

```

Získávám jej za pomoci funkce millis(), která vrací počet milisekund, které uplynuly od doby, kdy deska Arduino spustila aktuální program. Danou hodnotu ukládám do proměnné timer. Následně pak dochází k výpočtu d\_t, který však v prvním průchodu cyklu, nebude vracet korektní hodnotu. Neboť mi d\_t vyjadřuje čas mezi dvěma po sobě jdoucími měřeními je potřeba získat druhé měření, které nastane v následujícím cyklu. Proměnná timer\_helper tak uloží čas minulého měření. Rozdíl hodnot timer

a timer\_helper (timer > timer\_helper) tak dá za vznik d\_t, který bude velmi užitečný pro analýzu letu.

Následuje volání funkcí, které provádí načtení aktuálních hodnot senzorů z jejich interních registrů. Pro modul MPU-9250, se dle knihovny MPU9250.h využívá funkce readSensor().

```
108 | IMU.readSensor();
```

Pro modul BME280, se dle ukázky z knihovny BMx280I2C.h využívá následující postup získání hodnot.

```
111 | if (!bmx280.measure()) {
112 |     return;
113 | }
115 | do {
116 |     delay(10);
117 | } while (!bmx280.hasValue());
```

Hlavní díl této části spočívá ve výpisu, či zápisu naměřených hodnot.

```
120 | if (buff < 100) {
121 |     Serial.print("\t"); Serial.print(IMU.getAccelX_mss(), 2);
122 |     Serial.print("\t"); Serial.print(IMU.getAccelY_mss(), 2);
123 |     Serial.print("\t"); Serial.print(IMU.getAccelZ_mss(), 2);
124 |     Serial.print("\t"); Serial.print(IMU.getGyroX_rads(), 2);
125 |     Serial.print("\t"); Serial.print(IMU.getGyroY_rads(), 2);
126 |     Serial.print("\t"); Serial.print(IMU.getGyroZ_rads(), 2);
127 |     Serial.print("\t"); Serial.print(bmx280.getPressure64());
128 |     Serial.print("\t"); Serial.print(bmx280.getTemperature());
129 |     Serial.print("\t"); Serial.println(d_t);
130 |     myFile.print("\t"); myFile.print(IMU.getAccelX_mss(), 2);
131 |     myFile.print("\t"); myFile.print(IMU.getAccelY_mss(), 2);
132 |     myFile.print("\t"); myFile.print(IMU.getAccelZ_mss(), 2);
133 |     myFile.print("\t"); myFile.print(IMU.getGyroX_rads(), 2);
134 |     myFile.print("\t"); myFile.print(IMU.getGyroY_rads(), 2);
135 |     myFile.print("\t"); myFile.print(IMU.getGyroZ_rads(), 2);
136 |     myFile.print("\t"); myFile.print(bmx280.getPressure64());
137 |     myFile.print("\t"); myFile.print(bmx280.getTemperature());
138 |     myFile.print("\t"); myFile.println(d_t);
139 |     buff = buff + 1;
140 | }
```

Celý tento zápis je obsahem podmínky. Její kontrolovanou proměnnou je buff, který slouží jako inkrementační počítadlo. Myšlenkou je provést každých 100 cyklů kontrolu pro zajištění uložení dat. Jakmile je buff roven hodnotě 100 je provedena else část podmínky.

```

141 else {
142     //Vyčištění vyrovnávací paměti, po odeslání předchozích znaků
143     myFile.flush();
144     myFile.close();
145     buff = 0;
146     //Opětovné spuštění zápisu do SD karty
147     myFile = SD.open("test.txt", O_CREAT | O_WRITE | O_APPEND);
148 }

```

Posledním dílem, který zakončuje smyčku a tuto část je komplexní podmínka. Jejím záměrem je, aby se rozeznalo, kdy hlaviče rakety směřuje vzhůru, či k zemi a v závislosti na tom se prováděly jednotlivé podmínky.

