

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ**

Katedra mikroelektroniky

DIPLOMOVÁ PRÁCE

2015

Bc. Petr Bařina

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra mikroelektroniky

Pohyblivý modul na principu přenosu momentu síly

*Moving module based on the principle of force moment
transfer*

DIPLOMOVÁ PRÁCE

květen 2015

Diplomant: Bc. Petr Bařina
Vedoucí práce: Ing. Adam Bouřa, Ph.D.

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra mikroelektroniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. BAŘINA Petr**

Studijní program: Komunikace, multimédia a elektronika
Obor: Elektronika

Název tématu: **Pohyblivý modul na principu přenosu momentu síly**

Pokyny pro vypracování:

1. Seznamte se s problematikou řízení a regulace elektrických stejnosměrných motorů a mechanismem přenosu momentu síly. Prostudujte možnosti snímání rychlosti otáček pomocí gyroskopu a seznamte se s principy přenosu měřitelných dat.
2. Na základě získaných poznatků zvolte vhodné technické řešení a realizujte modul, který umožní kontrolovaný pohyb pomocí principu přenosu momentu síly.
3. Sestavte testovací pracoviště, proveďte měření a kriticky zhodnoťte dosažené parametry vašeho řešení. Na základě výsledků testování případně navrhnete další možná vylepšení.

Seznam odborné literatury:

- [1] Feynman, R. P., Leighton, R. B., Sands M.: Feynmanovy přednášky z fyziky – revidované vydání – 1. díl, Fragment 2013, ISBN: 978-80-253-1642-9
- [2] Matoušek, D.: Práce s mikrokontroléry ATMEL AVR AT mega 16, 4. díl, Praha 2006, Nakladatelství BEN - technická literatura, ISBN 80-7300-174-8.
- [3] Mann, B.: C pro mikrokontroléry, 1. Vydání, Praha 2003, Nakladatelství BEN - technická literatura, ISBN 80-7300-077-6.
- [4] Vedral, J., Fischer, J.: Elektronické obvody pro měřicí techniku, Vydavatelství ČVUT, Praha, 2004, ISBN 80-01-02966-2

Vedoucí: **Ing. Adam Bouřa, Ph.D.**

Platnost zadání: 31. 8. 2016

prof. Ing. Miroslav Husák, CSc.
vedoucí katedry



prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 4. 2. 2015

Čestné prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Datum: 7. 5. 2015

.....
podpis diplomanta

Poděkování

Rád bych poděkoval Ing. Adamu Bouřovi, Ph.D. za vedení a konzultace diplomové práce. Dále bych chtěl poděkovat svému otci za odborné rady při návrhu a výrobě konstrukčního řešení diplomové práce. Rád bych také poděkoval oběma svým rodičům a přítelkyni za podporu a možnost studia na vysoké škole.

Abstrakt

V úvodní části práce se zabývám obecnou problematikou stejnosměrných motorů, jejich řízení a regulací otáček. V této části se také věnuji obecnému principu mechanismu přenosu momentu síly. Následuje problematika snímání rychlosti otáček za pomoci MEMS gyroskopu, kde uvádím obecný princip gyroskopu a následný popis funkčnosti gyroskopu MEMS. Závěr úvodní části je věnován sériovému komunikačnímu rozhraní, kde popisuji možnost odesílání dat po sériové lince za pomoci bezdrátového spojení. V praktické části práce popisuji vlastní návrh a sestavení pohyblivého modulu, který se pohybuje na základě mechanismu přenosu momentu síly. Zařízení je zároveň schopné detekovat svou rychlost otáčení za pomoci MEMS gyroskopu a tato data následně odesílat po bezdrátovém sériovém komunikačním rozhraní do počítače. Součástí praktické části práce je i vytvoření ovládacího programu, ve kterém je možné zadávat požadovanou rychlost otáčení celého modulu a následně tuto rychlost kontrolovat. V ovládacím programu je dále možné určovat směr pohybu modulu, nebo jeho okamžité zastavení. Na závěr práce uvádím kritické zhodnocení naměřených parametrů během testování zařízení a v závislosti na nich navrhuji možná vylepšení funkčního zařízení.

Klíčová slova

Mikrokontrolér, stejnosměrný motor, UART, gyroskop, bluetooth, pohyblivý modul, ovládací program.

Abstract

In the introductory part of the work, I deal with general issues of DC motors, their control and regulation of rotations. In this part, I also deal with the principle of the mechanism of force moment transfer. Following is the issue of sensing rotational speed with the help of MEMS gyroscope, with regards to the general principle of the gyroscope and the subsequent description of the functionality of the MEMS gyroscope. The conclusion of the first part is devoted to a serial communications interface where I describe possibility of sending data over a serial line, using the wireless connection. The practical part describes the custom design and build of movable module which moves based on the mechanism of force moment transfer. The device is also able to detect its rotation of speed with the aid of MEMS gyroscope and the data subsequently sent over a wireless serial communication interface to the computer. The practical part of this work is to create an application in which you can select the desired speed of rotation of the whole module and then to control the speed. The application is also able to determine the direction of movement of the module or its immediate stop. In the end of the work, I mention the critical evaluation of measured parameters during testing of equipment, and depending on them, I suggest possible improvements of functional devices.

Keywords

Microcontroller, DC motor, UART, gyroscope, bluetooth, moving module, control program.

Seznam zkratek a pojmů

V následující tabulce jsou uvedeny zkratky a odborné termíny, které jsou užity v textu tohoto dokumentu. V tabulce nejsou uvedeny běžně užívané zkratky a pojmy.

A/D	Analog/Digital	Analogově digitální
AVR	Advanced Virtual RISC	Pokročilá virtuální RISC
Bluetooth	-	Bezdrátová komunikace na krátkou vzdálenost
COM	Common Object Model	Obecný objektový model
DC	Direct Current	Stejnoseměrný proud
EEPROM	Electrically Erasable Programmable ROM	Elektricky mazatelná paměť ROM
GPS	Global Positioning System	Americký vojenský polohový družicový systém
I/O	Input/Output	Vstup/výstup
I2C	Inter – Integrated Circuit	Multi-masterová sériová počítačová sběrnice
LCD	Liquid Crystal Display	Displej z tekutých krystalů
LED	Light Emitting Diode	Svítilná dioda
MEMS	Micro-Electro-Mechanical-Systems	Mikro-elektro-mechanické systémy
MIPS	Million Instructions Per Second	Milión instrukcí za vteřinu
PC	Personal Computer	Osobní počítač
PWM	Pulse Width Modulation	Pulsně šířková modulace
RC	Resistor/Capacitor	Rezistor/kondenzátor
RISC	Reduced Instruction Set Computing	Redukovaný instrukční soubor
ROM	Read Only Memory	Permanentní paměť
RX	Receive	Přijímat
SCL	Synchronous Clock	Hodinový signál
SDA	Synchronous Data	Datový signál
SPI	Serial Peripheral Interface	Sériové periferní rozhraní
SRAM	Static Random Access Memory	Statická paměť s libovolným výběrem
TX	Transmit	Vysílat
USART	Universal Synchronous Asynchronous Receiver Transmitter	Univerzální synchronní asynchronní přijímač a vysílač

Obsah

1. Úvod.....	1
2. Současný stav a motivace.....	2
3. Rozbor použitých prostředků	3
3.1 Řízení a regulace stejnosměrných motorů	3
3.1.1 Stejnosměrné motory	3
3.1.2 Řízení stejnosměrných motorů.....	4
3.1.3 Regulace otáček stejnosměrných motorů.....	5
3.1.4 PWM modulace u mikrokontroléru ATmega32.....	6
3.2 Mechanismus přenosu momentu síly	9
3.3 Snímání rychlosti otáček pomocí MEMS gyroskopu	11
3.3.1 Princip činnosti gyroskopu.....	12
3.3.2 Coriolisova síla.....	12
3.3.3 MEMS gyroskop	13
3.4 Sériové komunikační rozhraní	17
3.4.1 Sériová UART linka.....	17
3.4.2 Modul USART u mikrokontroléru ATmega32	18
4. Řešení funkčního zařízení	20
4.1 Úvod.....	20
4.2 Řešení - hardware	21
4.3 Elektronika pro pohyblivý modul	22
4.3.1 Řídicí jednotka	22
4.3.2 Senzor - gyroskop	25
4.3.3 Signalizační a ovládací prvky	27
4.3.4 H-můstek.....	28
4.3.5 DC – motor.....	29
4.3.6 Bluetooth modul.....	31
4.3.7 Napájecí napětí.....	33
4.4 Ovládací program.....	35
4.5 Řešení - software.....	37
4.5.1 Obecný vývojový diagram	38
4.5.2 Vývojový diagram pro BLOK č. 1.....	39
4.5.3 Vývojový diagram pro BLOK č. 2.....	40
4.5.4 Vývojový diagram pro BLOK č. 3.....	42

4.6	Konstrukční řešení	44
4.6.1	Kříž z pertinaxu	44
4.6.2	Plastová trubka	46
4.6.3	Olověné závaží pro udržení klidové polohy statoru	46
4.6.4	Celková konstrukce modulu	47
5.	Testování zařízení	48
6.	Možná vylepšení	52
7.	Závěr	53
8.	Seznam literatury	54
9.	Seznam obrázků	56
10.	Seznam tabulek	58
11.	Seznam příloh	59

1. Úvod

Tento projekt se zabývá problematikou řízení a regulace stejnosměrných motorů a mechanismu přenosu momentu síly. Cílem projektu je sestavení pohyblivého modulu, který bude svůj pohyb realizovat na základě přenosu momentu síly. Hlavní řídicí jednotkou modulu bude mikrokontrolér AVR, ke kterému bude připojený senzor rotace, komunikační modul a stejnosměrný motor. Celý modul bude ovládán pomocí bezdrátové komunikace z počítače, kde bude možné kontrolovat získaná data ze senzoru rotace a na základě těchto dat řídit a automaticky regulovat otáčky stejnosměrného motoru.

Práce je rozdělena na tři hlavní části. V první části uvedu teoretický rozbor použitých prostředků. V této části se budu nejprve věnovat popisu funkčnosti, řízení a regulaci otáček stejnosměrného motoru, dále principu přenosu momentu síly, snímání otáček pomocí senzoru rotace a nakonec se seznámím s problematikou okolo sériové komunikace UART. Ve druhé části se budu věnovat samotnému řešení funkčního zařízení, kde nejprve uvedu obecný návrh funkčního zařízení a následně jeho hardwarovou a softwarovou realizaci. Na konci tohoto projektu provedu testování celého funkčního zařízení a následně i možné návrhy pro zlepšení funkčnosti celého zařízení.

2. Současný stav a motivace

V dnešní době se velmi hojně využívají malé stejnosměrné motorky, které pohánějí různá zařízení. Ovšem velmi málo se využívá stejnosměrný elektromotor pro pohon modulu, kde podstatou pohybu je přenos momentu síly. Tento způsob pohybu se asi nejvíce využívá v zábavním průmyslu. Například známá hračka, kde se plyšové zvířátko neustále točí okolo malé koule, funguje právě na principu přenosu momentu síly. Dalším zábavním zařízením, které využívá pro svůj pohyb přenos momentu síly je robotická koule Sphero 2.0 z odolného plastu od společnosti Orbotix. Tato koule se jednoduše ovládá přes speciální aplikaci v chytrém telefonu. Pokud si uživatel k této kouli pořídí další zajímavé aplikace, může s takovou koulí hrát např. golf, nebo petanque.

Právě skutečnost, že se tento princip pohybu využívá velmi málo, mě motivovala k mé práci. Sestavení pohyblivého modulu na principu přenosu momentu síly je totiž z hlediska aplikace a využití velmi zajímavá myšlenka.

Jednou z hlavních částí takového pohyblivého zařízení by měl být inerciální snímač, který může snímat rychlost otáčení modulu, která je následně potřebná k regulaci otáčení celého modulu. Nápad sestavit systém, který bude mít na základě získaných dat z inerciálního snímače, regulovanou rychlost otáčení, se mi stal další motivací k tvorbě celé mé práce.

Hlavní myšlenkou mé práce je možnost zkonstruování pohyblivého modulu, který lze zcela uzavřít do trubky (pohyb dopředu a dozadu), nebo do koule (pohyb na všechny strany) a tím vytvořit izolované zařízení.

3. Rozbor použitých prostředků

3.1 Řízení a regulace stejnosměrných motorů

3.1.1 Stejnosměrné motory

Konstrukce

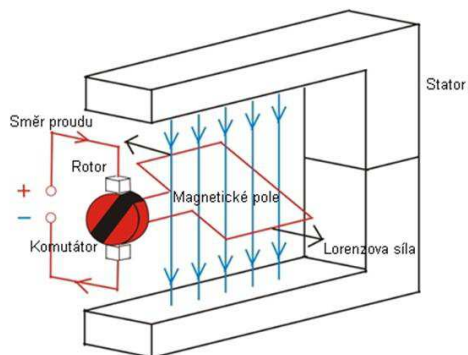
Stejnosměrný elektromotor se skládá ze dvou hlavních částí: rotoru a statoru.

- Rotor – obsahuje svazek elektrotechnických plechů s drážkami, ve kterých je umístěné vinutí. Vývody vinutí jsou připojené k lamelám komutátoru, které jsou vzájemně izolovány. Komutátor je přichycen k hřídeli elektromotoru. Další částí rotoru jsou kartáče, které dosedají na lamely komutátoru. Kartáči je přiváděn elektrický proud do vinutí rotoru.
- Stator – je tvořen permanentním magnetem, případně elektromagnetem. Permanentní magnet je složen ze svazků elektrotechnických plechů, které jsou na sebe naskládány tak, aby vytvořily tvar dutého válce. Druhá část, která je obsažená v permanentním magnetu je vinutí, které je napájené stejnosměrným elektrickým proudem. [20]

Princip činnosti

Po přivedení stejnosměrného napájecího napětí na kartáče rotoru vznikne elektrický proud na vinutí rotoru a tím se v rotoru vytvoří magnetické pole. Na vodiči s proudem v magnetickém poli působí Lorentzova magnetická síla, která vyvolá točivý moment. Tento moment způsobí otáčivý pohyb rotoru (viz Obr. 1)

Důležitou součástí rotoru je komutátor, který zajišťuje změnu průtoku proudu ve vinutí rotoru. Tím se vytváří točivý moment, který na rotor působí pouze v jednom směru a tím vzniká otáčivý pohyb.



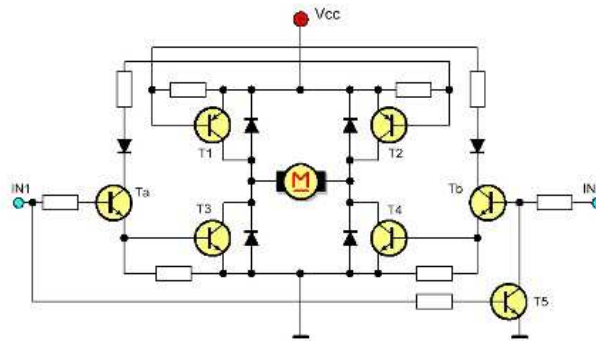
Obr. 1 Schéma stejnosměrného elektromotoru [20]

Přepólováním napájecího napětí rotoru lze měnit směr otáčení stejnosměrného elektromotoru. Změnou velikosti elektrického napětí v rotoru lze měnit rychlost otáčení stejnosměrného elektromotoru.

3.1.2 Řízení stejnosměrných motorů

Při řízení stejnosměrného motoru mikrokontrolérem není možné, aby byl motor napřímo připojený k mikrokontroléru, protože maximální proudový odběr, který výstupní piny mikrokontroléru nabízejí, není dostačující pro stejnosměrné motory, a tudíž by došlo ke zničení mikrokontroléru. Je tedy nutné zařadit mezi motor a mikrokontrolér spínací tranzistory, které tento problém vyřeší. Navíc, aby bylo možné stejnosměrný motor otáčet oběma směry, je nutné pro jeho řízení z mikrokontroléru použít např. H-můstek.

H-můstek není nic jiného, než jen čtyři tranzistory zapojené do tvaru písmene H. Na každý vývod motoru jsou dva tranzistory. Platí, že vždy v každé této dvojici musejí mít tranzistory opačnou logickou úroveň. Proto je vždy do druhého tranzistoru z každé dvojice přiváděn negovaný vstupní signál prvního tranzistoru. Tím vzniká pro každou stranu pouze jeden řídicí signál. Jakým směrem se bude motorek otáčet je pak dáno právě kombinací řídicího signálu. 00=neutrál, 01=vpřed, 10=vzad, 11=stop. Konkrétní příklad zapojení H-můstku je ukázán na Obr. 2.



Obr. 2 Schéma zapojení H-můstku [22]

Vstupy IN1 a IN2 se řídí, která dvojice tranzistorů (T1 a T4, nebo T2 a T3) se otevře, a tím se motor pohybuje doleva, nebo doprava. Tranzistory Ta a Tb jsou pouze řídicí pro každou dvojici tranzistorů v H-můstku. Z obrázku je dále vidět, že pokud je napětí přivedeno na oba vstupy (IN1 a IN2), otevrou se všechny čtyři tranzistory (T1, T2, T3 a T4) a přes ně se zkratuje napájecí zdroj. Aby se tomuto předešlo, je v zapojení ještě zařazen tranzistor T5, který chrání napájecí zdroj před zkratováním. Pokud je přivedeno napětí na vstup IN1, otevře se tranzistor Ta, T2 a T3 a také tranzistor T5, který spojí bázi tranzistoru Tb se zemním potenciálem. Pokud je tedy nyní přivedeno napětí na vstup IN2, báze tranzistoru Tb je blokována tranzistorem T5 a veškerý proud ze vstupu IN2 tak teče přes bázevý rezistor tranzistoru Tb a dále přes tranzistor T5 do země.

3.1.3 Regulace otáček stejnosměrných motorů

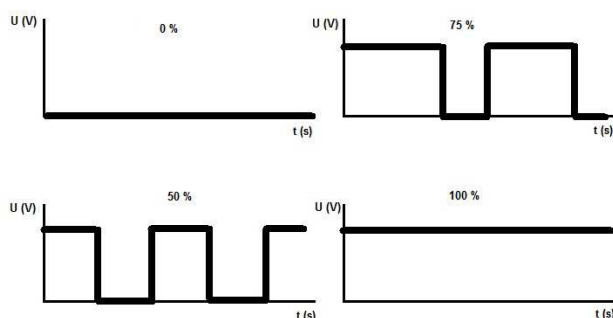
Způsobů jak regulovat otáčky stejnosměrného motoru je hned několik. Já se ovšem v následujících řádcích zaměřím především na regulaci otáček pomocí PWM modulace, protože právě tento způsob regulace budu později využívat při řešení funkčního zařízení.

Použití změny přiloženého napětí

Nejjednodušší způsob regulace otáček stejnosměrného motoru lze provádět změnou přiloženého napětí na vývody motoru (na kartáče rotoru), kde se následně mění velikost proudu na vinutí rotoru a tím dochází ke změně rychlosti otáčení stejnosměrného motoru.

Použití PWM modulace

Velmi využívaný způsob regulace otáček stejnosměrného motoru je použití PWM modulace. Výhodou této regulace je vysoká účinnost, protože zde nedochází ke ztrátám na regulačním obvodu. Podstatou PWM regulace je periodické spínání napětí s danou střídou (poměr času, kdy je napětí vypnuté a zapnuté). U PWM modulace tedy mění šířku impulsu. Kmitočet a amplituda mohou zůstat nezměněny. PWM vychází z principu, že stejnosměrný motor (popř. LED, žárovka apod.) pracuje u časově proměnného průběhu s jeho střední hodnotou (průměr za periodu). Časový průběh však musí mít dostatečně vysoký kmitočet.



Obr. 3 Příklady průběhů PWM modulace [10]

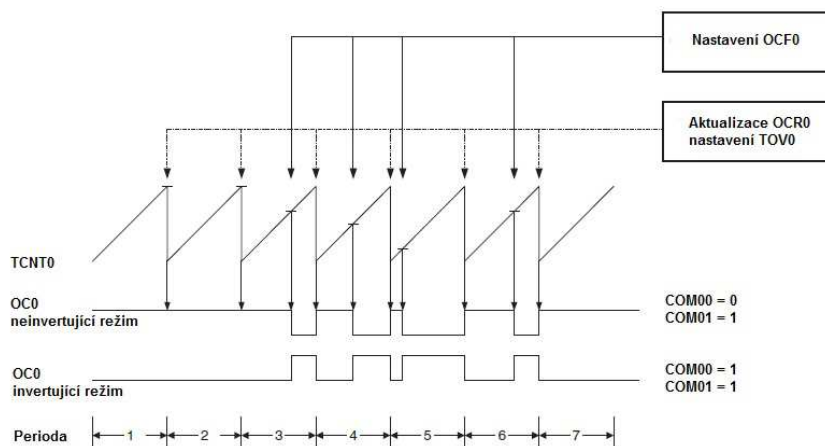
Obr. 3 ukazuje různé periody spínání napětí, kde 0 % znamená vypnutý stav, 50 a 75 % spínací stav a 100 % stále sepnutý stav. Z obrázku je také patrné, že napětí a tím i proud do motoru má stále stejnou hodnotu, ale mění se jeho aktivní doba, kdy motorem prochází. Právě díky stále stejné hodnotě proudu zůstává v celém rozsahu otáček stále stejná síla motoru.

3.1.4 PWM modulace u mikrokontroléru ATmega32

Mikrokontrolér ATmega32 obsahuje dva osmibitové čítače/časovače a jeden šestnáctibitový čítač/časovač. Protože práce s jednotlivými čítači/časovači je velmi obdobná, budu se dále zabývat pouze popisem čítače/časovače 0.

Osmibitový čítač/časovač 0 obsahuje desetibitovou předděličku hodinového taktu mikrokontroléru. Čítač/časovač 0 může sloužit k vyvolání přerušení, jakmile dojde k přetečení hodnoty čítače, nebo jakmile dojde ke shodě registru TCNT0 s nastavenou hodnotou registru OCR0. Další možností, jak využít čítač/časovač 0, je funkce PWM generátoru, který lze využít ve dvou pracovních režimech. Rychlý režim PWM, nebo fázově korigovaný režim.