```

151 //Hlavice směřující dolů
152 if (IMU.getAccelZ_mss() > 0) {
153     if (servo_stop) {
154         servo_help = 0;
155         myservo.attach(9);
156         myservo.write(160);
157         servo_stop = false;
158     }
159     else {
160         digitalWrite(3, LOW);
161         if (servo_help < 100) {
162             servo_help = servo_help + 1;
163         } else {
164             myservo.detach();
165         }
166     }
167 }
168 //Hlavice směřující nahoru
169 else {
170     if (servo_stop) {
171         digitalWrite(3, HIGH);
172         if (servo_help < 100) {
173             servo_help = servo_help + 1;
174         }
175     } else {
176         myservo.detach();
177     }
178 }
179 else {
180     servo_help = 0;
181     myservo.attach(9);
182     myservo.write(0);
183     servo_stop = true;
184 }
185 }

```

Osa z je v rámci hodnot z akcelerometru rovnoběžná s trupem a hlavicí. Je tedy hlavním indikátorem, jak zjistit, kterým směrem raketa směřuje. Vektor tíhové síly míří směrem

dolů, proto jsou jeho hodnoty záporné, když je raketa v klidu a míří vzhůru. V kódu to lze pozorovat přechodem z řádku č. 152 na č. 169 (1.). Zde dojde k rozsvícení LED pro viditelnou kontrolu, že dochází ke sběru dat. Při přetočení rakety přejdou hodnoty osy z akcelerometru do kladných hodnot a dojde k přechodu na podmínku na řádku č. 152 (2.). Zde dojde k zapnutí servomotoru a jeho otočení, které tím vypustí padák. Rychle se přejde na cyklus podmínky na řádcích č. 159–166 (3.), kde dojde po 100 cyklech k opětovnému odpojení servomotoru (zhruba tolik cyklů potřebuje na otočení). Jakmile dojde k opětovnému přetočení rakety vzhůru, tak se přejde na řádek č. 179 (4.), kde se servo přetočí na základní pozici a skočí se zpět na počáteční řádek č. 169 (5.), kterým jsme začínali. Tímto je tak vytvořen bezpečnostní cyklus, který slouží jak pro testování servomotoru a akcelerometru, před startem, tak pro případné letové problémy.

## 8 Další možnosti (rozšíření)

Za zmínku stojí obecné doporučení si vše pravidelně zálohovat, neboť se mi při dokončování kódu porouchal harddisk a vše jsem musel přepisovat. Ve výsledku to však, ale bylo poměrně dobré poučení a zároveň jsem si kód více utřídil.

Kromě tohoto kódu existovala ještě dvě jiná provedení, ve kterých jsem se snažil pracovat se senzory, přes vlastní funkce. Když jsem však porovnal výkonnost (latenci) funkcí napsaných mnou a dostupnými bylo zřejmě, že bude lepší použít právě ty.

Přemýšlel jsem i o využití Bluetooth modulu, kterým bych mohl zaslat přes mobilní aplikaci pokyn ke startu sběru dat na dálku. Přišlo mi to nakonec vcelku nadbytečné a nevím, jestli by to bylo vůbec možné vzhledem k již značně zaplněné paměti.

Zjistil jsem, že lze psát programy do Arduina v Pythonu, což bych rád někdy v budoucnu vyzkoušel.

Raketa by šla udělat samozřejmě mnohem větší (objemově), ale hlavně by šel její trup vytvořit od píky z karbonových vláken a dosáhnout tak mnohem vyšších vnitřních tlaků.

Trup by šel spojovat i přes hrdla lahví, pomocí spojek vytvořených z 3D tisku, na internetu je lze nalézt pod názvem „Tornado tube“. Nebo namísto komplikovaného vrtání do šroubů by bylo možné provést propojení mezi lahvemi pomocí bezdušových ventilků, ze kterých by byly vyjmuty vnitřky.

Pomocí 3D tisku lze zároveň vytvořit i speciálně upravené trysky, které poskytují lepší efektivitu výtlaku vody, či kompletní odpalovací mechanismy, nebo speciálně natvarované hlavice poskytující co nejnižší koeficient odporu prostředí.

## 9 Odpal a měření

Odpal byl proveden 17. 6. 2021 kolem čtvrté hodiny odpolední. Byl velmi horký den o teplotě vzduchu 30 °C bez mraků na obloze s velmi lehkým větrem. GPS lokace odpalu: 50.0405381N, 15.7625453E. Odpal se podařil na třetí pokus. Při prvním začala voda vytékat z trysky při tlaku 350 kPa. Při druhém pokusu jsem navlékl na hrdlo druhý o-kroužek, avšak voda začala vytékat již při tlaku 150 kPa. Odebral jsem tedy o-kroužek a pomocí sikových kleští dotáhl velmi pevně trysku. Při třetím pokusu a tlaku zhruba 400 kPa jsem provedl úspěšný odpal.

Raketa byla naplněná zhruba 2,4 litry vody. Malá část vody se dostala do střední úrovně rakety. Tato malá část vody zřejmě způsobila velmi krátce po startu krátký okamžik, kdy raketa nevytvářela tah. Tím se raketa nedostatečně urychlovala a po startu odbočila částečně do strany na sever. Raketa úspěšně vypustila padák přesně v momentu dosažení maximální vzletové výšky. Raketa dopadla horizontálně do vysoké trávy, která zaručila hladké přistání. Raketa dopadla zhruba 20 metrů od místa odpalu.

Byla zde možnost druhého odpalu s větším tlakem a menším množstvím vody. Avšak při tlaku 450 kPa došlo k odtržení tlakovací hadice zespona odpaliště. Tuto závadu nebylo možno opravit na místě, a tak se musel případný další odpal odložit.

Po prozkoumání dat z paměťové karty, byl zjištěn jistý problém. Sběr dat z úspěšného letu byl zaznamenán pouze částečně. Na samotnou analýzu dat to však postačuje.

### 9.1 Analýza dat letu

Pro analýzu letu bylo nejprve nutné najít v datech získaných na SD kartě blízký moment odpalu rakety. Data jsou zapsána do textového souboru *TEST.txt*, kde před každým startem měření je několik řádků textu potvrzující funkčnost modulů, čímž se dá rychle orientovat, kde je začátek a konec jednoho měření. Pro práci v textovém souboru jsem využil prostředí Notepad++. Dle Obrázek č. 57 lze vidět část surových dat, krátce před a po odpalu. Osobně jsem vybral 10 řádků dat před startem a všechny data po odpalu. Celkově jsem tak získal 199 řádků dat na zpracování.

38677	-0.24	-0.23	-9.60	0.01	0.00	0.00	99165.96	48.59	17
38678	-0.24	-0.40	-9.54	0.01	0.00	0.00	99165.35	48.59	11
38679	-0.24	-0.29	-9.40	0.01	0.00	-0.01	99165.35	48.59	14
38680	-0.24	-0.29	-9.40	0.01	0.00	-0.01	99168.64	48.58	14
38681	-0.32	-0.09	-10.08	0.01	0.00	-0.01	99168.86	48.59	14
38682	-0.44	-1.95	-19.46	0.03	0.01	-0.04	99169.00	48.60	14
38683	-0.44	-1.95	-19.46	0.03	0.01	-0.04	99169.92	48.61	14
38684	-0.63	-0.92	-25.09	-0.01	-0.00	-0.13	99169.00	48.60	15
38685	-0.26	-0.51	-27.76	-0.01	-0.01	-0.20	99167.68	48.60	15

Obrázek č. 57 – Ukázka surových dat

Dle kódu si lze uvědomit, že právě zvýrazněný sloupec na Obrázek č. 57 popisuje zrychlení v ose z. Prudké snížení této hodnoty jasně znamená, že raketa zrychluje směrem vzhůru.

Pro efektivní práci s daty je následně výhodné přenést letová data do tabulkového procesoru. Využil Google Sheets, které bezplatně nabízí příjemné a efektivní prostředí pro zpracování dat. Je vhodné si rozvrhnout data do několika listů, podle typu dat. Osobně jsem vytvořil listy pro data akcelerometrická, gyroskopická, tlaková a přidal list výhradně pro grafy a vedlejší výpočty. Celkově jsem tak pracoval na pěti listech.

akcelerometrická data			gyroskopická data			tlaková data	teplota	delta_t
-0.29	-1.26	-13.12	-0.12	-0.05	-0.56	99158.42	48.60	15
0.17	-1.04	-13.25	-0.15	-0.06	-0.62	99164.99	48.61	17
0.17	-1.04	-13.25	-0.15	-0.06	-0.62	99157.50	48.59	12
0.08	-0.90	-13.58	-0.16	-0.08	-0.64	99157.50	48.59	15
-0.05	-1.17	-13.38	-0.16	-0.09	-0.70	99155.49	48.59	14
-0.05	-1.17	-13.38	-0.16	-0.09	-0.70	99159.93	48.60	14
-0.11	-1.00	-13.44	-0.17	-0.09	-0.74	99157.11	48.60	16
-0.10	-0.68	-13.62	-0.18	-0.10	-0.79	99155.61	48.60	14
-0.29	-0.79	-13.98	-0.18	-0.10	-0.76	99158.42	48.60	15

Obrázek č. 58 – Grafické rozdělení dat

Teplotní data ze senzoru, bych bral jako velmi nepřesná, neboť se zcela jasně neblíží skutečné teplotě vzduchu kolem rakety, kterou jsou získal v čas odpalu z měřící stanice na internetu. Neboť je senzor uvnitř hlavičky bez otvorů pro odvětrání, lze si tak aspoň udělat představu, jak vysoká teplota byla uvnitř hlavičky.

### 9.1.1 Čas

Čas mezi jednotlivými naměřenými daty, který označuji jako delta\_t ( $\Delta t$ ), či  $d_t$  je zásadní údaj, který je potřebný pro všechny typy dat. Pomocí něho lze právě odvodit časový vývoj všech měření. Je tedy nezbytné pochopit princip, jak ho v rámci analýzy využít. Dle Obrázek č. 59 je naznačen postupný součet, jako způsob, jak lze dojít k výpočtu jedné z prvních důležitých informací letu. Tím je právě samotný čas  $t$ , který bude později využit jako osový údaj v grafech.

n	$d_t$ / ms	=	t / ms	/1000	t / s
1	12,00	+	12,00	→	0,012
2	10,00	+	22,00	→	0,022
3	12,00	+	34,00	→	0,034
4	12,00	→	46,00	→	0,046
5	14,00	→	60,00	→	0,060

Obrázek č. 59 – Ukázka postupného součtu

Je zásadní si uvědomovat jednotky ve kterých data získáváme. Údaj  $d_t$  je časový údaj v jednotkách ms, tudíž je vhodné po postupném součtu provést převedení i na sekundy vydělením tisícem.

Z celkových 199 řádků analyzovaných dat je tedy celkový čas 2,885 s. Delta\_t se tak průměrně rovná 14,5 ms (2 885 / 199).

### 9.1.2 Kalibrace akcelerometrických dat

V rámci akcelerometrických dat, lze dojít k mnohým zajímavým letovým údajům. Prvním z nich, který nám následně poslouží získat i ty ostatní je zrychlení. To již díky datům ze senzoru máme částečně získané, avšak je potřeba tyto data nejprve zkalibrovat.