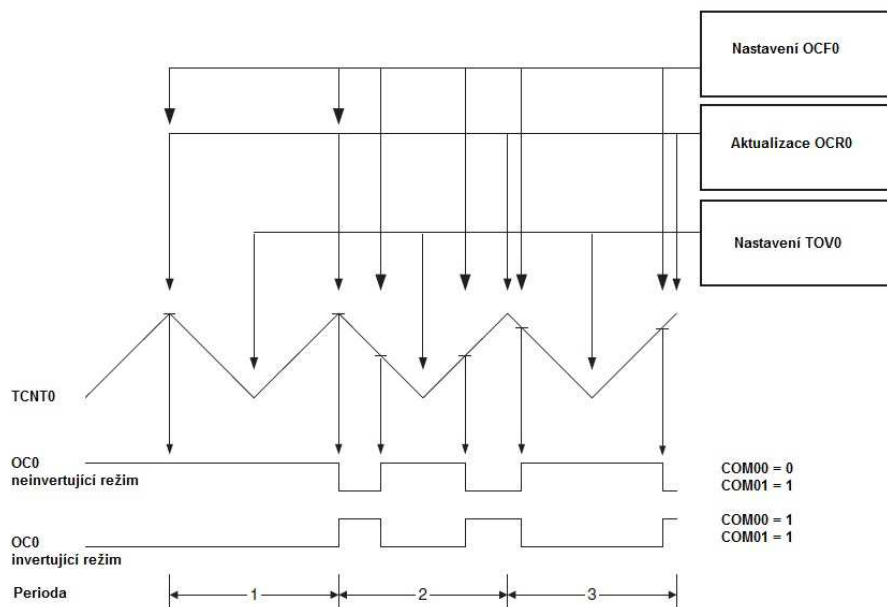
Rychlý režim PWM



Obr. 4 Časový diagram režimu rychlého PWM [12]

Na Obr. 4 je znázorněn princip činnosti rychlého režimu PWM, který pracuje s dvojnásobnou pracovní frekvencí oproti fázově korigovanému režimu. Princip tohoto režimu spočívá v čítání registru TCNT0 nahoru až dosáhne maxima 0xFF (hodnota „přeteče“) a poté se opět vrátí zpět do nuly. Při dosažení nuly se nastaví příznak přetečení TOV0. Zároveň se během načítání porovnává registr TCNT0 s registrem OCR0. Jakmile dojde ke shodě mezi těmito dvěma registry, nastaví se příznakový bit OCF0, který způsobí přenastavení výstupního stavu OC0 dle nastavených bitů COM00 a COM01 v registru TCCR0. Rychlý režim PWM se vybírá nastavením bitů WGM00 = 1 a WGM01 = 1 v registru TCCR0.

Fázově korigovaný režim PWM



Obr. 5 Časový diagram fázově korigovaného režimu PWM [12]

Na Obr. 5 je znázorněn princip činnosti fázově korigovaného režimu PWM, který se oproti rychlému režimu PWM liší v tom, že registr TCNT0 čítá nahoru opět až do maxima (hodnota 0xFF), ale po dosažení maximální hodnoty (hodnota „přetečení“) se nenastaví na nulu, ale registr čítá dolů. Výhodou tohoto režimu je vyšší rozlišení a to i přesto, že má nižší pracovní kmitočet oproti rychlému režimu PWM. Také v tomto režimu dochází k porovnávání registru TCNT0 s registrem OCR0. Zde však k porovnávání těchto dvou registrů dochází jak při čítání nahoru, tak při čítání dolů. Jakmile dojde ke shodě registru TCNT0 s registrem OCR0, přenastaví se výstupní stav OC0 dle nastavených bitů COM00 a COM01 v registru TCCR0. Příznak přetečení TOV0 se v tomto režimu aktivuje při dosažení nuly registrem TCNT0. Fázově korigovaný režim PWM se vybírá nastavením bitů WGM00 = 1 a WGM01 = 0 v registru TCCR0.

3.2 Mechanismus přenosu momentu síly

Síla může mít otáčivé účinky, a proto může vyvolat, zbrzdit, zastavit, nebo měnit otáčivý pohyb tělesa.

Moment síly

Otáčivý účinek síly závisí na dvou aspektech. Prvním je velikost síly a druhým je vzdálenost od osy otáčení. Aby bylo možné otáčivý účinek síly popsat, je nutné použít vektorovou fyzikální veličinu, moment síly: [1]

$$\vec{M} = \vec{F} \times \vec{r} \quad (1)$$

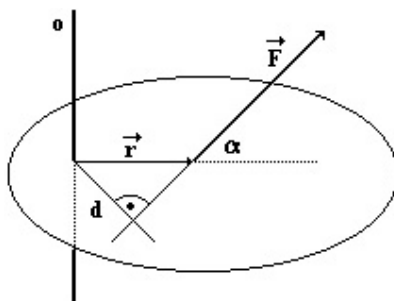
$$|\vec{M}| = |\vec{F}| \cdot |\vec{r}| \cdot \sin \alpha \quad (2)$$

\vec{M} = vektor momentu síly (N·m),

\vec{F} = vektor působící síly (N),

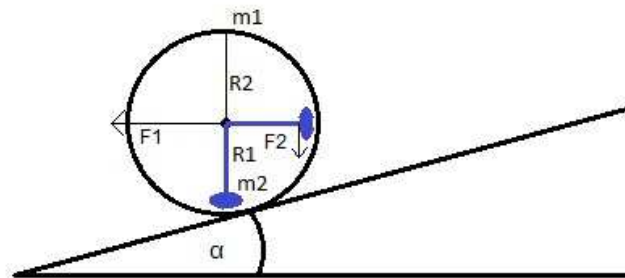
\vec{r} = vektor vzdálenosti od osy otáčení (m),

α = úhel mezi vektory F, r (°).



Obr. 6 Moment síly [1]

Odvození pro maximální úhel stoupání



Obr. 7 Příklad maximálního úhlu stoupání

$$\vec{M}_1 = \vec{M}_2 \quad (3)$$

$$M_1 = R_1 \cdot m_1 \cdot g \cdot \sin \alpha \quad (4)$$

$$M_2 = R_2 \cdot m_2 \cdot g \quad (5)$$

$$\alpha = \sin^{-1} \left(\frac{R_2 \cdot m_2}{R_1 \cdot m_1} \right) \quad (6)$$

Moment M_1 je vyvolaný silou F_1 , která působí na těleso na nakloněné rovině, a poloměrem R_1 . Pro maximální úhel stoupání α musí platit, že je tento moment vyvážen maximálním možným momentem M_2 , tedy situací, kdy je závaží m_2 nachýleno do úhlu 90 stupňů při poloměru R_2 .

V rovnici (5) je sinus α roven jedné, protože se předpokládá, že je závaží vychýleno do pravého úhlu.

\vec{M}_1 = moment síly vyvolaný celkovou hmotností zařízení m_1 , úhlem náklonu roviny α a poloměrem modulu R_1 (N·m),

\vec{M}_2 = moment síly vyvolaný hmotností závaží m_2 a délkou ramene pro závaží R_2 (N·m),

R_1 = poloměr modulu resp. trubky (cm),

R_2 = délka ramene pro závaží (cm),

m_1 = hmotnost celého modulu (g),

m_2 = hmotnost samotného závaží (g),

g = tíhové zrychlení (m/s^2),

α = maximální úhel stoupání ($^\circ$).

Z rovnice je patrné, že čím více rovné si budou hmotnosti m_1 a m_2 , tím větší bude úhel stoupání α , protože podíl hmotností se bude blížit k jedné. Totéž platí pro podíl mezi délkou ramene pro závaží R2 a délkou poloměru trubky R1.

Pokud by délka ramene pro závaží byla stejná jako poloměr trubky a hmotnosti m_1 a m_2 by se rovnaly, byla by hodnota úhlu stoupání α 90 °.

Lze tedy prohlásit, že čím těžší je hmotnost závaží m_2 a lehčí hmotnost samotného modulu bez závaží, tím se maximální úhel stoupání α zvětší. Pro délky R1 a R2 platí, že čím větší bude poloměr trubky R1, tím bude větší podíl délek R1 a R2 a tím se zvětší maximální úhel stoupání α .

3.3 Snímání rychlosti otáček pomocí MEMS gyroskopu

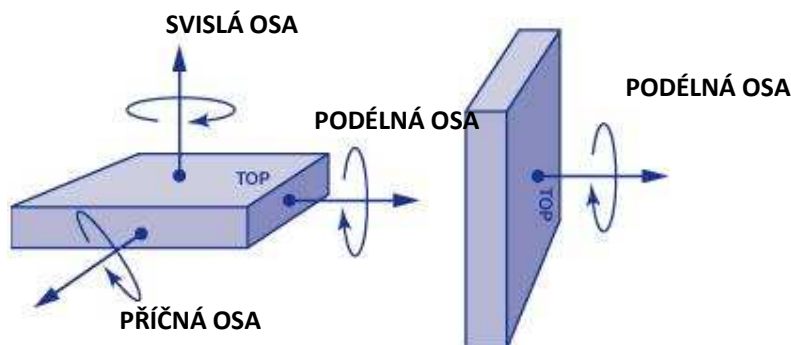
V této části mé práce se budu zabývat MEMS gyroskopy, protože právě tento druh senzoru budu později využívat pro své funkční zařízení.

V dnešní době je výhodné, pro snímání rychlosti otáček, využívat MEMS gyroskopy, protože jejich malé rozměry jsou z hlediska minimalizace rozměrů všemožných elektronických zařízení velmi výhodné.

Gyroskop je senzor, který je nejvíce známý jako senzor pro měření polohy nebo rotace libovolného předmětu, ke kterému je gyroskop připevněný. V dnešní době jsou hojně vyráběny integrované obvody, díky kterým lze vytvářet složitější funkce z několika jednoduchých elektrických součástí, které vytvářejí elektrický obvod v jednom malém pouzdře. Samozřejmě se nevyrábí integrované obvody, které obsahují pouze elektrické obvody. Podstatou MEMS integrovaných obvodů jsou různé mechanické struktury, které jsou v jednom integrovaném pouzdře společně s celou škálou vyhodnocovacích obvodů a logiky. Výstupem je pak digitální nebo analogová veličina. U modernějších MEMS integrovaných obvodů si lze vybrat, zda bude využit analogový, nebo digitální výstup. Možná nejznámější aplikací, kde jsou gyroskopy hojně využívány, je sledování pohybu objektů přes GPS.

3.3.1 Princip činnosti gyroskopu

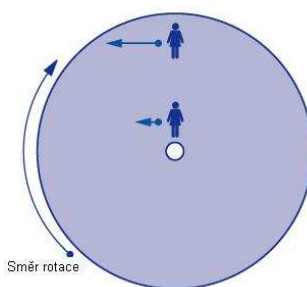
Obecně gyroskopy slouží pro měření úhlové rychlosti daného objektu. Přesněji, jak rychle se objekt otáčí v jednotkách stupňů/sekundu. Rotaci objektu lze měřit vzhledem k jedné ze tří os x, y, z. Tyto osy lze označovat jako svislá osa, příčná osa a podélná osa (viz Obr. 8). Integrované MEMS gyroskopy pracují na principu Coriolisovy síly.



Obr. 8 Náhled na osy gyroskopu [19]

3.3.2 Coriolisova síla

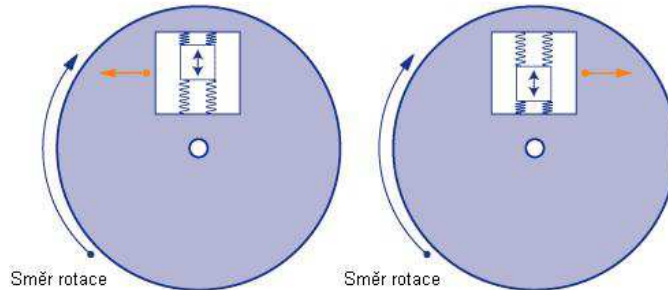
Je síla, která působí na každé těleso, které se volně pohybuje v rotující soustavě. V soustavě otáčející se ve směru hodinových ručiček se pohybující těleso vychyluje vlevo, pokud se těleso pohybuje od středu k okraji rotující soustavy. Bude-li se těleso v rotující soustavě pohybovat od okraje rotující soustavy k jejímu středu, je působení Coriolisovy síly na pohybující se těleso přesně opačné, tedy těleso se bude vychylovat vpravo. (viz Obr. 9).



Obr. 9 Příklad působení Coriolisovy síly (modrá šipka) na osobu pohybující se od středu k okraji rotujícího kruhu [19]

Tohoto účinku se využívá v mechanických gyroskopech, kde se periodicky pohybuje objekt upevněný na pružinách uvnitř rámu mechanické soustavy směrem od středu k okraji rotující soustavy a naopak a podle směru pohybu tělesa na něho působí Coriolisova síla (viz Obr. 10).

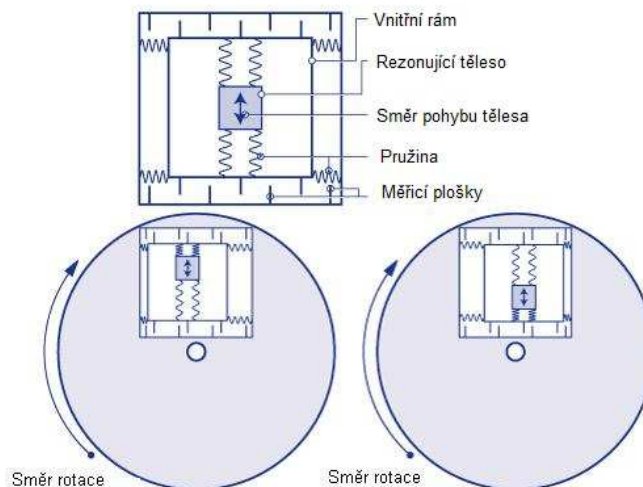
Protože je velikost a směr působící Coriolisovy síly na těleso úměrný i rychlosti a směru otáčení, lze tento mechanický systém použít pro měření úhlové rychlosti.



Obr. 10 Působení Coriolisovy síly na mechanický gyroskop tvořený hmotným objektem periodicky se pohybující (kmitající) k okraji kruhu (vlevo) a ke středu (vpravo) [19]

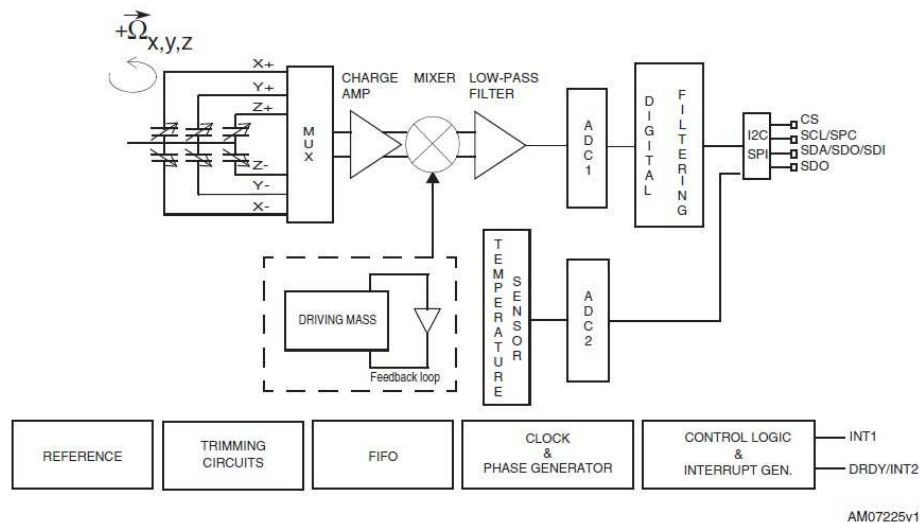
3.3.3 MEMS gyroskop

V integrovaných gyroskopech se při použití Coriolisovy síly využívá technologie MEMS, kde jsou hlavní součástí mechanické mikrosoučásti spolu s elektrickými obvody a tím je vytvořen mikrosnímač. Hlavním prvkem je pohybující se část, která mechanicky rezonuje v rámu pomocí pružin. Směr rezonujícího pohybu musí být kolmý ke směru rotace, aby se projevila Coriolisova síla, jejíž velikost je úměrná úhlové rychlosti otáčení. Tato síla způsobuje stlačení vnějších pružin a tím vzájemný posun měřicích plošek, které zde fungují jako elektrody vzduchového kondenzátoru. (viz Obr. 11). Dochází tak ke změně kapacity, která je přímo úměrná úhlové rychlosti otáčení v jednotkách stupňů za sekundu. Tato veličina je pak výstupem celého senzoru.



Obr. 11 Zjednodušená struktura snímače MEMS gyroskopu a příklad funkce struktury snímače gyroskopu při rotaci [19]

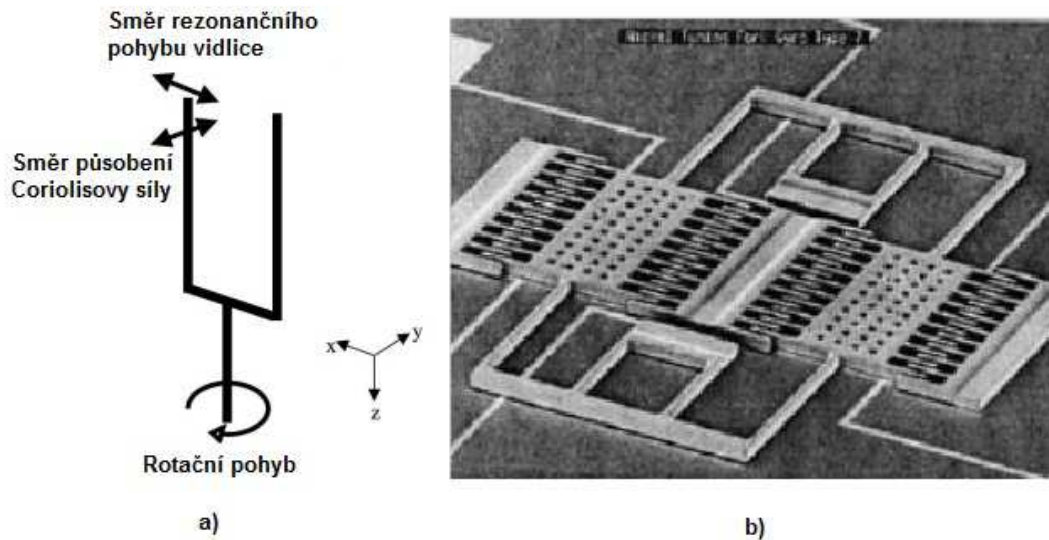
Na Obr. 12 je znázorněné blokové schéma MEMS gyroskopu. Je patrné, že tento gyroskop obsahuje dva hlavní bloky. Prvním je samotná mechanická struktura s pohyblivými elektrodami a logikou. Druhým blokem je senzor teploty, který je u některých MEMS gyroskopů součástí integrovaného pouzdra. Není zde ovšem žádná podmínka využívat tu či onu část. Lze využívat obojí najednou, nebo jen jednu z nich. Pokud se zaměřím na výstupy blokového schématu, je jich celkem šest. Krom napájecích vývodů, které nejsou v blokovém schématu zobrazeny, jsou zde čtyři výstupy, které slouží pro komunikaci I2C, nebo SPI. Vývod CS slouží pro výběr komunikace mezi gyroskopem a mikrokontrolérem. Pokud je vývod CS nastaven na log. 1, povoluje se tím I2C komunikace. V opačném případě (CS = log. 0) je komunikace I2C zakázána. Pokud je zvolena I2C komunikace, jsou dále využívány vývody SDA (datový vodič I2C komunikace) a SCL (vodič hodinového signálu). Pro komunikaci SPI jsou využívány také tyto dva vývody, ale nyní jako SPC, SDO či SDI. Spodní vývod SDO slouží jako signifikační bit pro identifikaci Slave zařízení. Poslední dva vývody INT1 a INT2 slouží pro přerušení.



Obr. 12 Blokové schéma MEMS gyroskopu [13]

Tuning fork gyroskop

Jedna z nejvíce využívaných struktur MEMS gyroskopu je Tuning fork gyroskop (viz Obr. 13), která pracuje na principu rezonující „vidlice“ v ose x dle dané amplitudy. Pokud se celý systém začne otáčet kolem osy Z, vzniklá Coriolisova síla zde začne vytvářet sílu působící na „vidlici“ ve směru osy Y. Tato vzniklá síla je pak úměrná vyvinuté úhlové rychlosti senzoru.

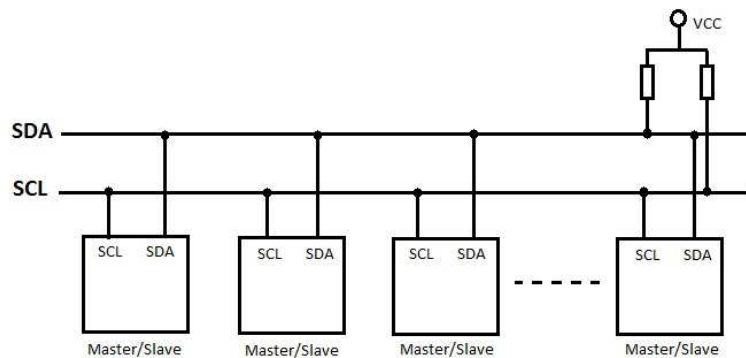


Obr. 13 Tuning fork gyroskop; a) fyzikální princip; b) struktura [11]

I2C komunikace

V předchozí kapitole jsem zmínil I2C komunikaci, která se pro komunikaci mezi gyroskopem a mikrokontrolérem využívá. Proto bych nyní rád lehce popsal I2C komunikaci a její funkčnost.

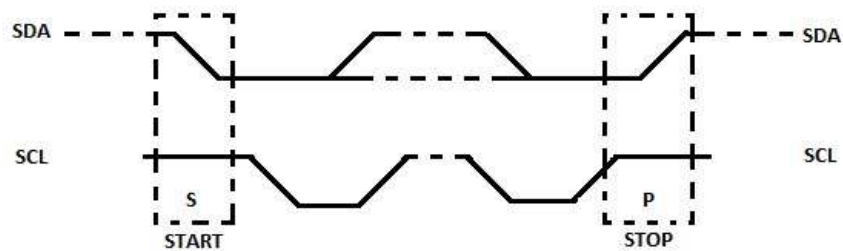
I2C komunikace je na rozdíl od SPI sběrnice typu multimaster. Může tedy obsahovat více než jedno Master zařízení. Každá stanice má svou vlastní adresu, která slouží k jejímu výběru. Komunikace pro I2C sběrnici je znázorněna na Obr. 14.



Obr. 14 Princip I2C komunikace [2]

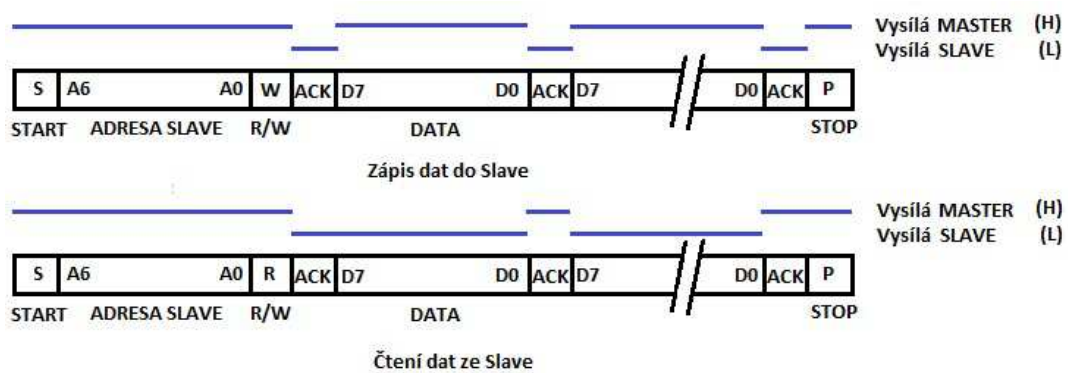
Z obrázku je vidět, že I2C komunikace používá pouze dva vodiče. Vodič SDA slouží pro přenos datového signálu a vodič SCL slouží pro přenos hodinového signálu. Jednotlivé stanice v I2C rozhraní jsou tak propojeny pouze za pomoci jednoho datového a jednoho hodinového vodiče.

V klidovém stavu jsou oba vodiče v logické úrovni high (log. 1). Při probíhající komunikaci jsou na SDA vodič postupně vysílány jednotlivé datové bity, přičemž platí, že logická úroveň na SDA se smí měnit pouze tehdy, je-li vodič SCL v logické úrovni low (log. 0) viz Obr. 15.