Kalibraci provedeme tak, že si do listu pro vedlejší výpočty vložíme akcelerometrická data, naměřená před odpalem. Vhodné je použít alespoň 100 řádků dat, které jsou poměrně neměnná (před odpalem v rámci tlakování, byly raketě způsobeny menší otřesy, který jsou v datech vidět, chceme tedy úsek, kde k těmto otřesům nedocházelo – raketa byla v “klidu”). Pro jednotlivé sloupce pak vytvoříme průměrnou hodnotu, která se stane kalibrační odchylkou, kterou později využijeme.

Je důležité si uvědomit, že získaná akcelerometrická data jsou vektory ve směru navzájem kolmých os x, y, z. Vytvoříme tedy nový sloupec, ve kterém bude jejich součet. Vznikne nám velikost vektoru tíhového zrychlení, z kterého zároveň získáme jeho průměrnou před odpalovou hodnotu.

116	-0,22	-0,24	-9,53	-9,99
117	-0,23	-0,22	-9,56	-10,01
118	-0,23	-0,23	-9,55	-10,01
119	-0,23	-0,23	-9,55	-10,01
120	-0,25	-0,21	-9,56	-10,02
121	-0,23	-0,22	-9,56	-10,01
<b>Průměry</b>	<b>-0,21</b>	<b>-0,29</b>	<b>-9,55</b>	<b>-10,06</b>
	<i>g<sub>x</sub> / m/s<sup>2</sup></i>	<i>g<sub>y</sub> / m/s<sup>2</sup></i>	<i>g<sub>z</sub> / m/s<sup>2</sup></i>	<i>g / m/s<sup>2</sup></i>

Obrázek č. 60 – Kalibrační odchylky tíhového zrychlení (nepřesné)

Lze si všimnout, že podle Obrázek č. 60, není výslednice tíhového zrychlení úplně přesná. Měla by se pohybovat blíže kolem hodnoty 9,81 m/s<sup>2</sup>. Je tedy potřeba provést výpočet kalibračního čísla. Zde nám stačí jednoduchá trojčlenka. Potřebujeme zjistit kolik procent je 9,81 z 10,06.

$$\text{kalibrační číslo} = \frac{9,81 \cdot 100}{10,06} \doteq 97,52 \% = 0,9752 \quad (2)$$

Kalibračním číslem nyní stačí jednotlivé osy tíhového zrychlení vynásobit a jejich nový součet bude přesnější viz Obrázek č. 61.



116	-0,220	-0,240	-9,530	-0,215	-0,234	-9,294	-9,742
117	-0,230	-0,220	-9,560	-0,224	-0,215	-9,323	-9,762
118	-0,230	-0,230	-9,550	-0,224	-0,224	-9,313	-9,762
119	-0,230	-0,230	-9,550	-0,224	-0,224	-9,313	-9,762
120	-0,250	-0,210	-9,560	-0,244	-0,205	-9,323	-9,772
121	-0,230	-0,220	-9,560	-0,224	-0,215	-9,323	-9,762
<b>Průměry</b>	<b>-0,215</b>	<b>-0,294</b>	<b>-9,555</b>	<b>-0,209</b>	<b>-0,287</b>	<b>-9,318</b>	<b>-9,814</b>
	<i>g<sub>x</sub> / m/s<sup>2</sup></i>	<i>g<sub>y</sub> / m/s<sup>2</sup></i>	<i>g<sub>z</sub> / m/s<sup>2</sup></i>	<i>g<sub>x_k</sub> / m/s<sup>2</sup></i>	<i>g<sub>y_k</sub> / m/s<sup>2</sup></i>	<i>g<sub>z_k</sub> / m/s<sup>2</sup></i>	<i>g / m/s<sup>2</sup></i>

Obrázek č. 61 – Kalibrační odchylky tíhového zrychlení (zpřesněné)

### 9.1.3 Zrychlení

Na listu s akcelerometrickými údaji letu nyní vytvoříme 3 nové sloupce ve kterých zkalibrujeme hodnoty zrychlení stejným způsobem jako na Obrázek č. 61. Hodnoty se budou lišit jen lehce, ale budou o to přesnější. S těmito akcelerometrickými údaji nyní budeme dále pracovat.

kalibrační číslo=	0,9752					
n	<i>ac<sub>x</sub> / m/s<sup>2</sup></i>	<i>ac<sub>y</sub> / m/s<sup>2</sup></i>	<i>ac<sub>z</sub> / m/s<sup>2</sup></i>	<i>ac<sub>x_k</sub> / m/s<sup>2</sup></i>	<i>ac<sub>y_k</sub> / m/s<sup>2</sup></i>	<i>ac<sub>z_k</sub> / m/s<sup>2</sup></i>
1	-0,25	-0,22	-9,55	-0,24	-0,21	-9,31
2	-0,25	-0,22	-9,55	-0,24	-0,21	-9,31
3	-0,25	-0,23	-9,58	-0,24	-0,22	-9,34
4	-0,25	-0,23	-9,58	-0,24	-0,22	-9,34
5	-0,25	-0,23	-9,58	-0,24	-0,22	-9,34
6	-0,27	-0,22	-9,58	-0,26	-0,21	-9,34

Obrázek č. 62 – Kalibrovaná akcelerometrická data

Sečtením jednotlivých kalibrovaných zrychlení v osách x, y, z bychom měli dostat v případě klidového (nehybného) stavu nulové hodnoty. Důsledek tíhové síly na senzor akcelerometru musíme vykompenzovat tím, že odečteme průměrné tíhového zrychlení od jednotlivých naměřených hodnot a zároveň převrátíme znaménkovou logiku viz Obrázek č. 63. Tak aby zrychlení směrem vzhůru (dopředu) bylo bráno jako kladná hodnota.

	E	F	G	H
1	velikost vektoru tíhového zrychlení před startem			
2		-9,814		
3				
4				
5	<i>ac<sub>x_k</sub> / m/s<sup>2</sup></i>	<i>ac<sub>y_k</sub> / m/s<sup>2</sup></i>	<i>ac<sub>z_k</sub> / m/s<sup>2</sup></i>	<i>a / m/s<sup>2</sup></i>
6	-0,24	-0,21	-9,31	$=((E6+F6+G6)-F52)*(-1)$
7	-0,24	-0,21	-9,31	-0,04
8	-0,24	-0,22	-9,34	0,00
9	-0,24	-0,22	-9,34	0,00
10	-0,24	-0,22	-9,34	0,00

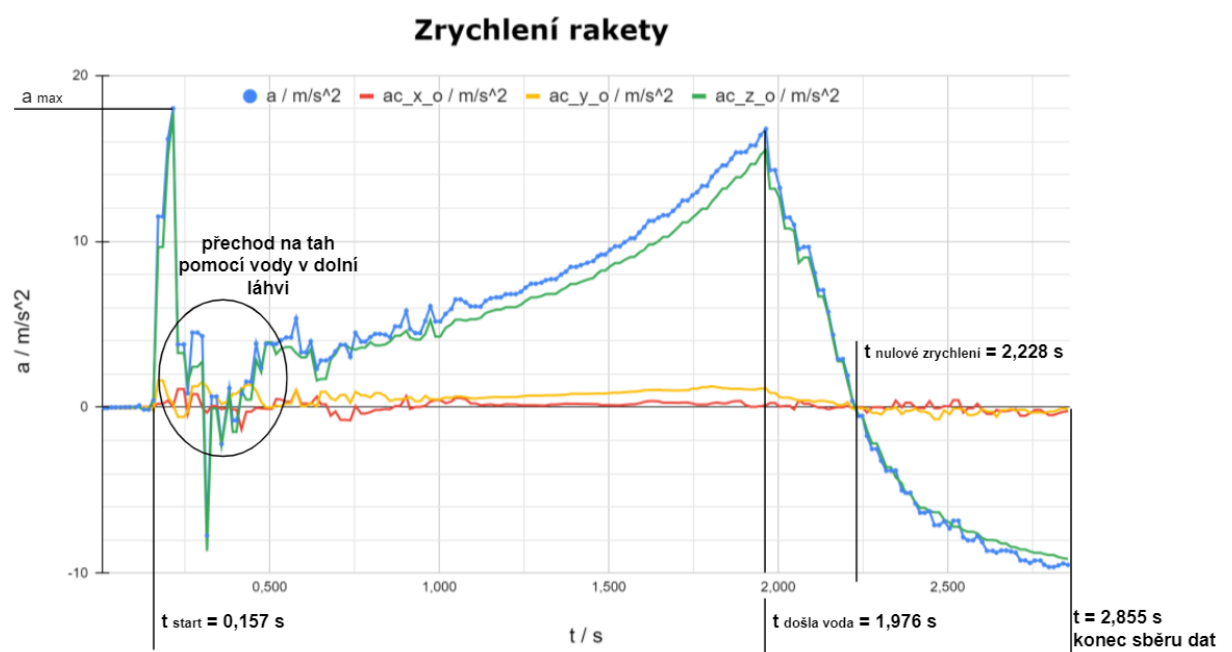
Obrázek č. 63 – Velikost zrychlení

Užitečný údaj je také zrychlení jednotlivých os “očištěných” od hodnot tíhového zrychlení v klidovém stavu s převrácenou znaménkovou logikou.

	E	F	G	H	I	J	K	
1	velikost vektoru tíhového zrychlení před startem				Průměry před startem	x	y	z
2		-9,814				-0,209	-0,287	-9,318
3								
4								
5	ac_x_k / m/s^2	ac_y_k / m/s^2	ac_z_k / m/s^2	a / m/s^2	ac_x_o / m/s^2	ac_y_o / m/s^2	ac_z_o / m/s^2	
6	-0,24	-0,21	-9,31	-0,04	=(E6-\$I\$2)*(-1)	-0,07	0,00	
7	-0,24	-0,21	-9,31	-0,04	0,03	-0,07	0,00	
8	-0,24	-0,22	-9,34	0,00	0,03	-0,06	0,02	
9	-0,24	-0,22	-9,34	0,00	0,03	-0,06	0,02	
10	-0,24	-0,22	-9,34	0,00	0,03	-0,06	0,02	
11	-0,26	-0,21	-9,34	0,01	0,05	-0,07	0,02	

Obrázek č. 64 – Zrychlení jednotlivých os (očistěných)

Nyní lze využít tyto data, k získání prvního zajímavého Graf č. 2, týkajícího se čistě zrychlení.



Graf č. 2 – Průběh zrychlení rakety

Raketa byla při startu naplněna z malé části do druhé láhve. Vzhledem k úzkému průchodu mezi lahvemi, tak v momentu vyprázdnění nastala krátká chvíle bez tahu. To způsobilo časový úsek, kdy raketa zpomalovala. Naštěstí nedošlo v ose z k dosažení hodnoty ke splnění podmínky otevření padáku, při startu. Menší zrychlení v osách x a y přičítám z části rotaci rakety, která vytvořila na senzor akcelerometru odstředivou sílu. Optimalizování by mohlo být provedeno připevněním senzoru akcelerometru co nejbližší do osy otáčení, což zde nebylo úplně docíleno.

#### 9.1.4 Rychlost

Výpočet rychlosti rakety je nyní velmi snadný. Ze získaného zrychlení a údajů  $d_t$  k němu lze dojít podobně jako na Obrázek č. 59, kde se postupně sčítali časové dílky v jeden celek.

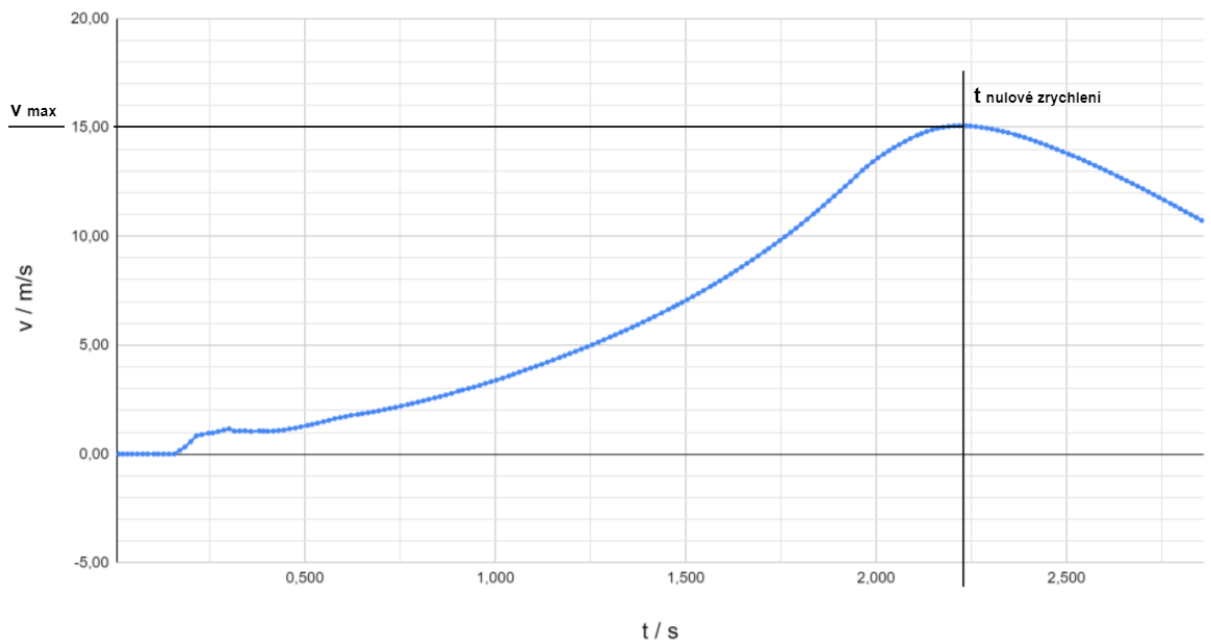
Nejprve je však nutné vypočítat dílčí rychlosti. Ty vychází z předpokladu, že se raketa pohybovala s určitým zrychlením po určitou dobu. Postupný součet těchto dílčích rychlostí nám určí rychlost v daný moment viz Obrázek č. 65.

$d_v = a \cdot (d_t / 1000)$	
$d_v / \text{m/s}$	$v / \text{m/s}$
$=H6 \cdot (L6 / 1000)$	-0,00051
-0,00042	-0,00093
-0,00004	-0,00098
-0,00004	-0,00102
-0,00005	-0,00107
0,00009	-0,00098

Obrázek č. 65 – Výpočet dílčích rychlostí a jejich postupný součet

Takto získaná data lze opět jednoduše využít pro tvorbu názorného Graf č. 3.

### Rychlost rakety



Graf č. 3 – Průběh rychlosti rakety

Krásně si lze všimnout, že maximální rychlost byla docílena v momentě, kdy zrychlení bylo nulové.

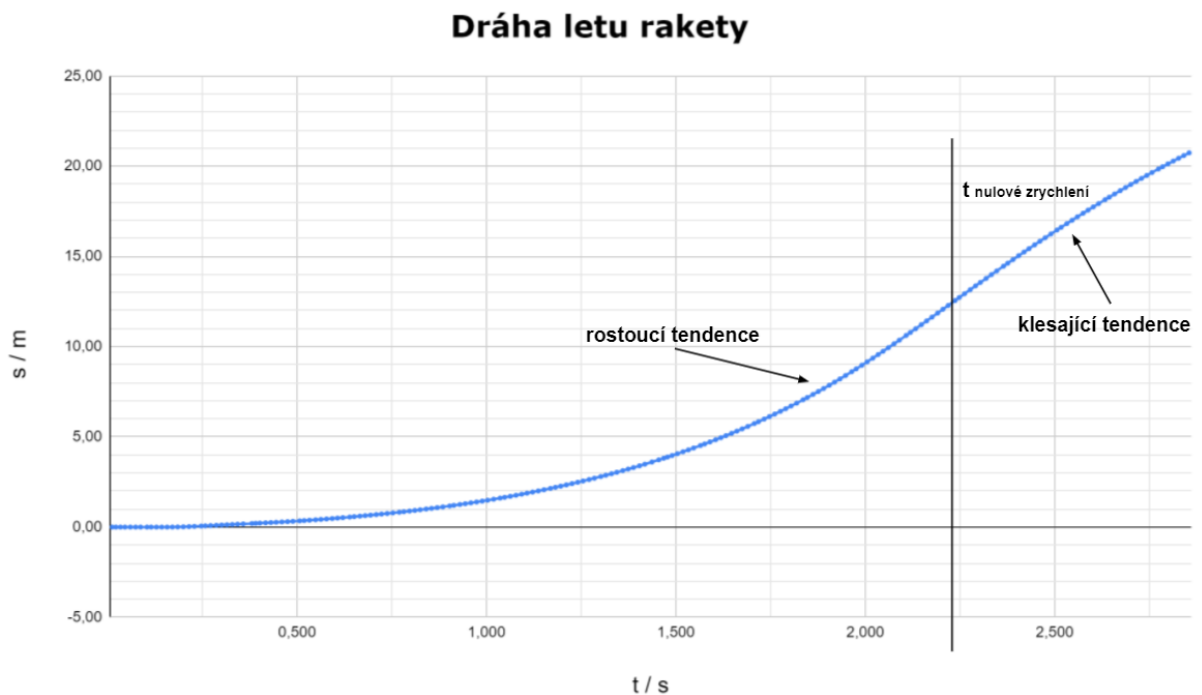
### 9.1.5 Dráha

Naprostu obdobným způsobem jako u rychlosti, lze získat vývoj dráhy letu rakety postupným součtem dílčích drah viz Obrázek č. 66.

$d_s = v \cdot (d_t / 1000)$	
$d_s / m$	$s / m$
$=P6 \cdot (L6 / 1000)$	-0,0000061
-0,0000093	-0,0000155
-0,0000117	-0,0000272
-0,0000122	-0,0000394
-0,0000149	-0,0000544
-0,0000137	-0,0000681

Obrázek č. 66 – Výpočet dílčích drah a jejich postupný součet

Vznikne tak graf dráhy letu rakety.



Graf č. 4 – Průběh dráhy letu rakety

Dráha opisuje trajektorii letu, tudíž nelze její vývoj brát jako předpoklad pro dosaženou vzletovou výšku, neboť již z pozorování došlo krátce po startu k změně směru letu vůči kolmé rovině od startovní polohy. Lze však na ní vidět jak tendenci rostoucí, tak klesající, což odpovídá postupnému zvyšování zrychlení a následnému zpomalování. Dost pravděpodobně by zde bylo další opakování rostoucí a klesající tendence, kdyby záznam pokračoval i při postupném volném pádu (nabírání zrychlení) a otevření padáku (zpomalování do určité rychlosti).

### 9.1.6 Gyroskopická data – rotace rakety

Ideálně by raketa neměla v rámci letu v podélné ose (osa z) rotovat. Snižuje to její dolet, neboť dochází k většímu tření a turbulencím se vzduchem. Zároveň začnou vznikat odstředivé síly, které mohou mít dopad na funkčnost rakety a bezpečí jejího nákladu.

Získaná gyroskopická data jsou fyzikální veličinou úhlové rychlosti, jejíž jednotkou jsou radiány za sekundu. Pro lepší názornost je vhodné převedení na jednotky stupňů za sekundu. To se provede dle Obrázek č. 67 následovně.

	B	C
1	gy_x / rad/s	gy_x / °/s
2	0,000	=B2*180/PI()
3	0,000	0,000
4	0,010	0,573
5	0,010	0,573

Obrázek č. 67 – Převod rad/s na °/s

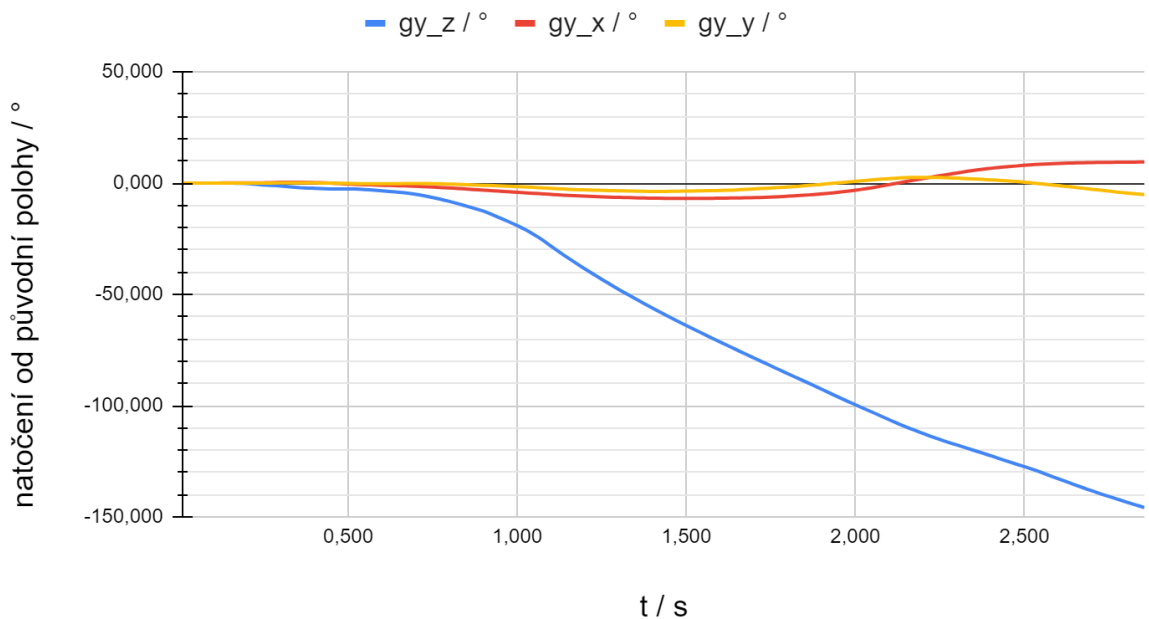
Po převedení jsou následně provedeny stejné operace, jako při výpočtu rychlosti a dráhy. To znamená, že v rámci jednoho sloupce budou vypočítány dílčí natočení a v druhém se dílčí hodnoty postupně sečtou.

	C	D	E	N
1	gy_x / °/s	d_gy_x / °	gy_x / °	dt/ms
2	0,000	=C2*(N2/1000)	0,000	12,000
3	0,000	0,000	0,000	10,000
4	0,573	0,007	0,007	12,000
5	0,573	0,007	0,014	12,000

Obrázek č. 68 – Výpočet dílčího natočení a jeho postupný součet

Tyto operace se provedou stejně pro každou z os x, y, z. Následně se postupné součty natočení využijí do Graf č. 5 znázorňující průběh rotace rakety.