Obr. 15 I2C komunikace - START a STOP [2]

Každá stanice připojená na I2C sběrnici má přidělenou vlastní adresu. Po zachycení podmínky START porovnávají všechny obvody svou adresu s adresou, která je vysílána na sběrnici. Zjistí-li některá stanice shodu, je vysílání určeno právě této stanici a musí přijetí adresy potvrdit bitem ACK. Po tomto potvrzení může přijímat resp. vysílat další data (viz Obr. 16).



Obr. 16 Průběh vysílání po I2C sběrnici [2]

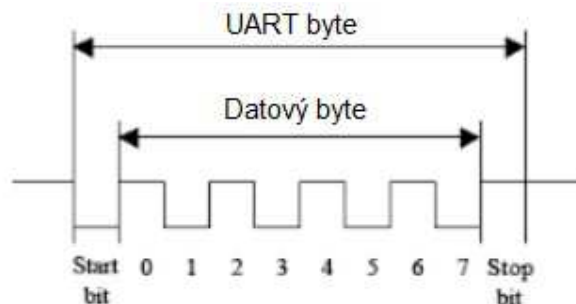
3.4 Sériové komunikační rozhraní

Při připojování různých zařízení s jednočipem nebo procesorem můžeme využít nejběžnějšího rozhraní UART. Tedy sériové linky, která se ve většině PC ukazuje jako COM port. Takto propojená zařízení mají ovšem jeden zásadní problém. Zařízení musí být propojena pomocí kabelu, což není vždy zcela výhodné. Tím se dostávám k jádru problematiky, kterou chci na dalších řádcích přiblížit o něco více. Pokud je potřeba propojit nějaké zařízení například s počítačem přes sériovou linku UART, nemusím nutně využívat kabelové spojení. Pro bezdrátové propojení různých zařízení s jednočipem či procesorem je k dispozici tzv. bluetooth. Pomocí bluetooth je možné připojit jakékoli zařízení s UART komunikací přes virtuální COM port. Pokud tedy nahradíme sériový kabel bluetooth modulem na přijímací i vysílací straně, lze vytvořit bezdrátovou komunikaci po sériové lince UART.

3.4.1 Sériová UART linka

UART je zkratka Universal Asynchronous Receiver Transmitter, což v překladu znamená univerzální asynchronní sériový přijímač a vysílač. Jde o nejčastěji používané sériové rozhraní, které umožňuje obousměrnou komunikaci. [9]

UART vysílá data na vývodu označovaném jako TX (Transmit), přijímá na vývodu RX (Receive). Klidová úroveň signálu (stav kdy neprobíhá příjem ani vysílání dat) je log. 1. Při vysílání se klidová úroveň signálu přepne na úroveň log. 0 po dobu jednoho bitu. Tento bit se nazývá Start-bit. Následně je vyslán datový bit s nejnižší prioritou. Jako poslední je vyslán nejvýznamnější datový bit, za kterým následuje Stop-bit, který má opět logickou úroveň log. 1 (viz Obr. 17). Jakmile je odvysílání ukončeno Stop-bitem, může začít odvysílání dalšího bytu.



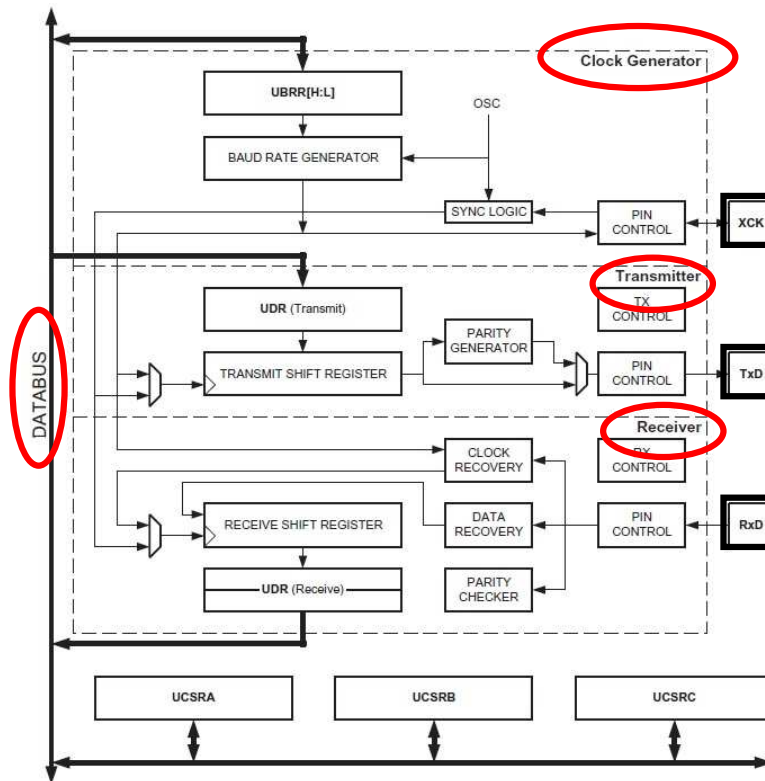
Obr. 17 Asynchronní osmibitový přenos [9]

3.4.2 Modul USART u mikrokontroléru ATmega32

V této části lehce popíšu blokové schéma jednotky UART u mikrokontroléru ATmega32, protože právě tento mikrokontrolér později využiji v realizaci funkčního zařízení.

Na začátku bych rád upřesnil, že mikrokontrolér ATmega32 neobsahuje jednotku UART, ale USART. Tedy Universal Synchronous and Asynchronous Receiver Transmitter, což lze přeložit jako univerzální synchronní a asynchronní sériový přijímač a vysílač. Asynchronní část rozhraní USART se nazývá UART. Lze si tedy zvolit, zda bude využit synchronní režim (USRT), nebo asynchronní režim (UART).

Na Obr. 18 je znázorněno blokové schéma jednotky USART pro mikrokontrolér ATmega32 AVR. Z obrázku je patrné, že jednotka USART obsahuje tři hlavní části, které jsou v obrázku vyznačeny červeně. První částí je Clock Generator (generátor hodinového signálu), následuje Transmitter (vysílač) a poslední částí je Receiver (přijímač). Všechny tyto tři hlavní části jsou propojené s DATABUS (datová sběrnice). Z obrázku je dále vidět, že s datovou sběrnicí jsou propojené i tři řídicí registry UCSRA, UCSRB a UCSRC. První blok, generátor hodin, slouží pro příjem externích hodin, ale pouze v případě, že je použit synchronní přenos. V opačném případě, při asynchronním režimu, není hodinový signál vysílán. Blok generátor hodinového signálu dále obsahuje blok BAUD RATE GENERATOR (generátor přenosové rychlosti), který slouží pro generování přenosové rychlosti. Tučně vyznačené bloky TxD, RxD a XCK jsou přístupné na pinech mikrokontroléru.



Obr. 18 Blokové schéma modulu USART pro mikrokontroler ATmega32 [12]

Piny TxD a RxD slouží pro obousměrnou komunikaci. Pin TxD slouží pro vysílání dat a vývod RxD pro příjem dat.

Z výše uvedených vět vyplývá, že mikrokontrolér ATmega32 nabízí při USART komunikaci dva režimy – Asynchronní a Synchronní.

- Asynchronní režim – označován jako UART lze také nazvat plný duplex. Tzn., že obousměrná komunikace může probíhat současně. V tomto režimu, kde generátor hodinového signálu nevysílá žádný signál, je nutné, aby obě zařízení komunikující mezi sebou, měla nastavenou stejnou přenosovou rychlost.
- Synchronní režim – označován jako USRT lze také nazvat poloviční duplex. Tzn., že obousměrná komunikace nemůže probíhat současně. V tomto režimu je využit generátor hodinového signálu, který je přijímán mikrokontrolérem, který je nastaven jako Synchronní-Slave. V režimu Synchronní-Master, mikrokontrolér vysílá data.

4. Řešení funkčního zařízení

4.1 Úvod

Cílem práce je sestavit pohyblivý modul na základě přenosu momentu síly. Při návrhu jsem tedy musel velmi pečlivě zvážit, jak tento princip pohybu použít a jak zkonstruovat modul, který bude princip přenosu momentu síly využívat ke svému pohybu. Nakonec jsem došel k závěru, že hlavní rotující částí bude plastová trubka o průměru 10 cm, ve které bude umístěn stejnosměrný motor, řídicí elektronika a závaží. Celé zařízení se tak bude pohybovat v jedné ose. Tento princip pohybu je ovšem možné využít i pro dvouosý pohyb.

Takto realizovaný modul bude fungovat tak, že stejnosměrný motor umístěn přesně v ose plastové trubky bude otáčet stator, kterým je plášť motoru a plastová trubka s veškerou elektronikou včetně baterie. Rotor motoru bude pouze vychylovat závaží do vodorovné polohy. Elektronika v pohyblivém modulu bude snímat rychlost otáčení modulu a na základě požadavků uživatele bude tuto rychlost regulovat tak, aby byla rychlost modulu totožná se zadanou rychlostí od uživatele. Hodnotu požadované rychlosti zadá uživatel pomocí počítače.

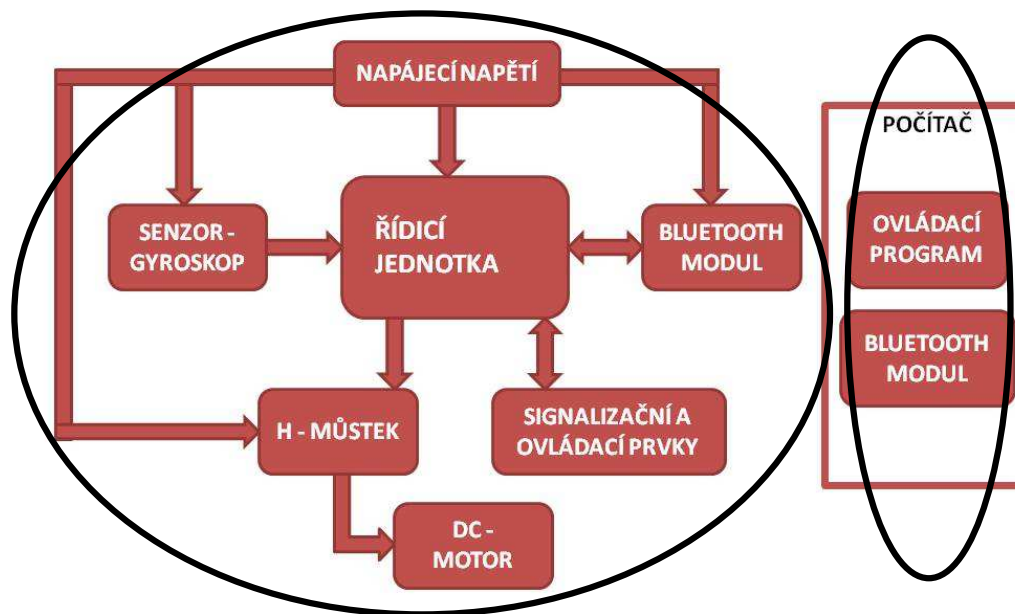
Aby bylo ovládání modulu co nejvíce snadné a efektivní, vytvořím v počítači ovládací program, který bude sloužit jako uživatelské rozhraní pro řízení celého modulu. Ovládací program bude obsahovat tlačítka pro možnost volby pohybu modulu dopředu i dozadu a také okamžité zastavení. Další částí ovládacího programu budou dvě okénka, kde první bude sloužit pro zadávání rychlosti otáčení modulu a druhé pro zobrazení aktuální rychlosti modulu. Důležitým blokem ovládacího programu budou tlačítka pro párování bezdrátového bluetooth modulu s počítačem.

4.2 Řešení - hardware

Na začátku tohoto projektu jsem uvedl teoretické rozборы jednotlivých oblastí, které budu nyní prakticky využívat k sestavení pohyblivého modulu.

V následující části se budu zabývat konkrétním návrhem zařízení, konstrukcí a vlastnostmi jednotlivých částí.

Z Obr. 19 je vidět obecný návrh zařízení, které se skládá ze dvou hlavních částí. První částí je elektronika pro pohyblivý modul a druhou částí je ovládací program v počítači.



Obr. 19 Blokové schéma - obecný návrh

Elektronika pro pohyblivý modul se skládá ze sedmi funkčních bloků. Řízení modulu je zajištěno blokem ŘÍDICÍ JEDNOTKA, který slouží pro vyhodnocení dat získaných z bloku SENZOR-GYROSKOP a na základě těchto dat odesílá data do bloku H-MŮSTEK, který řídí blok DC-MOTOR. Tento blok je hlavní částí pohybového aparátu v modulu. Blok SIGNALIZAČNÍ A OVLÁDACÍ PRVKY je v modulu umístěn pouze pro kontrolu funkčnosti některých částí modulu. Důležitým blokem je BLUETOOTH MODUL, díky kterému lze realizovat bezdrátové řízení modulu a odesílání dat do ovládacího programu a zpět.

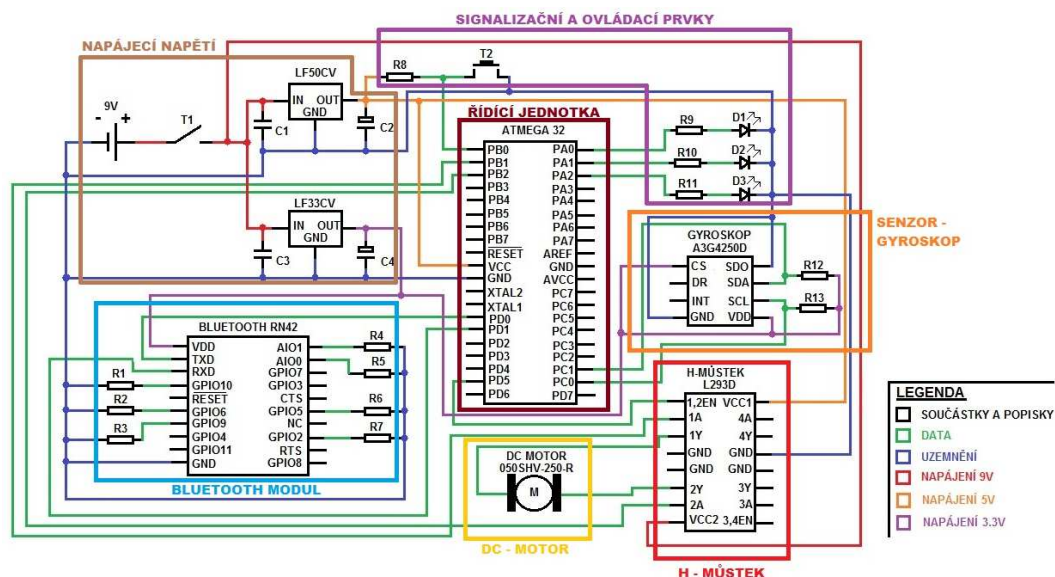
Aby mohla být celá řídicí část funkční, je třeba bloku NAPÁJECÍ NAPĚTÍ, který zajišťuje elektrickou energii pro celou elektroniku pohyblivého modulu.

Dále budu popisovat funkční celky jednotlivých bloků obecného návrhu.

4.3 Elektronika pro pohyblivý modul

Celkové schéma zapojení elektroniky pro pohyblivý modul je znázorněno na Obr. 20.

Aby bylo lépe vidět porovnání celkového zapojení elektroniky pro pohyblivý modul s obecným návrhem z Obr. 19, jsou jednotlivé bloky vyznačeny barevně právě podle bloků z obecného návrhu.



Obr. 20 Celkové schéma elektroniky pro pohyblivý modul

4.3.1 Řídicí jednotka

Řídicí jednotkou celého zařízení je osmibitový RISC mikrokontrolér ATmega32 od firmy Atmel AVR. Hlavním kritériem pro výběr tohoto mikrokontroléru byla jeho velká paměť a velké množství I/O pinů, které jsem při počátečním návrhu zařízení potřeboval pro připojení LCD displeje. Protože i pro finální návrh jsem využil větší množství I/O pinů a nepotřeboval jsem omezit velikost čipu, vybral jsem tento mikrokontrolér i pro finální zařízení.

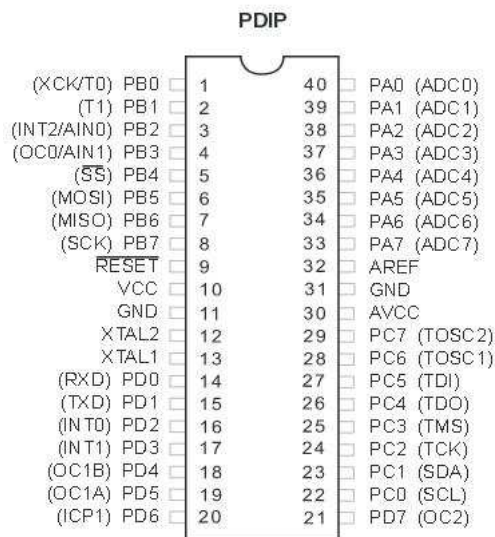
Přehled základních vlastností mikrokontroléru ATmega32 je uveden v Tab. 1.

Tab. 1 Základní parametry ATmega32 [12]

Počet instrukcí	131
Počet osmibitových	32
Počet I/O pinů	32
Hodinový kmitočet	0 – 16 MHz
Maximální výkon	16 MIPS
Kapacita FLASH	32 kB
Kapacita SRAM	2 kB
Kapacita EEPROM	1 kB
Analogový komparátor	Ano
A/D převodník	Ano
USART	Ano
PWM kanály	Ano
Power - on reset	Ano
Zabudovaný RC oscilátor	Ano
Aktivní stav	1 MHz, 3 V: 1,1 mA
Klidový stav	<0,1 μ A: 3 V

Rychlost	Napájecí napětí
0 – 16 MHz	4,5 – 5,5 V

Mikrokontrolér ATmega32 (viz Obr. 21) disponuje velkým množstvím I/O pinů. Této skutečnosti jsem využil zejména tehdy, když jsem potřeboval mít čip zapojený jak pro řízení pohyblivého modulu, tak pro naprogramování.



Obr. 21 Rozložení vývodů mikrokontroléru ATmega32 na pouzdře PDIP [12]

Pro naprogramování mikrokontroléru jsem použil Programátor AVR ATK500 v2 .

Aby bylo možné nahrát do mikrokontroléru řídicí program pomocí programátoru, je zapotřebí 6 pinů. První dva piny VCC a GND slouží pro samotné napájení programátoru. Dále je zapotřebí pinů PB5 až PB7 a pinu RESET. Takto zapojený mikrokontrolér k programátoru je vhodný pro sériový download¹.

Po připojení programátoru k řídicímu mikrokontroléru jsem dále využil další piny pro řízení celého zařízení. Jednotlivé piny, které jsem potřeboval využít pro funkčnost zařízení, jsem zapojil následujícím způsobem:

- Piny PA0 až PA2 jsem použil pro připojení signalizačních prvků (LED diody).
- Pin PBO je použit pro připojení testovacího tlačítka.
- Piny PB1 a PB2 slouží pro řízení H-můstku, který následně ovládá směr otáčení stejnosměrného motoru.
- Piny PC0 a PC1 slouží pro digitální komunikaci po I2C sběrnici mezi mikrokontrolérem a gyroskopem. Jsou tedy využity jako SDA a SCL vývody.
- Piny PD0 a PD1 jsou využity pro sériovou komunikaci jako RX a TX vývody, které jsou propojené s bluetooth modulem.
- Pin PD5 slouží pro PWM signál, kterým je přiveden do H-můstku a následně je jím řízena rychlost otáčení stejnosměrného motoru.

Mikrokontrolér ATmega32 obsahuje několik vnitřních periférií. Ve svém zařízení jsem několik těchto vnitřních periférií využil. První z nich je komunikační rozhraní USART, dále PWM signál generovaný z čítače/časovače 1 a nakonec digitální I2C komunikace.

Komunikační rozhraní USART využívám pro připojení bluetooth modulu s mikrokontrolérem, aby bylo možné provádět bezdrátovou sériovou komunikaci. Periférii PWM signál využívám pro řízení otáček stejnosměrného motoru a konečně I2C komunikaci využívám pro připojení gyroskopu k mikrokontroléru, protože gyroskop je využit v digitálním režimu.

¹ Jedná se o programovací metodu, při které je možné využívat mikrokontrolér pro aplikaci a zároveň pro naprogramování. Není tedy nutné mikrokontrolér odpojovat od aplikační desky.

4.3.2 Senzor - gyroskop

Ke snímání rotace jsem použil MEMS gyroskop A3G4250D, jehož základní parametry popisuje Tab. 2.

Tab. 2 Základní parametry gyroskopu A3G4250D [13]

Provozní napětí	2,4 – 3,6 V
Provozní proud	6,1 mA
Proud ve sleep módu	1,5 mA
Klidový proud	5 – 10 μ A
Teplotní rozsah	-40 až +85°C

Rozsah úhlové rychlosti, kterou je tento gyroskop schopen snímat je ± 245 °/s a protože výstupní data gyroskopu jsou šestnáctibitová, je nutné si uvědomit, jak bude potřeba podělit výstupní hodnotu gyroskopu, aby skutečně odpovídala reálné úhlové rychlosti. Takový přepoččet bude vypadat následovně:

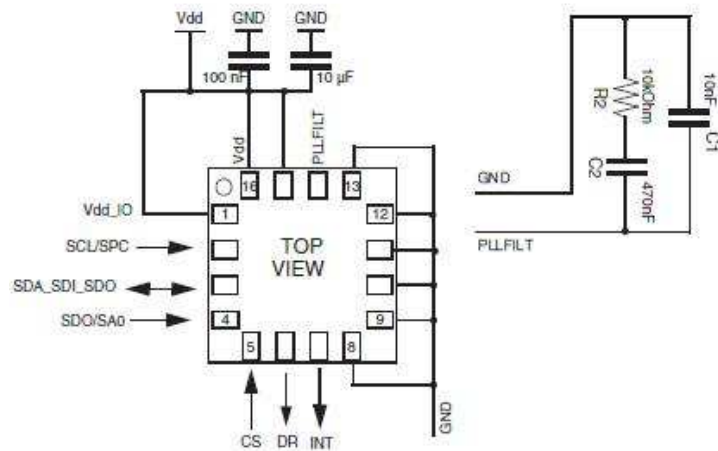
$$\text{Dělicí poměr} = \frac{\text{Max. hodnota šestnáctibitového čísla}}{\text{Rozlišení gyroskopu}} / 2 \quad (7)$$

$$\text{Dělicí poměr} = \frac{2^{16}}{245} / 2$$

$$\text{Dělicí poměr} = 134$$

Z rovnice (7) mi vyšel dělicí poměr 134. V rovnici bylo nutné dělicí poměr vydělit dvěma, protože gyroskop snímá rychlost dopředu (max. + 245 °/s), ale také dozadu (max. -245 °/s). Každé výstupní číslo z gyroskopu je tedy nutné dělit číslem 134. Tuto konstantu budu dále využívat ve zdrojovém kódu pro program řídicí jednotky v pohyblivém modulu.

Jak je patrné z Obr. 22, z gyroskopu je vyvedeno 8 pinů (Vdd, GND, SCL, SDA, SDO, CS, DR, INT).