## Rotace rakety



Graf č. 5 – Průběh rotace rakety

Dle Graf č. 5, lze jasně vidět, že se raketa postupně otáčela podél osy z (podélně). Mezi první a druhou sekundou záznamu lze vidět 80° rotaci. Vzhledem k tomu, že jsem si přesnost gyroskopu zkušel samostatně před startem, lze tyto data brát jako velmi přesná. Kromě toho lze vidět rotaci i v osách x a y, což naznačuje zmiňované natočení rakety do strany při startu.

Zajímavým pohledem na rotaci rakety může být i provedení derivace dat z Graf č. 5. Tím lze získat grafický průběh rychlosti otáčení rakety. Derivace je matematická funkce, která se většinou vyučuje až v posledních ročnících středních škol s maturitou a nebývá většinou moc oblíbená. Je tedy tato část určena spíše šikovnějším žákům.

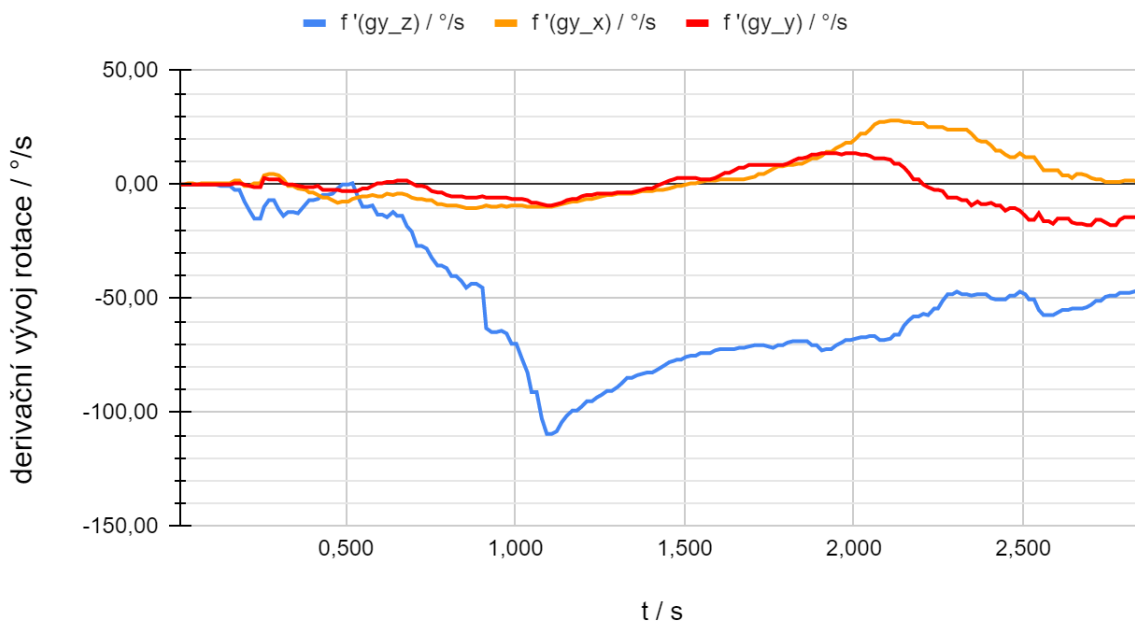
Derivaci dat z grafu rotace rakety získáme jako rozdíl rotace pro daný okamžik vydělený časovým úsekem  $d_t$  neboli názorně pro osu z za pomoci výstřižku viz Obrázek č. 69.

$f_x = \frac{(M3 - M2) / (N3 / 1000)}$

	M	N	S
1	gy_z / °	dt/ms	f'(gy_z) / °/s
2	0,000	12,000	0,00
3	0,000	10,000	$=(M3 - M2) / (N3 / 1000)$
4	0,000	12,000	0,00

Obrázek č. 69 – Derivace rotace rakety osy z pro získání rychlosti rotace

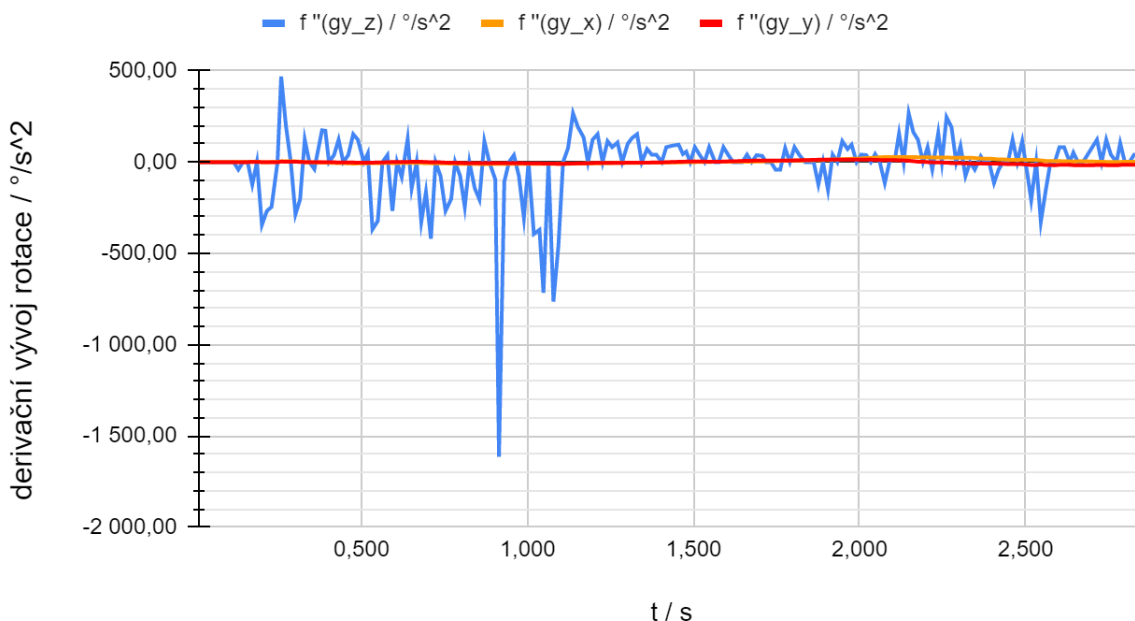
## Derivace rotace rakety => rychlost rotace



Graf č. 6 – Průběh rychlosti rotace rakety

Graf č. 6 nabízí zajímavé informace, jako je například u osy z dosažení maximální rychlosti rotace ( $109,4 °/s$ ), kdy následně docházelo postupně k jejímu zpomalování. Derivací takto derivovaných dat stejným způsobem jako v předešlém případě získáme časový průběh zrychlení rotace viz Graf č. 7.

## Druhá derivace rotace rakety => zrychlení rotace



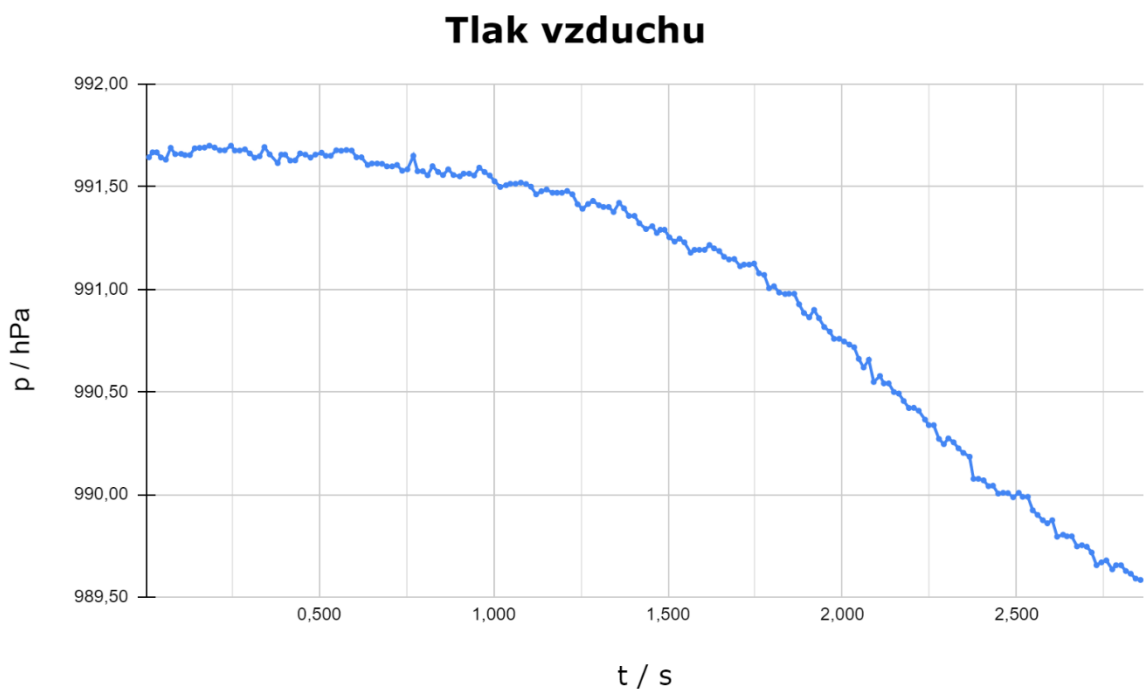
Graf č. 7 – Průběh zrychlení rotace rakety

Graf zrychlení rotace rakety není na první pohled moc zajímavý, neboť jsou hodnoty průběhu osy z mnohonásobně vyšší než hodnoty zbylých os x a y. Dle hodnot lze vidět častou oscilaci, která často zachází do nulových hodnot. To je zřejmě způsobeno momenty, kdy jednak nedocházelo v rámci naměřených hodnot rychlosti rotace ke změnám a následně se razantně hodnoty změnili. V rámci grafu zrychlení rotace rakety mě napadla myšlenka získat vývoj rotace u předmětů, kde získání rotace je požadováno, jako například u projektilů (šíp, kulka). U nich by tento graf měl zajímavější použití.

### 9.1.7 Barometrická data

Barometrická data slouží primárně k přesnějšímu určení vzletové výšky, pomocí hypsometrického vzorce (1) vypsáno v sekci 6.2.2. Pro výpočet vzorce je potřeba 3 neznámých. Jednu z nich (tlak okolního vzduchu) získáme právě skrze barometrická data. Ostatní dvě (tlak vzduchu přepočtený na hladinu moře + okolní teplota) je potřeba získat v době letu přes internet, nebo pomocí jiných měřících zařízení a zaznamenat si je pro pozdější výpočet.

Prvním grafem je samotný časový průběh změny tlaku, získaného pomocí barometrického senzoru viz Graf č. 8.



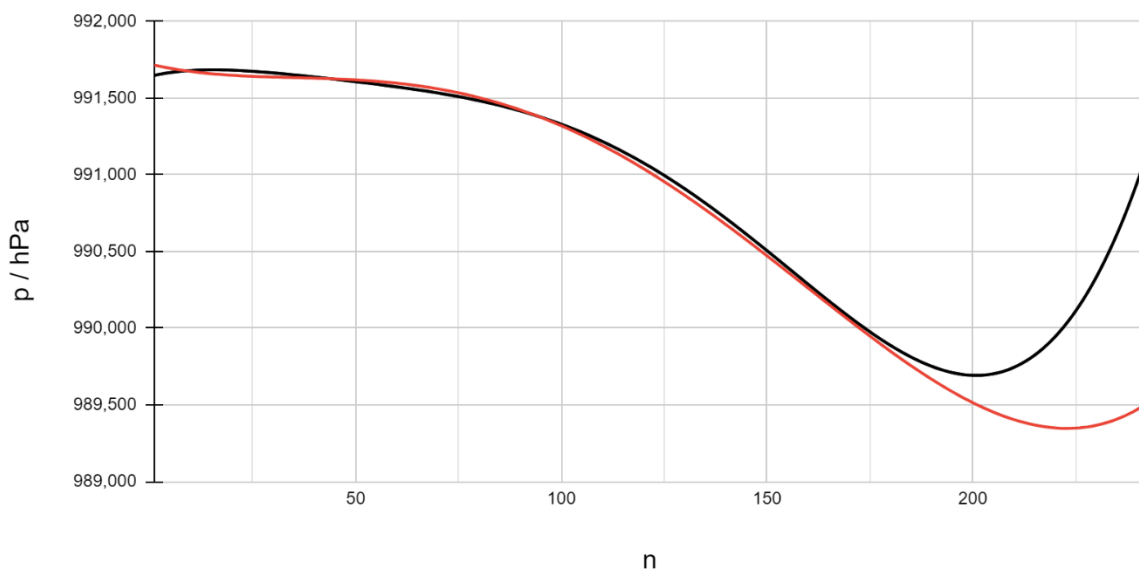
Graf č. 8 – Průběh změny tlaku vzduchu

Dochází na něm k poklesu tlaku vzduchu, což je způsobeno právě zvyšováním nadmořské výšky. Pro tento graf je vhodné použít polynomické spojnice trendu, které aproximují vývoj dat, jež nebyly získány a získat tak přibližnou předpověď budoucnosti.



## Aproximovaný tlak vzduchu

Černá křivka => polynomický trend 6. stupně; Červená křivka = polynomický trend 4. stupně



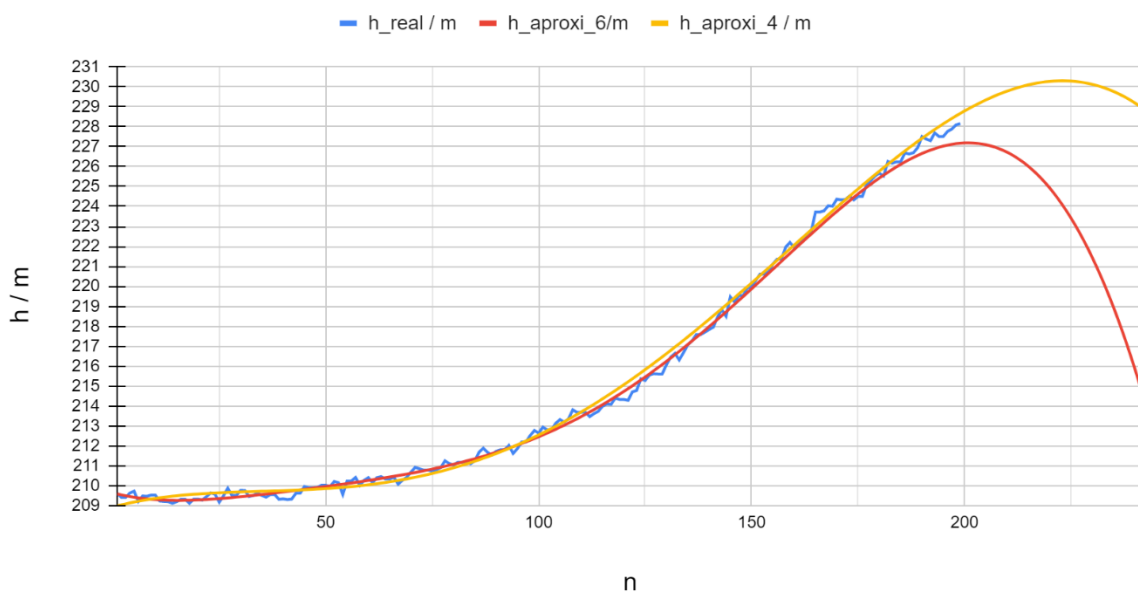
Graf č. 9 – Průběh změny tlaku vzduchu (aproximace)

Využití polynomických trendů 4. (volnější) a 6. (příkřejší) stupně je později využito k získání průměrné aproximované vzletové výšky.

Dle naměřených a aproximovaných hodnot tlaku vzduchu, lze tak vytvořit graf vývoje nadmořské výšky zanesením jejich hodnot do hypsometrického vzorce (1).

## Nadmořská výška aproximovaná a naměřená

Červená křivka => polynomický trend 6. stupně; Žlutá křivka = polynomický trend 4. stupně



Graf č. 10 – Průběh změny nadmořské výšky

Rozdíl mezi maximální dosaženou nadmořskou výškou a počáteční jednoduše určí vzletovou výšku. Ta vzhledem ke skutečnosti, že raketa neletěla kolmo vzhůru bude menší než dráha letu. Derivací grafu vývoje nadmořské výšky by dále bylo možné získat vertikální rychlost a zrychlení letu.

### 9.1.8 Tah

Tah je síla, kterou vyvíjí motory rakety k tomu, aby se urychlovala. K tomu, aby k urychlování došlo musí tah nejprve překonat sílu tíhovou. Tíhová síla vychází z hmotnosti rakety v daném čase a vzhledem k tomu, že se hmotnost rakety, během odpalu snižuje, neboť její obsah (zde voda) vytéká, je důležité se nejprve zabývat tím, jakou hmotnost má raketa v daném čase. K tomu je zaprvé potřeba znát časy, kdy došlo ke startu a kdy začala raketa zpomalovat (došla voda). Rozdíl těchto časů označuji jako dobu výtlačku vody.

$$t_{\text{výtlačk vody}} = t_{\text{došla voda}} - t_{\text{start}} = 1,976 - 0,157 = 1,819 \text{ s} \quad (3)$$

Zadruhé je potřeba znát hmotnost vody, která je v raketě při startu. Vydělením této hmotnosti časem výtlačku vody získáme průměrnou spotřebu vody.

$$\text{průměrná spotřeba vody} = \frac{m_{\text{vody při startu}}}{t_{\text{výtlačk vody}}} = \frac{2,4}{1,819} = 1,319 \frac{\text{kg}}{\text{s}} \quad (4)$$

Hmotnost rakety je součtem suché hmotnosti (rakety bez vody) a aktuální hmotností vody v ní.

$$m_{\text{rakety}} = m_{\text{suchá hmotnost}} + m_{\text{vody}} \quad (5)$$

Je tedy potřeba v momentě odpalu od tohoto součtu začít odečítat průměrnou spotřebu vody.

$$m_{\text{vody}} = m_{\text{vody před startem}} - \left( \sum_{t_{\text{start}}}^{t_{\text{po startu}}} \text{průměrná spotřeba vody} \cdot \Delta t \right) \quad (6)$$

V čase, kdy došlo k vytečení všechny vody bude hmotnost rakety již pouze rovna suché hmotnosti. Takto nyní lze vypočítat tíhovou sílu v daném čase.

$$\mathbf{F}_G = m_{\text{rakety}} \cdot \mathbf{g} \quad (7)$$

Druhý Newtonův pohybový zákon říká, že pokud na těleso působí síly, jejichž výslednice se nerovná nule, tak se těleso pohybuje se zrychlením, které je přímo úměrné velikosti výslednici působící síly a nepřímo úměrné hmotnosti tělesa [20] neboli:

$$\mathbf{a} = \frac{\mathbf{F}}{m} \rightarrow \mathbf{F} = m \cdot \mathbf{a} \quad (8)$$

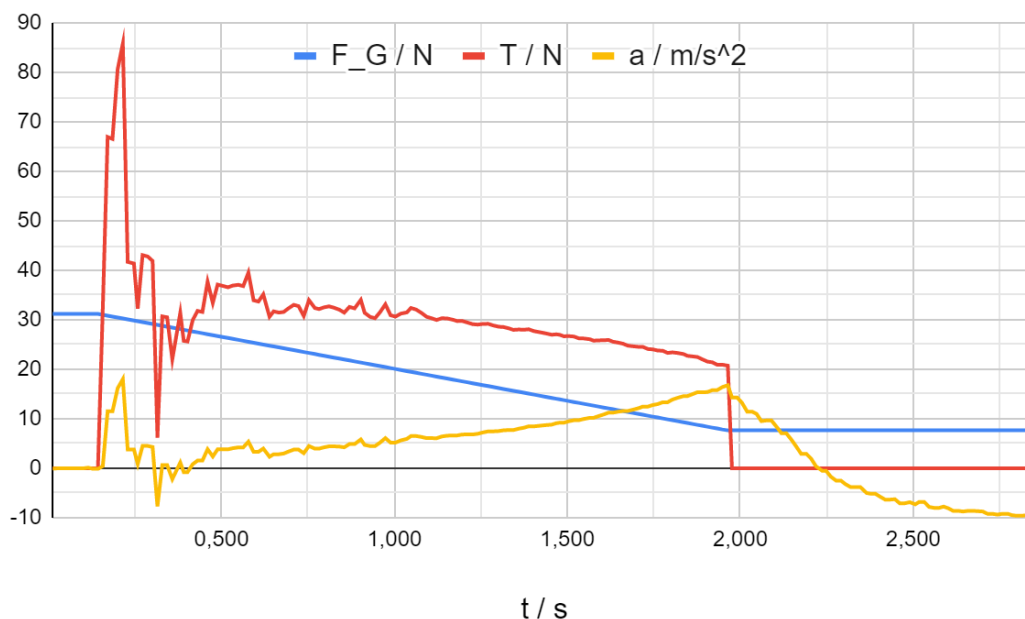
Vzhledem k tomu, že známe zrychlení rakety i její hmotnost (obojí v daném čase), lze stanovit sílu, která musela překonat sílu tíhovou a která způsobila urychlení rakety. Tuto sílu jsem pojmenoval jako sílu akcelerační.

$$\mathbf{F}_{\text{akcelerační}} = m_{\text{rakety}} \cdot \mathbf{a} \quad (9)$$

Součet akcelerační síly a tíhové síly v době výtlačku vody dá za vznik tahové síly (T).

$$\mathbf{T} = \mathbf{F}_{\text{akcelerační}} + \mathbf{F}_G \quad (10)$$

Takto lze nechat vytvořit zajímavý graf průběhu tíhové síly, tahové síly a zrychlení viz Graf č. 11. Síla odporu prostředí byla pro tento let zanedbatelná.



Graf č. 11 – Průběh tíhové síly, tahové síly a zrychlení

## 9.2 Parametry letu

### PARAMETRY LETU RAKETY TOM II, LET Č.1

<b>HMOTNOST VODY</b>	~ 2,4 kg
<b>STARTOVNÍ HMOTNOST</b>	~ 3,2 kg
<b>TLAK PŘI ODPALU</b>	~ 400 kPa
<b>OKOLNÍ TEPLOTA</b>	30 °C
<b>TLAK VZDUCHU PŘEPOČTENÝ NA HLADINU MOŘE</b>	1015,3 hPa
<b>PRŮMĚRNÉ <math>\Delta T</math></b>	$14,5 \cdot 10^{-3}$ s
<b>POČET ANALYZOVANÝCH ŘÁDKŮ DAT (N)</b>	199
<b>ODCHYLKA AKCELEROMETRU</b>	2,48 %
<b>MAXIMÁLNÍ AKCELERACE</b>	18 m/s <sup>2</sup>
<b>DOBA VYTLAČOVÁNÍ VODY</b>	~ 1,8 s
<b>MAXIMÁLNÍ RYCHLOST</b>	15,1 m/s ⇔ 54,3 km/h
<b>DRÁHA LETU</b>	20,74 m
<b>ROTACE V OSE Z</b>	145 °
<b>MAXIMÁLNÍ RYCHLOST ROTACE</b>	109 °/s
<b>MAXIMÁLNÍ NAMĚŘENÁ VZLETOVÁ VÝŠKA</b>	18,5 m
<b>PRŮMĚRNÁ APROXIMOVANÁ MAXIMÁLNÍ VZLETOVÁ VÝŠKA</b>	19,6 m
<b>PRŮMĚRNÁ SPOTŘEBA KAPALINY</b>	1,319 kg/s
<b>PRŮMĚRNÝ TAH</b>	30,9 N

Množství získaných parametrů předčilo moje původní očekávání, které jsem měl před začátkem práce na tvorbě rakety. Nejvíce mě potěšila spolehlivost dat z barometru, neboť jsem měl právě největší obavy z jeho případné nefunkčnosti během letu rakety. Dále jsem byl potěšen s poměrně malou rotací rakety, která měla ve výsledku tendenci i zpomalovat.

### 9.3 Cena

Přesná částka vložená do tvorby této práce lze těžko stanovit přesně, neboť do nich není započítaná cena nástrojů, které jsem již vlastnil (odlamovací nože, nůžky, vrtačka, pájecí pero). Ceny budou tedy spíše orientační.

#### CENÍK TVORBY VODNÍ RAKETY TOM II

<b>PET FLAŠKY (5X KOFOLA + 3X PEPSI)</b>	240,- Kč
<b>LEPÍCÍ PÁSKY</b>	100,- Kč
<b>DESKA PRO TVORBU OSAZENÍ</b>	100,- Kč
<b>SPOJNÝ MATERIÁL (ŠROUBY, PODLOŽKY, MATKY)</b>	100,- Kč
<b>ELEKTRONICKÉ KOMPONENTY (ARDUINO NANO, SENZORY, SD MODUL, NEPÁJIVÁ POLE, DRÁTKY, BATERIE, SERVO)</b>	~ 1000,- Kč
<b>TRYSKA (MOSAZNÁ RYCHLOSPOJKA + ČÁST ÚPRAV)</b>	200,- Kč
<b>OSTATNÍ (LANKO, PADÁK, ...)</b>	100,- Kč
<b>CELKEM</b>	<b>1840,- Kč</b>