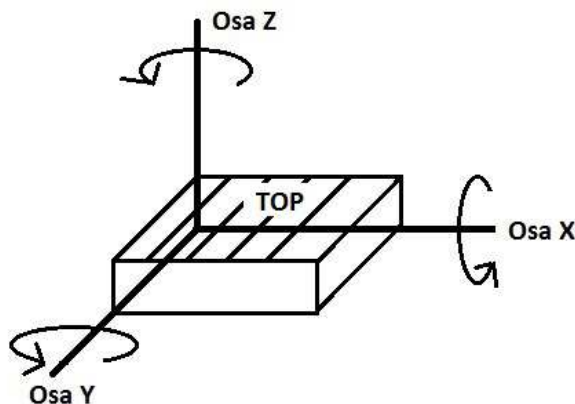


Obr. 22 Schéma zapojení gyroskopu A3G4250D [13]

Piny Vdd a GND jsou určeny pro připojení napájecího napětí (3.3 V). Protože tento gyroskop je digitální, pracuje s I2C sběrnici. Pro propojení gyroskopu s řídicí jednotkou jsou tedy určeny dva výstupní piny SCL a SDA sloužící pro přenos po I2C sběrnici. Pin SCL slouží pro přenos hodinového signálu a pin SDA pro přenos datového signálu. Pin CS aktivuje I2C sběrnici. Protože pro svou úlohu potřebují I2C sběrnici aktivovat, musí být CS připojen na Log. 1 (tedy na napájení 3.3 V). Pin SDO slouží pro identifikace Slave zařízení. To je užitečné pouze tehdy, pokud připojím po I2C sběrnici k mikrokontroléru dvě stejná Slave zařízení, tedy dva stejné gyroskopy. Já ovšem potřebuji pouze jeden gyroskop, tedy je jedno, zda si zvolím signifikační bit jako Log. 0, nebo jako Log. 1. Vybral jsem si první možnost. Zbylé dva piny (INT a DR) jsem ve své práci nevyužil.

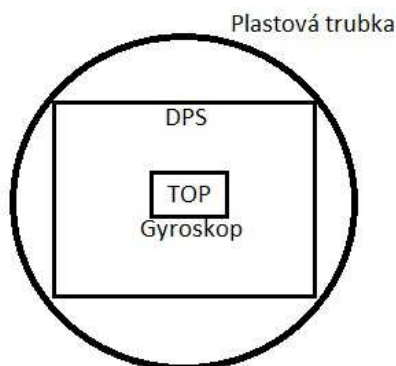
Kondenzátory C1 až C4 slouží pro potlačení rušivé složky.

V kapitole 3.3.1 jsem zmínil, že gyroskopem lze měřit rotaci ve třech osách. V ose X, Y a Z. Na Obr. 23 je znázorněné měření ve všech třech osách.



Obr. 23 Ukázka měření gyroskopem ve třech osách [19]

Protože mým cílem je zkonstruovat pohyblivý modul umístěný v plastové trubce, je zřejmé, že mi pro takový pohyb bude stačit využít pouze jedné osy, protože se modul bude pohybovat právě v jedné ose ve směru dopředu či dozadu. Gyroskop, který je na DPS, je v plastové trubce umístěn dle Obr. 24. Z obrázku je vidět, že pro pohyb trubky využívám osu „Z“, protože vrchní vrstva (TOP) gyroskopu je kolmá k ose otáčení plastové trubky.



Obr. 24 Ukázka konstrukce uložení gyroskopu do plastové trubky

4.3.3 Signalizační a ovládací prvky

Rád bych upozornil, že všechny signalizační prvky (3x LED dioda) a ovládací prvek (1x mikrospínač) nemají na přímou funkci modulu žádný vliv.

Pro postupný vývoj, testování a ověřování správnosti jednotlivých částí programu jsem používal právě tyto prvky.

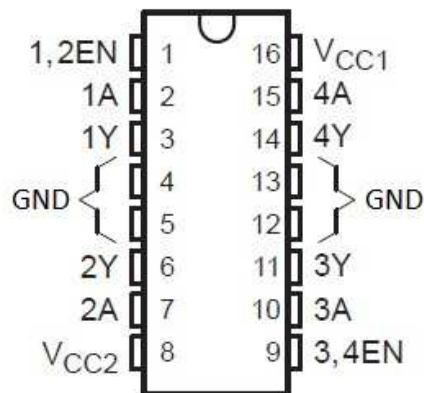
Abych tedy těchto prvků využil pro finální zařízení alespoň z vizuálního hlediska, fungují tři LED diody jako signalizace, že došlo k odeslání požadavku o pohybu modulu zadaném uživatelem v ovládacím programu.

Pokud je tedy v ovládacím programu zvolen pohyb dopředu, problikne pro kontrolu správného přenosu LED označená jako D1. Pokud uživatel zvolí mód pro zastavení modulu, problikne druhá LED (D2). Výběrem pohybu zpět dojde k probliknutí třetí LED značené jako D3.

Mikrospínač jsem nakonec pro finální funkčnost modulu vůbec nevyužil. Ovšem je dobré toto tlačítko použít např. tehdy, pokud bych potřeboval přepínat osu (X, Y, Z), ve které má probíhat měření gyroskopu. Při vývoji jsem právě této možnosti využil, ale pro finální část je tato funkce zbytečná a tudíž je tlačítko nevyužito.

4.3.4 H-můstek

Pro správnou funkci stejnosměrného motoru je H-můstek velmi důležitou součástí, jak jsem již uvedl v teoretické části mé práce (viz kapitola 3.1.2). Pro samotné funkční zařízení jsem použil již hotový H-můstek v integrovaném obvodu, který je z hlediska rozměrů nejvýhodnější a také jeho zapojení do obvodu není nikterak složité. Konkrétně jsem použil integrovaný H-můstek L293D, který disponuje dvojicí H-můstků. Pro můj modul budu ovšem využívat pouze jeden můstek, protože v plastové trubce bude umístěn pouze jeden motor. Rozmístění vývodu můstku L293D znázorňuje Obr. 25.



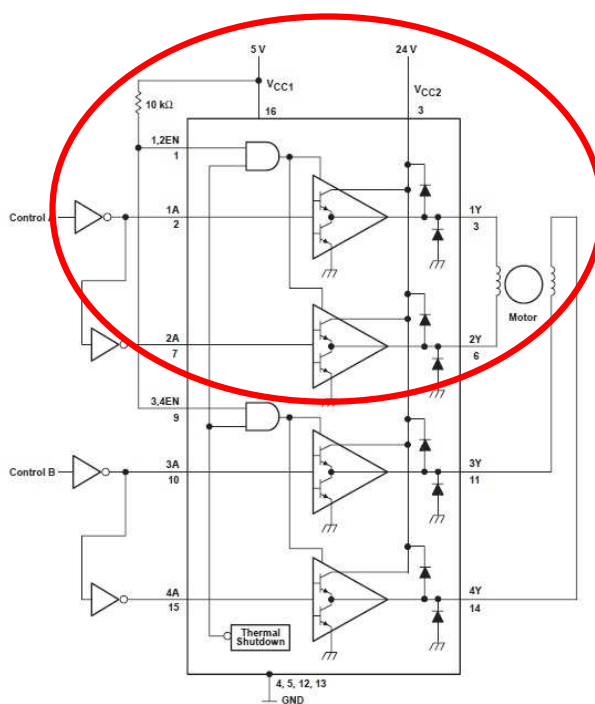
Obr. 25 Rozložení vývodů Integrovaného H-můstku L293D [17]

Vývod VCC1 slouží pro napájení logiky H-můstku a pro správnou funkčnost se musí toto napětí pohybovat od 4,5 V do 7 V. Druhý napájecí vývod VCC2 slouží pro napájení stejnosměrného motoru, který je následně k H-můstku připojen. Toto napětí se musí pohybovat v mezích VCC1 až 36 V. Další výstupní pin 1,2EN slouží pro řízení otáček motoru.

V mém případě tedy pro signál PWM z mikrokontroléru. Piny 1Y a 2Y slouží pro připojení stejnosměrného motoru a poslední dva vývody prvního H-můstku 1 A a 2 A slouží pro řízení směru otáčení motoru. Jsou tedy napřímo připojené k mikrokontroléru.

Maximální proudové zatížení pro jeden můstek je 0,5 A, což pro mé zařízení bohatě stačí, protože stejnosměrný motor nemá při plné zátěži větší odběr než 0,2 A.

Výhodou tohoto obvodu jsou již zaimplementované ochranné diody, které svádí napěťové špičky, jež vznikají na indukčnosti (viz blokové schéma na Obr. 26). Z blokového schématu je dále vidět, že řízení otáček má na starosti signál 1,2EN (v mém případě PWM), který je zařazen před řídicí signály 1A a 2A. Signál 1,2EN určuje, zda je motor vůbec ovládán, což je zajištěno logickým hradlem AND. Pokud je totiž tento signál nulový, motor se nikdy otáčet nebude a to ani s ohledem na stav řídicích signálů 1A a 2A.



Obr. 26 Blokové schéma integrovaného H-můstku L293D [17]

4.3.5 DC – motor

V této části práce se budu věnovat výběru stejnosměrného elektromotoru, který má v pohyblivém modulu zásadní význam, neboť slouží jako pohon celého zařízení. Výběr správného motoru je v mém zařízení zcela zásadní, protože zdaleka ne každý stejnosměrný motor lze pro mé zařízení použít.

Základní parametry při jmenovitém zatížení prvního testovaného stejnosměrného motoru bez převodovky popisuje Tab. 3:

Tab. 3 Základní parametry prvně testovaného stejnosměrného motoru

Provozní napětí	6 V
Provozní proud	1,34 A
Počet otáček	1253 ot/min.
Točivý moment	46,2 g·cm

Tento stejnosměrný motor se ukázal být nevhodný hned z několika důvodů. Jedním z nich byl jeho malý točivý moment. Příliš velký odebíraný proud také nebyl přijatelný, protože H-můstek, který využívám, lze použít pouze pro 500 mA proudové zatížení. Samozřejmě bych mohl použít jiný H-můstek, ale příliš velký odebíraný proud by vedl i k rychlému vybití baterie. Tento stejnosměrný motor jsem tedy rychle zavrhnul a došel jsem k závěru, že budu muset hledat stejnosměrné motory s převodovkou. Takové motory mají sice málo otáček/min., ale pro pohyb mého zařízení postačí pomalý pohyb, protože v mé práci není nutné dosáhnout co nejvyšší rychlosti modulu, ale aby měl modul co největší točivý moment a tím bylo možné modul roztáčet i do nakloněné roviny.

Po pečlivém zvážení jsem se nakonec rozhodl použít stejnosměrný elektrický motor s převodovkou 050SHV-250-R (viz Obr. 27), který má při jmenovitém zatížení a napájecím napětí 7 V tyto základní parametry (viz Tab. 4):

Tab. 4 Základní parametry stejnosměrného motoru s převodovkou 050SHV-250-R

Provozní napětí	7 V
Provozní proud	184 mA
Počet otáček	33 ot/min.
Točivý moment	2537 g·cm



Obr. 27 Stejnosměrný elektrický motor s převodovkou 050SHV-250-R

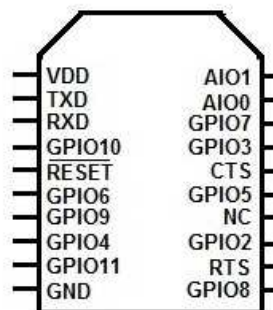
4.3.6 Bluetooth modul

Při výběru bluetooth modulu jsem se rozhodl pro modul RN42-I/RM. Základní parametry tohoto bluetooth modulu popisuje Tab. 5.

Tab. 5 Základní parametry bluetooth modulu RN42-I/RM [15]

Provozní napětí	3 – 3,6 V
Provozní proud – přenos dat	30 mA
Provozní proud – párování	3 mA
Klidový proud	26 μ A

Rozmístění vývodů tohoto modulu znázorňuje Obr. 28. Pro připojení napájecího napětí slouží vývody VDD a GND. Vývody TXD a RXD slouží pro propojení bluetooth modulu s mikrokontrolérem. Vývod TXD slouží pro sériové odesílání dat a RXD slouží pro sériový příjem dat. Podle katalogového listu modulu jsem mezi vývody GPIO2, GPIO5, GPIO6, GPIO9, GPIO10 a uzemnění zapojil rezistory s odporovou hodnotou 100 k Ω . Dále jsem vložil rezistory s odporovou hodnotou 10 k Ω mezi vývody AIO0, AIO1 a uzemnění.