Kromě rakety jsou nezbytnou součástí i další části, jako je odpaliště a k němu připojené komponenty potřebné k odpalu rakety (pumpička s manometrem a postřikovač). Cena připojených komponentů se může velmi různit, podle použitých modelů.

#### CENÍK TVORBY ODPALIŠTĚ PRO RAKETU TOM II

<b>ODHADOVANÁ CENA ODPALIŠTĚ PŘED ÚPRAVAMI</b>	700,- Kč
<b>HUSTILKA S MANOMETREM</b>	600,- Kč
<b>OSTŘIKOVAČ PRO PŘÍVOD VODY</b>	300,- Kč
<b>PODPŮRNÉ STOJNÉ TYČE</b>	120,- Kč
<b>PRŮCHODKY PRO STOJNÉ TYČE</b>	100,- Kč
<b>HADICE</b>	50,- Kč
<b>ROZDVOJKA + 2X RYCHLOSPOJKY</b>	250,- Kč
<b>CELKEM</b>	<b>2120,- Kč</b>

Odpaliště má svou výhodu v tom, že je s ním možné odpalovat opakovaně různé rakety. Tudíž by se dalo využívat, bez další investice. Je proto vhodné si u jeho konstrukce dát záležet při lakování, tak aby jednotlivé části, nebyly ovlivňovány vodou a Sluncem.

## 10 Využití ve výuce (metodická část)

V rámci Příloha č. 8-11 je vytvořen čtyřstránkový pracovní list, který uceleně shrnuje všechny potřebné předletové a letové informace. Potom, co žáci vytvoří vlastní rakety by tak do nich měli postupně doplňovat údaje. Učitel tedy může jednoduše rozdat žákům jednotlivě či do skupin pracovní listy, díky kterým lze procházet jednotlivé kroky obsahující název rakety, nákres rakety (či vložení její fotky, popř. nákres plánku), měření různých hodnot, základní výpočty, a mnoho dalších.

Pro Příloha č. 8 je možnost obrázek rakety zakreslit, či vlepít fotku. Zároveň si zde žáci mohou zakreslit, či vlepít plánek rakety a později tak vidět jak se jejich představa a výsledek shodují. Příloha č. 9 je poměrně přímočará, neboť se zde doplňují informace, dle zadaných instrukcí. Příloha č. 10 je rozdělena na 3 úrovně (levely), pro úroveň základní školy postačí vyplnění první a druhé (šikovnější žáci) úrovně. V případě střední školy se otevře úroveň třetí. Příloha č. 11 je místo pro vložení grafů. Žáci by si měli vybrat 3 grafy, dle výběru v odrážkách, které zpracují a pak je vytisknou a vlepí je. Šikovní žáci mají možnost vložit i bonusový graf tahu.

Pro potřeby více letů s jednou raketou, při kterých by se měly analyzovat, porovnávat a vyhodnocovat jednotlivé parametry letu, je vytvořena Příloha č. 12. Ideálně bych do poznámky zapsal, co bylo záměrem testů (např. zjistit vývoj vzletové výšky postupným zvětšováním tlaku, hledání průměrných hodnot letu při dodržení stejného tlaku a množství vody, ...)

Kromě fotek a obrázků vzniklých při konstrukci rakety TOM II byly pořízeny i videozáznamy, které se více zaměřují na předletovou přípravu a samotný let. Jsou uloženy na internetovém serveru YouTube [44][45]. Záznam [44] je z části doplněn mluveným komentářem a [45] je pořízený záběr letu rakety TOM II.

Perfektní záležitostí jsou též simulátory letu. Pomocí nich lze získat parametry letu rakety, bez skutečného odpalu. V rámci internetu lze najít řadu povedených tabulkových a appletových simulátorů odpalů vodních raket, kde po zadání předletových údajů lze zjistit řadu dat, které vycházejí z fyzikálních výpočtů. Simulátory se různí podle množství vstupních hodnot a zjistitelných údajů. Navíc je vždy potřeba brát jejich výsledky s rezervou, neboť ne vždy je fyzikální model výpočtu natolik kvalitní, aby odpovídal přesně skutečnosti. Musíme si uvědomit, že let raket je spjat s mnohými ne na první pohled zřejmými problémy a jejich přesné podchycení v matematickém modelu není vůbec snadné. Doporučuji například tento [46], nicméně v případě nefunkčnosti lze využít i jiné po zadání "water rocket simulator" do vyhledávače internetu.

## Závěr

Práce se zabývala problematikou získávání letových dat jednoduché (vodní) rakety pomocí zabudovaných senzorů (akcelerometr, gyroskop, barometr) a jejich následné analýzy. Zároveň měla posloužit jako výukový materiál, který by rozšířil možnosti učitelů informatiky a fyziky. Důležitým prvkem práce bylo samotné sestavení funkční vodní rakety TOM II, jenž v sobě zahrnuje již pokročilejší stavební prvky, které vycházely z nedostatků předchozího modelu TOM I.

V rámci cílů zmíněných v úvodu teoretické části, byl nejprve objasněn pojem jednoduché rakety, spolu se všemi podstatnými částmi, které ji tvoří. Následně byl vytvořen ucelený popis jednotlivých elektrických komponentů použitých pro tvorbu měřicího zařízení. Též bylo popsáno možné propojení tématu diplomové práce s předměty informatiky a fyziky v rámci nejaktuálnějších verzí dokumentů Rámcového vzdělávacího programu pro základní vzdělávání a gymnázia. Vzhledem k atypičnosti tématu práce byla velká část podkladů součástí internetových zdrojů, kde však bylo zájmem použít pouze těch, které se svým typickým obsahem nejvíce blíží využitému záměru, tak aby se zaručila relevantnost a kvalita citovaného obsahu. Celkově lze tak všechny cíle teoretické části brát jako úspěšně dosažené.

Praktická část práce byla značně podřízena samotné tvorbě rakety, úpravám odpaliště, sestavení měřicího zařízení a jeho naprogramování. Bez těchto čtyř prvků by nebylo možné dosáhnout hlavního cíle diplomové práce, čímž byla právě analýza letu jednoduché rakety. Pro dosažení hlavního cíle práce, zmíněného v úvodu praktické části, byla použita metoda sběru dat, jež byla získána v rámci úspěšného odpalu a letu rakety TOM II. Proces analýzy dat byl chronologicky sepsán, tak aby čtenář práce dokázal napodobit způsob provedené analýzy, dle svých letů, a získal (tak jak je tomu v této práci), co nejobsáhlejší množství hodnot fyzikálních veličin a informací popisující let jednoduché rakety. I přes nekompletní letová data se podařilo hlavní cíl diplomové práce splnit. Doplnění analýzy pro kompletní letová data by mohlo být bráno jako případná možnost, jak tuto práci rozvinout, či jako domácí úkol pro zájemce tohoto tématu.

Vedlejší cíle praktické části doplňují a rozšiřují cíl hlavní. Byl tedy popsán postup tvorby vodní rakety TOM II spolu s úpravami odpaliště. Dále též využité hardwarové prvky měřicího zařízení se schématem zapojení a softwarová část ve formě rozboru kódu. Jednotlivé kroky tvorby byly doplněny dostatečným množstvím obrázků, tak aby byla snadná případná rekonstrukce. V rámci předletové i letové části vznikly i videozáznamy doplněné mluveným komentářem. Pro určení finanční náročnosti bylo provedeno vyčíslení ceny rakety, odpaliště a měřicího zařízení, tak aby vznikla jasnější představa proveditelnosti. V posledním bodě byly vytvořeny pracovní listy, se kterými mohou učitelé a žáci pracovat uceleným a přehledným způsobem při tvorbě a po odpalu rakety.

## Seznam použité literatury

- [1] MUSIL, Tomáš. *Měření zrychlení a rychlosti jednoduché rakety* [online]. Hradec Králové, 2018. Dostupné z: <<https://theses.cz/id/m8ueh6/>>.
- [2] National Physical Laboratory. *A guide to building and understanding the physics of Water Rockets* [online]. Poslední revize 1. 7. 2007 [cit. 2021-04-12], verze 1.02, 50 s., Dostupné z: [https://www.npl.co.uk/skills-learning/outreach/water-rockets/wr\\_booklet\\_print.pdf](https://www.npl.co.uk/skills-learning/outreach/water-rockets/wr_booklet_print.pdf)
- [3] National Aeronautics and Space Administration. *All about water rockets* [online]. Poslední revize 12. 6. 2014 [cit. 2021-04-12]. Dostupné z: <https://www.grc.nasa.gov/www/k-12/rocket/BottleRocket/about.htm>
- [4] National Physical Laboratory. *Water rockets* [online]. © National Physical Laboratory 2021 [cit. 2021-04-12]. Dostupné z: <https://www.npl.co.uk/water-rockets>
- [5] Arduino®. *Introduction* [online]. © 2021 Arduino [cit. 2021-04-02]. Dostupné z: <https://www.arduino.cc/en/Guide/Introduction>
- [6] Bastlírna HWKitchen. *Co je to Arduino?* [online]. [cit. 2021-04-02]. Dostupné z: <https://bastlirna.hwkitchen.cz/co-je-to-arduino/>
- [7] VODA, Zbyšek. *Průvodce světem Arduina*. Bučovice: Martin Stříž, 2015. ISBN 978-80-87106-90-7.
- [8] BABJAK, Ján. *Řídicí systémy servisních robotů: Control systems of service robots*. Ostrava: VŠB – Technická univerzita, 2013. ISBN 978-80-248-3123-7. Dostupné z: <https://ndk.cz/view/uuid:f9271120-a4ae-11e5-b5dc-005056827e51?page=uuid:36055480-bec2-11e5-b404-005056825209>
- [9] Arduino®. *Arduino Nano* [online]. © 2021 Arduino [cit. 2021-04-02]. Dostupné z: <https://store.arduino.cc/arduino-nano>
- [10] FITZGERALD, Scott and SHILOH, Michael. *ARDUINO – THE PROJECTS BOOK*. Kniha je součástí „The Arduino Starter kit“
- [11] *Wiring* [online]. [cit. 2021-04-02]. Dostupné z: <http://wiring.org.co/>
- [12] *Processing* [online]. [cit. 2021-04-02]. Dostupné z: <https://processing.org/>
- [13] SELECKÝ, Matúš. *Arduino: uživatelská příručka*. Přeložil Martin HERODEK. Brno: Computer Press, 2016. ISBN 978-80-251-4840-2.
- [14] NOVÁK, Jan A. *Drony: kompletní průvodce včetně přehledu nové legislativy*. Praha: Grada Publishing, 2021. ISBN 978-80-271-0775-9.
- [15] MARTINEK, Radislav. *Senzory v průmyslové praxi*. Praha: BEN – technická literatura, 2004. s. 11. ISBN 80-7300-114-4. Dostupné také z: <https://ndk.cz/view/uuid:425f58e0-ec94-11e8-a5a4-005056827e52>
- [16] WARREN, John-David, Josh ADAMS a Harald MOLLE. *Arduino robotics*. [New York, NY]: Apress, c2011. Technology in action series. ISBN 978-1-4302-3183-7.
- [17] Octopart [online]. © 2021 Octopart. [cit. 2021-04-02]. Dostupné z: <https://datasheet.octopart.com/MPU-6500-InvenSense-datasheet-138896167.pdf>
- [18] How To Mechatronics. *MEMS Accelerometer Gyroscope Magnetometer & Arduino* [online]. © 2019 HowToMechatronics.com [cit. 2021-04-02]. Dostupné z:

- <https://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyroscope-magnetometer-arduino/>
- [19] Science Direct. *Inertial Measurement* [online]. © 2021 Elsevier B.V. or its licensors or contributors [cit. 2021-04-02]. Dostupné z: <https://www.sciencedirect.com/topics/engineering/inertial-measurement>
- [20] SVOBODA, Emanuel. Přehled středoškolské fyziky. 5., přeprac. vyd. Praha: Prometheus, 2014. ISBN 978-80-7196-438-4.
- [21] Air Command Water Rockets. *Day 171 - How to Measure Altitude* [online]. © 2006-2020 Air Command Water Rockets [cit. 2021-04-02]. Dostupné z: <http://www.aircommandrockets.com/day171.htm>
- [22] Avnet Abacus. *MEMS pressure sensors* [online]. ©2021 Avnet, Inc [cit. 2021-04-02]. Dostupné z: <https://www.avnet.com/wps/portal/abacus/solutions/technologies/sensors/pressure-sensors/core-technologies/mems/>
- [23] ZEHNULA, Karel. Čidla robotů. Praha: SNTL-Nakladatelství technické literatury, 1990. s. 258. Dostupné také z: <https://ndk.cz/uuid/uuid:e1434af8-c3e7-4039-a6f0-4582e5ce517c>
- [24] Encyklopedie fyziky. *Wheatstoneův můstek* [online]. © 2006–2021, Encyklopedie podléhá licenci CC, Jaroslav Reichl, Martin Všeticka, Podpora vývoje [cit. 2021-04-02]. Dostupné z: <http://fyzika.jreichl.com/main.article/view/1631-wheatstoneuv-mustek>
- [25] How to Mechatronics. *How I2C Communication Works and How To Use It with Arduino* [online]. © 2019 HowToMechatronics.com [cit. 2021-04-02]. Dostupné z: <https://howtomechatronics.com/uncategorized/how-i2c-communication-works-and-how-to-use-it-with-arduino/>
- [26] Open Electronics. *Feeding power to Arduino: the ultimate guide* [online]. © 2020 Open Electronics [cit. 2021-04-03]. Dostupné z: <https://www.open-electronics.org/the-power-of-arduino-this-unknown/>
- [27] Battery Solutions. *Battery Types* [online]. © 2021 Battery Solutions [cit. 2021-04-05]. Dostupné z: <https://batterysolutions.com/learning-center/battery-types/>
- [28] Cybergibbons. *Arduino misconceptions 6: a 9 V battery is a good power source* [online]. Powered by WordPress [cit. 2021-04-05]. Dostupné z: <https://cybergibbons.com/arduino/arduino-misconceptions-6-a-9v-battery-is-a-good-power-source/>
- [29] Arduino forum. *Servo Motor not working when used with a 9v battery* [online]. © 2021 Arduino [cit. 2021-04-05]. Dostupné z: <https://forum.arduino.cc/index.php?topic=656402.0>
- [30] Nejlepší baterie. *Nezávislý test baterií* [online]. A SiteOrigin Theme [cit. 2021-04-05]. Dostupné z: <http://www.nejlepsi-baterie.cz/>
- [31] Last Minute Engineers. *How Servo Motor Works & Interface It With Arduino* [online]. © 2021 LastMinuteEngineers.com [cit. 2021-04-09]. Dostupné z: <https://lastminuteengineers.com/servo-motor-arduino-tutorial/>



- [32] Arduino. *Memory* [online]. © 2021 Arduino [cit. 2021-04-09]. Dostupné z: <https://www.arduino.cc/en/Tutorial/Foundations/Memory>
- [33] Automa. *Proč kalibrovat? Význam kalibrace* [online]. © 2016 Automa – časopis pro automatizační techniku, s. r. o.. Dostupné z: [https://automa.cz/Aton/FileRepository/pdf\\_articles/11886.pdf](https://automa.cz/Aton/FileRepository/pdf_articles/11886.pdf)
- [34] KANIA, Patrik a URBAN, Štěpán. Modulace a Šum signálu. Ústav analytické chemie, VŠCHT Praha [online]. [cit. 2021-04-12]. Dostupné z: <https://old.vscht.cz/anl/lmsvr/Modulace%20a%20%C5%A1um%20sign%C3%A1lu.pdf>
- [35] Rámcový vzdělávací program pro základní vzdělávání. [online]. Praha: Výzkumný ústav pedagogický v Praze, 2021. 164 s. [cit. 2021-07-06]. Dostupné z: [http://www.nuv.cz/file/4983\\_1\\_1/](http://www.nuv.cz/file/4983_1_1/)
- [36] Rámcový vzdělávací program pro gymnázia. [online]. Praha: Výzkumný ústav pedagogický v Praze, 2007. 100 s. [cit. 2021-07-06]. Dostupné z: [http://www.vuppraha.rvp.cz/wp-content/uploads/2009/12/RVPG-2007-07\\_final.pdf](http://www.vuppraha.rvp.cz/wp-content/uploads/2009/12/RVPG-2007-07_final.pdf). ISBN 978-80-87000-11-3.
- [37] Air Command Water Rockers. *How much water* [online]? © 2006-2020 Air Command Water Rockets [cit. 2021-04-26]. Dostupné z: <http://www.aircommandrockets.com/water.htm>
- [38] Air Command Water Rockers. *Robinson Coupling* [online]. © 2006-2020 Air Command Water Rockets [cit. 2021-04-26]. Dostupné z: [http://www.aircommandrockets.com/construction\\_2.htm](http://www.aircommandrockets.com/construction_2.htm)
- [39] Air Command Water Rockers. *Parachutes* [online]? © 2006-2020 Air Command Water Rockets [cit. 2021-04-26]. Dostupné z: <http://www.aircommandrockets.com/parachutes.htm>
- [40] Keisan Online Calculator. *Altitude from atmospheric pressure Calculator* [online]. © 2021 CASIO COMPUTER CO., LTD. [cit. 2021-04-26]. Dostupné z: <https://keisan.casio.com/has10/SpecExec.cgi?id=system/2006/1224585971>
- [41] Arduino forum. *Why is the SD library slow* [online]? © 2020 Arduino [cit. 2021-04-26]. Dostupné z: <https://forum.arduino.cc/t/why-is-the-sd-library-slow/49791>
- [42] GitHub. *Library for MPU-9250 by Bolder Flight Systems* [online]. © 2021 GitHub, Inc. [cit. 2021-04-26]. Dostupné z: <https://github.com/bolderflight/MPU9250>
- [43] GitHub. *Library for BME280 and BMP 280 by Christandlg* [online]. © 2021 GitHub, Inc. [cit. 2021-04-26]. Dostupné z: <https://github.com/christandlg/BMx280MI>
- [44] MUSIL, Tomáš. *Videozáznam diplomové práce* [online]. Dostupné z: <https://youtu.be/tBL8LQxcQFM>
- [45] MUSIL, Tomáš. *Odpal rakety TOM II* [online]. Dostupné z: <https://youtu.be/bjIQZlfrxLY>
- [46] HEATH, Clifford. *Water Rocket Simulator* [online]. [cit. 2021-07-07]. Dostupné z: <https://cjh.polyplex.org/rockets/simulation>

# Seznam a zdroje obrázků

Obrázek č. 1 – Vodní raketa jednoduše.....	12
Převzato a upraveno z: National Physical Laboratory. <i>A guide to building and understanding the physics of Water Rockets</i> [online]. Poslední revize 1. 7. 2007 [cit. 2021-04-12], verze 1.02, 50 s., Dostupné z: <a href="https://www.npl.co.uk/skills-learning/outreach/water-rockets/wr_booklet_print.pdf">https://www.npl.co.uk/skills-learning/outreach/water-rockets/wr_booklet_print.pdf</a>	
Obrázek č. 2 – Různé způsoby spojování PET lahví (odshora: jenom Robinson coupling, jenom Robinson coupling druhý způsob, splicing + Robinson coupling, jenom splicing)....	14
Převzato a upraveno z: Fullbore Water Rocket Launchers. <i>Threaded Tube Coupling</i> [online]. © Copyright S.J.Maidment 2014 [cit. 2021-04-16]. Dostupné z: <a href="https://rocketfun.com/PDF%20Files/ThreadedTubeJoiner.pdf">https://rocketfun.com/PDF%20Files/ThreadedTubeJoiner.pdf</a>	
Obrázek č. 3 – Logo Arduino .....	16
Převzato z: Wikimedia Commons. <i>Arduino Logo</i> [online]. [cit. 2021-04-20]. Dostupné z: <a href="https://commons.wikimedia.org/wiki/File:Arduino_Logo.svg">https://commons.wikimedia.org/wiki/File:Arduino_Logo.svg</a>	
Obrázek č. 4 – Arduino Nano v3.0 (vrchní a spodní pohled) .....	17
Zdroj: Vlastní tvorba	
Obrázek č. 5 – IDE Arduino.....	19
Zdroj: Vlastní tvorba	
Obrázek č. 6 – Ukázka přídavných (externích) modulů, dole Arduino Nano 3.0 .....	20
Zdroj: Vlastní tvorba	
Obrázek č. 7 – Ukázka shieldu pro Arduino Nano .....	20
Převzato z: Laskarduino. <i>Arduino Nano terminál shield</i> [online]. © Copyright 2021 laskarduino.cz. Všechna práva vyhrazena [cit. 2021-05-02]. Dostupné z: <a href="https://www.laskarduino.cz/arduino-nano-terminal-shield/">https://www.laskarduino.cz/arduino-nano-terminal-shield/</a>	
Obrázek č. 8 – Ukázka nepájivých polí.....	21
Zdroj: Vlastní tvorba	
Obrázek č. 9 – Schéma nepájivého pole.....	21
Zdroj: Vlastní tvorba	
Obrázek č. 10 – Vodiče neosazené.....	22
Zdroj: Vlastní tvorba	
Obrázek č. 11 – Vodiče osazené (typ od shora: M-M, M-F, F-F, M-M).....	22
Zdroj: Vlastní tvorba	
Obrázek č. 12 – MEMS – akcelerometr.....	24
Převzato z: Moodle FEL ČVUT. <i>Moderní senzory a zpracování informací</i> [online]. [cit. 2021-05-10]. Dostupné z: <a href="https://moodle.fel.cvut.cz/pluginfile.php/87858/mod_resource/content/1/01-%C3%BAvod-MSZ-2017.pdf">https://moodle.fel.cvut.cz/pluginfile.php/87858/mod_resource/content/1/01-%C3%BAvod-MSZ-2017.pdf</a>	
Obrázek č. 13 – MEMS – akcelerometr – nákres .....	24
Zdroj: Vlastní tvorba	
Obrázek č. 14 – Coriolisův efekt .....	25
Převzato a upraveno z: How to Mechatronics. <i>MEMS Accelerometer Gyroscope Magnetometer &amp; Arduino</i> [online]. Copyright © 2019 HowToMechatronics.com. All	

rights reserved [cit. 2021-05-10]. Dostupné z: <a href="https://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyroscope-magnetometer-arduino/">https://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyroscope-magnetometer-arduino/</a>	
Obrázek č. 15 – MEMS – gyroskop – nákras.....	26
Zdroj: Vlastní tvorba	
Obrázek č. 16 – MEMS – barometr – schéma.....	27
Převzato a upraveno z: AVNET. <i>MEMS pressure sensors</i> [online]. Copyright ©2021 Avnet, Inc. All rights reserved [cit. 2021-05-16]. Dostupné z: <a href="https://www.avnet.com/wps/portal/abacus/solutions/technologies/sensors/pressure-sensors/core-technologies/mems/">https://www.avnet.com/wps/portal/abacus/solutions/technologies/sensors/pressure-sensors/core-technologies/mems/</a>	
Obrázek č. 17 – I <sup>2</sup> C protokol.....	28
Převzato a upraveno z: How to Mechatronics. <i>How I2C Communication Works and How To Use It with Arduino</i> [online]. Copyright © 2019 HowToMechatronics.com. All rights reserved [cit. 2021-05-20]. Dostupné z: <a href="https://howtomechatronics.com/tutorials/arduino/how-i2c-communication-works-and-how-to-use-it-with-arduino/">https://howtomechatronics.com/tutorials/arduino/how-i2c-communication-works-and-how-to-use-it-with-arduino/</a>	
Obrázek č. 18 – 9 V alkalická baterie v boxu se zabudovaným vypínačem .....	31
Zdroj: Vlastní tvorba	
Obrázek č. 19 – 4 lithiové AA baterie v boxu (sériově) s přibudovaným vypínačem.....	31
Zdroj: Vlastní tvorba	
Obrázek č. 20 – Li-Pol akumulátor .....	32
Převzato z: Laskarduino. <i>EHAO LiPol Baterie 104050 2000mAh 3.7V</i> [online]. © Copyright 2021 laskarduino.cz. Všechna práva vyhrazena [cit. 2021-05-20]. Dostupné z: <a href="https://www.laskarduino.cz/baterie-li-po-3-7v-2000mah-lipo/">https://www.laskarduino.cz/baterie-li-po-3-7v-2000mah-lipo/</a>	
Obrázek č. 21 – Ukázka servomotoru.....	32
Zdroj: Vlastní tvorba	
Obrázek č. 22 – Řídící systém servomotoru.....	33
Převzato a upraveno z: Last Minute Engineers. <i>How Servo Motor Works &amp; Interface It With Arduino</i> [online]. © 2021 LastMinuteEngineers.com [cit. 2021-04-09]. Dostupné z: <a href="https://lastminuteengineers.com/servo-motor-arduino-tutorial/">https://lastminuteengineers.com/servo-motor-arduino-tutorial/</a>	
Obrázek č. 23 – Průřez servomotoru .....	33
Převzato a upraveno z: Last Minute Engineers. <i>How Servo Motor Works &amp; Interface It With Arduino</i> [online]. © 2021 LastMinuteEngineers.com [cit. 2021-04-09]. Dostupné z: <a href="https://lastminuteengineers.com/servo-motor-arduino-tutorial/">https://lastminuteengineers.com/servo-motor-arduino-tutorial/</a>	
Obrázek č. 24 – Ukázka pulzně šířkové modulace pro servomotory .....	34
Převzato a upraveno z: Last Minute Engineers. <i>How Servo Motor Works &amp; Interface It With Arduino</i> [online]. © 2021 LastMinuteEngineers.com [cit. 2021-04-09]. Dostupné z: <a href="https://lastminuteengineers.com/servo-motor-arduino-tutorial/">https://lastminuteengineers.com/servo-motor-arduino-tutorial/</a>	
Obrázek č. 25 – Modul pro micro SD kartu .....	34
Převzato z: Laskarduino. <i>microSD Card modul SPI</i> [online]. © Copyright 2021 laskarduino.cz. Všechna práva vyhrazena [cit. 2021-04-23]. Dostupné z: <a href="https://www.laskarduino.cz/microsd-card-modul-spi/">https://www.laskarduino.cz/microsd-card-modul-spi/</a>	
Obrázek č. 26 – Raketa TOM I.....	42
Zdroj: Vlastní tvorba	

Obrázek č. 27 – Konstrukční PET lahve 2,25 l.....	43
Převzato z: JASO. <i>PEPSI COLA 2,25 l</i> [online]. [cit. 2021-05-10]. Dostupné z: <a href="https://www.jaso.cz/pepsi-cola-2-25-l-ean42024-skup4030901.php?18_let=ano">https://www.jaso.cz/pepsi-cola-2-25-l-ean42024-skup4030901.php?18_let=ano</a>	
Převzato z: Facebook (KB Market Nižný Žipov). <i>Kofola 2,25 l</i> [online]. [cit. 2021-05-10]. Dostupné z: <a href="https://www.facebook.com/409395389892074/photos/%EF%B8%8Fakcia%EF%B8%8Fkofola-225l-097/717488589082751/">https://www.facebook.com/409395389892074/photos/%EF%B8%8Fakcia%EF%B8%8Fkofola-225l-097/717488589082751/</a>	
Obrázek č. 28 – Robinson coupling .....	44
Převzato a upraveno z: Air Command Water Rockers. <i>Robinson Coupling</i> [online]. © 2006-2020 Air Command Water Rockets [cit. 2021-04-26]. Dostupné z: <a href="http://www.aircommandrockets.com/construction_2.htm">http://www.aircommandrockets.com/construction_2.htm</a>	
Obrázek č. 29 – Trup rakety TOM II s přilepenou kapotáží .....	45
Zdroj: Vlastní tvorba	
Obrázek č. 30 – Detail kapotáže.....	45
Zdroj: Vlastní tvorba	
Obrázek č. 31 – Dva výstřižky PET lahve Pepsi na hlavici.....	46
Zdroj: Vlastní tvorba	
Obrázek č. 32 – Padák.....	46
Zdroj: Vlastní tvorba	
Obrázek č. 33 – Osazení pro měřící zařízení před a po vložení komponentů.....	47
Zdroj: Vlastní tvorba	
Obrázek č. 34 – Náskres pro osazení měřícího zařízení.....	48
Zdroj: Vlastní tvorba	
Obrázek č. 35 – Osazení pro padák a servomotor .....	48
Zdroj: Vlastní tvorba	
Obrázek č. 36 – Osazení pro měřící zařízení a padák s hlavici před sestavením.....	49
Zdroj: Vlastní tvorba	
Obrázek č. 37 – Dvířka hlavice pro padák .....	49
Zdroj: Vlastní tvorba	
Obrázek č. 38 – Kompletní hlavice .....	50
Zdroj: Vlastní tvorba	
Obrázek č. 39 – Upevněné stabilizátory .....	50
Zdroj: Vlastní tvorba	
Obrázek č. 40 – Tryska z rychlospojky.....	51
Zdroj: Vlastní tvorba	
Obrázek č. 41 – Kartonový disk mezi osazením a trupem .....	52
Zdroj: Vlastní tvorba	
Obrázek č. 42 – Podstavce na raketu .....	52
Zdroj: Vlastní tvorba	
Obrázek č. 43 – Body úchytu pro lanko padáku.....	53
Zdroj: Vlastní tvorba	
Obrázek č. 44 – Schéma uchycení padáku .....	53
Zdroj: Vlastní tvorba	

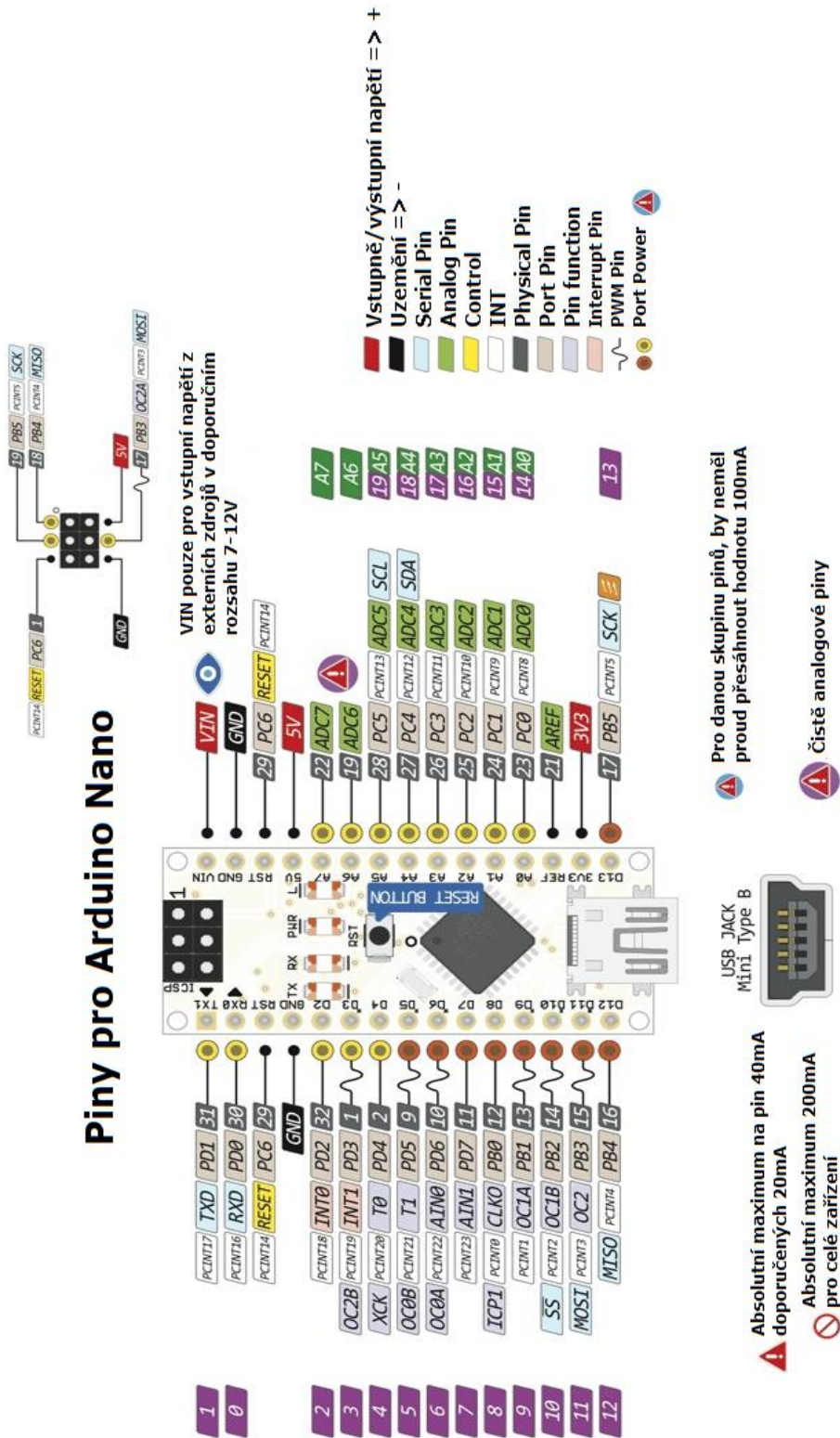
Obrázek č. 45 – Raketa TOM II bez zabaleného padáku.....	54
Zdroj: Vlastní tvorba	
Obrázek č. 46 – Držáky stojných tyčí.....	54
Zdroj: Vlastní tvorba	
Obrázek č. 47 – Připevnění nezaslepené spojky .....	55
Zdroj: Vlastní tvorba	
Obrázek č. 48 – Přívodní hadice pro tlakování rakety.....	56
Zdroj: Vlastní tvorba	
Obrázek č. 49 – Hustilka s manometrem .....	56
Zdroj: Vlastní tvorba	
Obrázek č. 50 – Dva přidržovače podpůrných tyčí.....	56
Zdroj: Vlastní tvorba	
Obrázek č. 51 – Rozbočovač.....	57
Zdroj: Vlastní tvorba	
Obrázek č. 52 – Postřikovač jako tlaková nádoba pro přívod vody do rakety .....	57
Zdroj: Vlastní tvorba	
Obrázek č. 53 – MPU-9250 .....	59
Převzato z: Drátek.cz. <i>9DOF Gyroskop + Akcelerometr + Magnetometr MPU 9250 SPI/IIC Modul pro Arduino</i> [online]. © Copyright ECLIPSERA s.r.o. Všechna práva vyhrazena [cit. 2021-05-25]. Dostupné z: <a href="https://dratek.cz/arduino/1330-9dof-gyroskop-akcelerometr-magnetometr-mpu-9250-spi-iic-modul-pro-arduino.html">https://dratek.cz/arduino/1330-9dof-gyroskop-akcelerometr-magnetometr-mpu-9250-spi-iic-modul-pro-arduino.html</a>	
Obrázek č. 55 – BME280.....	59
Převzato z: Vokolo. <i>SENZOR TEPLoty, TLAKU A VLHKOSTI VZDUCHU BME280</i> [online]. 2021 © Vokolo, všechna práva vyhrazena [cit. 2021-05-25]. Dostupné z: <a href="https://www.vokolo.cz/bme280/?variantId=65651">https://www.vokolo.cz/bme280/?variantId=65651</a>	
Obrázek č. 55 – 5 V lineární stabilizátor napětí.....	60
Zdroj: Vlastní tvorba	
Obrázek č. 56 – Pracovní zapojení.....	60
Zdroj: Vlastní tvorba	
Obrázek č. 57 – Ukázka surových dat.....	69
Zdroj: Vlastní tvorba	
Obrázek č. 58 – Grafické rozdělení dat .....	70
Zdroj: Vlastní tvorba	
Obrázek č. 59 – Ukázka postupného součtu .....	70
Zdroj: Vlastní tvorba	
Obrázek č. 60 – Kalibrační odchylky tíhového zrychlení (nepřesné) .....	71
Zdroj: Vlastní tvorba	
Obrázek č. 61 – Kalibrační odchylky tíhového zrychlení (zpřesněné) .....	72
Zdroj: Vlastní tvorba	
Obrázek č. 62 – Kalibrovaná akcelerometrická data.....	72
Zdroj: Vlastní tvorba	
Obrázek č. 63 – Velikost zrychlení.....	72
Zdroj: Vlastní tvorba	

Obrázek č. 64 – Zrychlení jednotlivých os (očistěných).....	73
Zdroj: Vlastní tvorba	
Obrázek č. 65 – Výpočet dílčích rychlostí a jejich postupný součet .....	74
Zdroj: Vlastní tvorba	
Obrázek č. 66 – Výpočet dílčích drah a jejich postupný součet.....	75
Zdroj: Vlastní tvorba	
Obrázek č. 67 – Převod rad/s na °/s .....	76
Zdroj: Vlastní tvorba	
Obrázek č. 68 – Výpočet dílčího natočení a jeho postupný součet.....	76
Zdroj: Vlastní tvorba	
Obrázek č. 69 – Derivace rotace rakety osy z pro získání rychlosti rotace.....	77
Zdroj: Vlastní tvorba	

## Seznam grafů

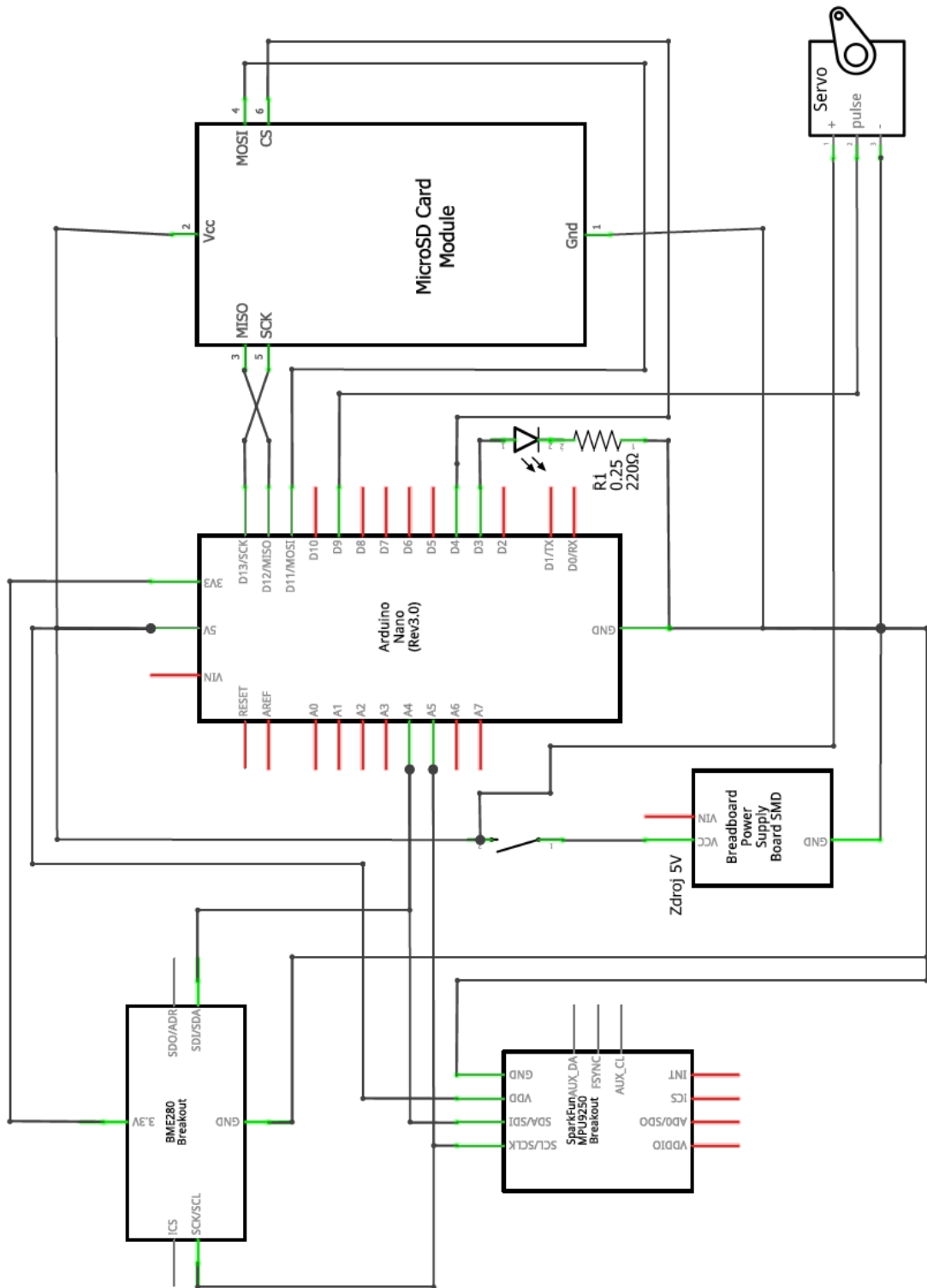
Graf č. 1 – Ukázka šumu v signálu .....	37
Graf č. 2 – Průběh zrychlení rakety .....	73
Graf č. 3 – Průběh rychlosti rakety .....	74
Graf č. 4 – Průběh dráhy letu rakety .....	75
Graf č. 5 – Průběh rotace rakety .....	77
Graf č. 6 – Průběh rychlosti rotace rakety .....	78
Graf č. 7 – Průběh zrychlení rotace rakety .....	78
Graf č. 8 – Průběh změny tlaku vzduchu .....	79
Graf č. 9 – Průběh změny tlaku vzduchu (aproximace).....	80
Graf č. 10 – Průběh změny nadmořské výšky .....	80
Graf č. 11 – Průběh tíhové síly, tahové síly a zrychlení .....	82

# Příloha



Příloha č. 1 - Rozložení pinů na Arduino Nano [Upraveno ze zdroje: Twins Chip. ARDUINO NANO V3.0 [online]. Copyright © 2010–2020, Twins Chip ® [cit. 2021-05-20]. Dostupné z: [https://www.twinschip.com/Arduino\\_Nano\\_V3.0](https://www.twinschip.com/Arduino_Nano_V3.0)]





Příloha č. 2 – Schéma zapojení měřícího zařízení [Zdroj: Vlastní tvorba]

```

1 //Vytvořeno Tomášem Musilem, 2020/21
2 //-----
3 //Inicializační část
4 //-----
5 //Inicializace I2C sběrnice
6 #include <Wire.h>
7 #include "I2Cdev.h"
8
9 //Inicializace senzoru MPU-9250 (akcelerometr + gyroskop + magnetometr)
10 #include "MPU9250.h"
11 MPU9250 IMU(Wire, 0x68);
12 int status;
13
14 //Inicializace senzoru BME280 (barometr - výškoměr + teploměr)
15 #include <Arduino.h>
16 #include "BMx280I2C.h"
17 #define I2C_ADDRESS 0x76
18 BMx280I2C bmx280(I2C_ADDRESS);
19
20 //Inicializace micro SD modulu
21 #include <SPI.h>
22 #include <SD.h>
23 File myFile;
24
25 //Inicializace servomotoru
26 #include <Servo.h>
27 Servo myservo;
28
29 //Inicializace proměnných
30 long timer, d_t, timer_helper;
31 int const pinCS = 4;
32 byte buff = 0;
33 boolean servo_stop = true;
34 byte servo_help = 0;

```

*Příloha č. 3 – Kód – inicializační část [Zdroj: Vlastní tvorba]*

```

36 //Nastavovací část
37 //-----
38 void setup() {
39   //Nastavení rychlosti přenosu dat (baud rate)
40   Serial.begin(38400);
41
42   //Nastavení servomotoru
43   myservo.attach(9);
44   myservo.write(0);
45
46   //Nastavení kontrolní LED
47   pinMode(3, OUTPUT);
48
49   //Kontrola a spuštění micro SD modulu
50   if (!SD.begin(pinCS)) {
51     Serial.println(F("Inicializace SD karty selhala!"));
52     while (1) {}
53   }
54   else {
55     myFile = SD.open("test.txt", O_CREAT | O_WRITE | O_APPEND);
56     myFile.println(F("Inicializace SD karty uspesna!"));
57     Serial.println("Inicializace SD karty uspesna!");
58   }
59
60   //Nastavení senzoru MPU-9250
61   status = IMU.begin();
62   if (status < 0) {
63     Serial.println(F("MPU fail!"));
64     myFile.println(F("MPU fail!"));
65     myFile.println(status);
66     while (1) {}
67   }
68   myFile.println(F("Senzor MPU-9250 pripraven"));
69   Serial.println(F("Senzor MPU-9250 pripraven"));

```

*Příloha č. 4 – Kód – nastavovací část (stránka č. 1) [Zdroj: Vlastní tvorba]*

```

70
71 //Nastavení senzoru BME280
72 if (!bmx280.begin()) {
73     Serial.println(F("BME280 fail!"));
74     myFile.println(F("BME280 fail!"));
75     while (1) {}
76 }
77 if (bmx280.isBME280()) {
78     Serial.println(F("Senzor BME280 připraven"));
79     myFile.println(F("Senzor BME280 připraven"));
80 }
81 //Překonfigurování BME280
82 bmx280.resetToDefaults();
83 //Převzorkování senzoru tlaku
84 bmx280.writeOversamplingPressure(BMx280MI::OSRS_P_x04);
85 //Převzorkování senzoru teploty
86 bmx280.writeOversamplingTemperature(BMx280MI::OSRS_T_x01);
87
88 //Servo do klidového módu (šetří energii, ale může se povolit při pnutí!)
89 myservo.detach();
90
91 //Start měření
92 Serial.println(F("Start měření"));
93 myFile.println(F("Start měření"));
94 myFile.flush();
95 myFile.close();
96 myFile = SD.open("test.txt", O_CREAT | O_WRITE | O_APPEND);
97 }
98

```

*Příloha č. 5 – Kód – nastavovací část (stránka č. 2) [Zdroj: Vlastní tvorba]*

```

99 //Část datového sběru
100 //-----
101 void loop() {
102     //Zanesení delta_t
103     timer = millis();
104     d_t = timer - timer_helper;
105     timer_helper = timer;
106
107     //Získání hodnot z akcelerometru
108     IMU.readSensor();
109
110     //Podmínka pro zajištění funkčnosti barometru
111     if (!bmx280.measure())
112     {
113         return;
114     }
115     //Čekání na barometr pro získání hodnot
116     do
117     {
118
119     } while (!bmx280.hasValue());
120
121     //Zápis a výpis hodnot
122     if (buff < 100) {
123         Serial.print("\t"); Serial.print(IMU.getAccelX_mss(), 2);
124         Serial.print("\t"); Serial.print(IMU.getAccelY_mss(), 2);
125         Serial.print("\t"); Serial.print(IMU.getAccelZ_mss(), 2);
126         Serial.print("\t"); Serial.print(IMU.getGyroX_rads(), 2);
127         Serial.print("\t"); Serial.print(IMU.getGyroY_rads(), 2);
128         Serial.print("\t"); Serial.print(IMU.getGyroZ_rads(), 2);
129         Serial.print("\t"); Serial.print(bmx280.getPressure64());
130         Serial.print("\t"); Serial.print(bmx280.getTemperature());
131         Serial.print("\t"); Serial.println(d_t);
132         myFile.print("\t"); myFile.print(IMU.getAccelX_mss(), 2);
133         myFile.print("\t"); myFile.print(IMU.getAccelY_mss(), 2);
134         myFile.print("\t"); myFile.print(IMU.getAccelZ_mss(), 2);
135         myFile.print("\t"); myFile.print(IMU.getGyroX_rads(), 2);
136         myFile.print("\t"); myFile.print(IMU.getGyroY_rads(), 2);
137         myFile.print("\t"); myFile.print(IMU.getGyroZ_rads(), 2);
138         myFile.print("\t"); myFile.print(bmx280.getPressure64());
139         myFile.print("\t"); myFile.print(bmx280.getTemperature());
140         myFile.print("\t"); myFile.println(d_t);
141         buff = buff + 1;
142     }

```

```

143 else
144 {
145 //Vyčištění vyrovnávací paměti, po odeslání předchozích znaků
146 myFile.flush();
147 myFile.close();
148 myFile = SD.open("test.txt", O_CREAT | O_WRITE | O_APPEND);
149 buff = 0;
150 }
151
152 //Podmínka pro sepnutí serva při přetočení rakety
153 //Hlavice směřující dolů
154 if (IMU.getAccelZ_mss() > 0) {
155     if (servo_stop) {
156         servo_help = 0;
157         myservo.attach(9);
158         myservo.write(160);
159         servo_stop = false;
160     }
161     else {
162         digitalWrite(3, LOW);
163         if (servo_help < 100) {
164             servo_help = servo_help + 1;
165         } else {
166             myservo.detach();
167         }
168     }
169 }
170 //Hlavice směřující nahoru
171 else {
172     if (servo_stop) {
173         digitalWrite(3, HIGH);
174         if (servo_help < 100) {
175             servo_help = servo_help + 1;
176         }
177         else {
178             myservo.detach();
179         }
180     }
181     else {
182         servo_help = 0;
183         myservo.attach(9);
184         myservo.write(0);
185         servo_stop = true;
186     }
187 }
188 }

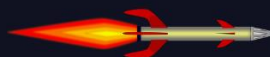
```

# RAKETÁCKÝ ZÁZNAMNÍK

**JMÉNO/A RAKETÁKA/Ů:**

**NÁZEV RAKETY:**

**PROSTOR PRO OBRÁZEK RAKETY**



# PŘEDLETOVÉ INFORMACE

## INFORMACE O RAKETĚ

**VNITŘNÍ OBJEM TRUPU:**

**SUCHÁ HMOTNOST:**

**VÝŠKA RAKETY:**

**PRŮMĚR RAKETY:**

**PRŮMĚR TRYSKY:**

**HMOTNOST KAPALINY:**

**STARTOVACÍ HMOTNOST:**

**TLAK PŘI ODPALU:**

## DOPLŇUJÍCÍ INFORMACE

**DATUM A ČAS ODPALU:**

**MÍSTO ODPALU:**

**TEPLOTA VZDUCHU:**

**TLAK VZDUCHU PŘEPOČTENÝ NA HLADINU MOŘE:**

**POČASÍ:**



# PARAMENTRY LETU

**LEVEL 1**

**LEVEL 2**

**LEVEL 3**

**Doba letu:**

**Průměrné  $\Delta t$ :**

**Maximální zrychlení:**

**Nadmořská výška před startem:**

**Maximální rychlost:**

**Délka dráhy letu:**

**Celková podélná rotace:**

**Dosažená vzletová výška:**

**Průměrná spotřeba kapaliny:**

**Maximální rychlost podélné rotace:**

**Maximální zrychlení podélné rotace:**

**Průměrný tah:**

**Poznámky z letu:**



# GRAFY

**VYBER JEDNU Z MOŽNOSTÍ U KAŽDÉ ODRÁŽKY A VLOŽ GRAF**

- ZRYCHLENÍ, RYCHLOST, DRÁHA
- PRŮBĚH ROTACE OS, RYCHLOST PODÉLNÉ ROTACE
- ATMOSFÉRICKÝ TLAK, NADMOŘSKÁ VÝŠKA

**BONUSOVÝ**

- TAH



# ZÁZNAMNÍK LETŮ RAKET

**Jméno/a raketáka/ů:**

**Název rakety:**

Číslo letu	Hmotnost kapaliny / kg	Tlak při startu / kPa	Maximální akcelerace / m/s <sup>2</sup>	Maximální rychlost / m/s	Vzletová výška / m	Podélná rotace / °	Průměrná spotřeba kapaliny / kg/s	Průměrný tah / N
1.								
2.								
3.								
4.								
5.								
6.								
7.								
8.								
9.								
10.								
Průměr								

**Nejprovedenější číslo letu:**

**Poznámky:**