Obr. 28 Rozložení vývodů bluetooth modulu RN42-I/RM [15]

Aby bylo možné využívat sériovou komunikaci mezi bluetooth modulem a počítačem, je nutné, nakonfigurovat bluetooth modul tak, aby sériová komunikace probíhala bezchybně. Pro konfiguraci bluetooth slouží tzv. terminál.

Terminál lze využít ve dvou módech. První mód je řídicí a slouží pro odesílání konfiguračních příkazů do bluetooth modulu a zároveň pro zpětný příjem potvrzovacích zpráv. V tomto módu je možné měnit např. přenosovou rychlost nebo mód bluetooth modulu (Master, nebo Slave). Druhý mód je čistě datový, tedy v tomto módu lze použít terminál jako zobrazovací okno pro příjem dat z bluetooth modulu. Pokud bych tedy například odeslal z bluetooth modulu řetězec „Ahoj“, zobrazí se mi v komunikačním módu v terminálu právě tento řetězec „Ahoj“.

Standartně je každý bluetooth modul nastaven jako Slave. V mém zařízení potřebuji bluetooth modul právě v módu Slave, protože Masterem je počítač. Proto jsem si pouze ověřil, že mnou použitý bluetooth modul je skutečně nastaven jako Slave.

Je ovšem nutné přenastavit přenosovou rychlost, která je standartně nastavena na 9600 Baud. Protože hodinový kmitočet mikrokontroléru, který bluetooth modul řídí je 1 MHz, je podle Tab. 6 nutné, nastavit přenosovou rychlost maximálně na 4800 Baud, kde je chybovost ještě v zeleném poli, tedy 0.2 %. Pro přenosovou rychlost 9600 Baud je chybovost již v červeném poli, tedy 7.5 %. Proto jsem externí bluetooth modul v terminálu přenastavil na přenosovou rychlost 4800 Baud.

Tab. 6 Hodnota chybovosti při hodinovém kmitočtu 1MHz [21]

Baud Rate	UBRR(dec)	UBRR(hex)	Chybovost %
300	207	0x00CF	0.2
600	103	0x0067	0.2
1200	51	0x0033	0.2
2400	25	0x0019	0.2
4800	12	0x000C	0.2
9600	6	0x0006	7.5

Seznam základních příkazů, kterými si lze volit mód terminálu a dále měnit nastavení bluetooth modulu je následující: [16]

- „\$\$\$“ – slouží pro vstup do řídicího módu. Je-li vše v pořádku, vrátí se do terminálu příkaz „CMD“ a spustí se řídicí mód terminálu (viz Obr. 29).
- „---<cr>“ – slouží pro výstup z řídicího módu. Je-li vše v pořádku, vrátí se do terminálu příkaz „END“.
- „SU,48<cr>“ – slouží pro nastavení přenosové rychlosti na 4800 Baud. Je-li vše v pořádku, vrátí se do terminálu příkaz „AOK“. Tento příkaz je nutné zadávat pouze v řídicím módu terminálu.
- „SM,0<cr>“ – slouží pro nastavení bluetooth modulu do Master módu. Je-li vše v pořádku, vrátí se do terminálu příkaz „AOK“. Tento příkaz je nutné zadávat pouze v řídicím módu terminálu.
- „SM,1<cr>“ – slouží pro navrácení bluetooth modulu do Slave módu. Je-li vše v pořádku, vrátí se do terminálu příkaz „AOK“. Tento příkaz je nutné zadávat pouze v řídicím módu terminálu.

Ukázku výpisu řídicího módu v terminálu je znázorněn na Obr. 29.



```
COM16
Send
CMD
***Settings***
BTA= 0006666C6153
BTName=RN42 - 6153
Baudrt (SW4) = 9600
Parity = None
Mode = Slave
Authen = 0
Encryp = 0
PinCod = 1234
Bonded = 0
Rem = NONE SET
Autoscroll Newline 9600 baud
```

Obr. 29 Ukázka výpisu řídicího módu terminálu

4.3.7 Napájecí napětí

Při návrhu elektroniky pro pohyblivý modul jsem dlouho zvažoval, jakým způsobem vyřeším napájení celého modulu. Protože je modul bezdrátový, je nutné využít primární baterii. Musí se brát v úvahu, že celá elektronika modulu má při plném zatížení proudový odběr cca 150 mA. Musel jsem tak pečlivě zvážit, jakou baterii pro pohyblivý modul použiji.

Nakonec jsem se rozhodl pro primární 9 V baterii GP Rechargeable 200 mAh, Ni-MH. Výhodou této baterie je možnost dobíjení, naopak nevýhodou je vyšší pořizovací cena (cca 180 Kč).

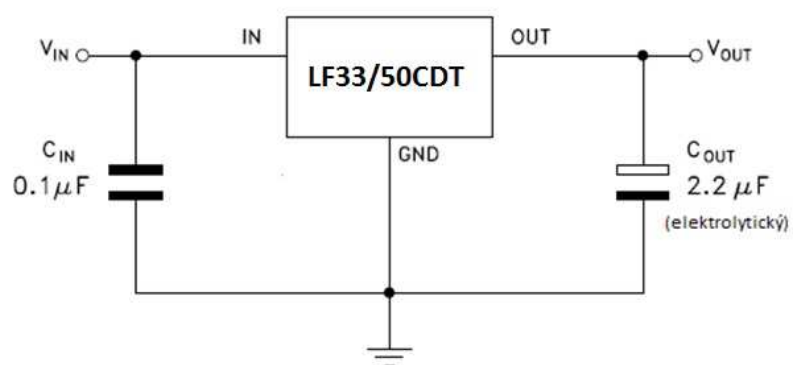
Celá elektronika pohyblivého modulu obsahuje tři napájecí úrovně, protože dle katalogových listů jednotlivých součástek nelze všechny součástky napájet stejným napětím.

První úroveň je 9 V, která je přímo využita z primární baterie jako vstup pro stabilizátory LF33CDT a LF50CDT a zároveň je využita pro napájení stejnosměrného elektromotoru přes H-můstek.

Další napětí, které je v elektronice pro pohyblivý modul použito, je 5 V, které je výstupem ze stabilizátoru LF50CDT. Napětí 5 V slouží pro napájení řídicí jednotky (mikrokontrolér ATmega32) a logiky v H-můstku.

Posledním napájecím napětím, které využívám pro napájení elektroniky pro pohyblivý modul, je napětí 3,3 V, které je výstupem stabilizátoru LF33CDT. Toto napětí slouží pro napájení bluetooth modulu a MEMS gyroskopu.

Abych dosáhl požadované hodnoty napětí 5 V a 3,3 V, použil jsem stabilizátor LF50CDT a LF33CDT. Schéma zapojení obou stabilizátorů je na Obr. 30.



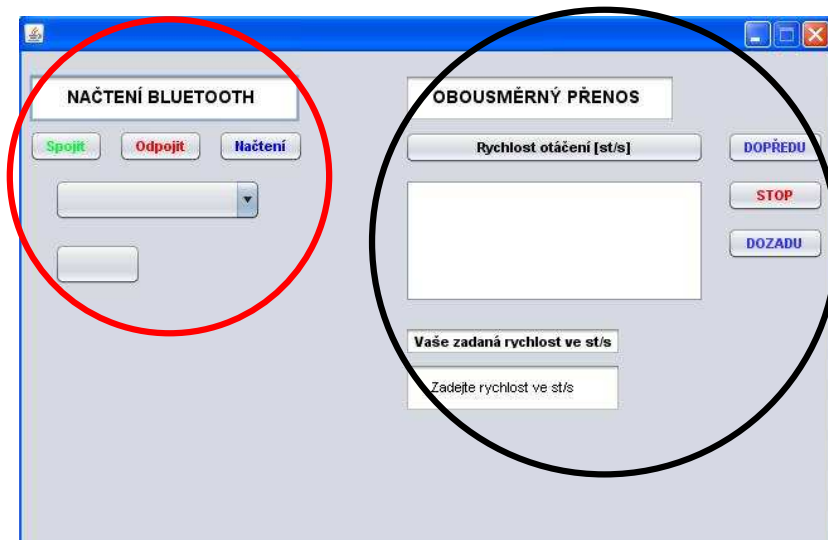
Obr. 30 Schéma zapojení stabilizátoru LF50CDT a LF33CDT [14]

4.4 Ovládací program

V této části mé práce se zaměřím na popis funkčnosti ovládacího programu mého zařízení, který slouží pro řízení pohyblivého modulu.

Zdrojové kódy ovládacího programu jsem napsal v programovém prostředí NetBeans. V tomto prostředí se využívá programovacího jazyka Java, který je pro objektové programování hojně využíván.

Na Obr. 31 je ukázán vizuální vzhled ovládacího programu. Při vývoji tohoto programu jsem se snažil vždy myslet na to, aby ovládání bylo snadné a intuitivní. Funkčnost a ovládání se dělí na dvě základní části. První část (na Obr. 31 v červeném kolečku pod názvem „NAČTENÍ BLUETOOTH“) slouží pro spárování bluetooth modulu v mém zařízení s počítačem. Druhá část ovládacího programu (na Obr. 31 v černém kolečku „OBOUSMĚRNÝ PŘENOS“) slouží pro ovládání celého pohyblivého modulu a zároveň je v této části možnost vyčítat aktuální rychlost otáčení zařízení v jednotkách %/s.



Obr. 31 Ukázka ovládacího programu pro řízení celého modulu

Při spuštění ovládacího programu se vždy musí začít první částí.

První část – „NAČTENÍ BLUETOOTH“

V této části jsou umístěna tři tlačítka: „Spojit“, „Odpojit“, „Načtení“. Při spuštění ovládacího programu je nutné, stisknout tlačítko „Načtení“, které zajistí vyhledání všech COM portů, které jsou v počítači využívány. Zde je nutné si ověřit, na jakém COM portu je právě mé zařízení.

Pro výběr správného COM portu slouží výběrové pole, které je umístěno pod tlačítkem „Načtení“. V tomto poli se tedy vybere příslušný COM port a stiskne se tlačítko „Spojit“. Za několik málo vteřin se provede spárování mezi pohyblivým modulem a počítačem. Pokud proběhlo spárování úspěšně, zezelená políčko, které je umístěno pod výběrovým polem pro COM porty. Nyní je ovládací program připraven zahájit obousměrný přenos, který probíhá ve druhé části.

Ukončení přenosu se provede kliknutím na tlačítko „Odpojit“, které přeruší spárování a tím ukončí komunikaci.

Druhá část – „OBOUSMĚRNÝ PŘENOS“

Tato část slouží výhradně pro ovládání modulu a zároveň pro čtení rychlosti otáčení. Funkce je postavena tak, že na začátku se zadá rychlost otáčení modulu (v jednotkách °/s) do výběrového pole, které je umístěno úplně dole. Rychlost otáčení musí být zadána v mezích od 0 do 255.

Nyní se čeká, jaký směr uživatel vybere pro pohyb modulu. Zde je na výběr ze tří tlačítek. První tlačítko „Dopředu“ slouží pro pohyb modulu jedním směrem, druhé tlačítko „Stop“ slouží pro zastavení modulu a třetí tlačítko „Zpět“ slouží pro opačný směr pohybu modulu oproti směru pohybu po stisknutí prvního tlačítka. Samozřejmě pokud uživatel zadá do výběrového pole pro rychlost otáčení číslo „0“, modul se nikdy neroztočí, bez ohledu na to, jestli dále stisknul tlačítko „Dopředu“, nebo „Dozadu“.

Poslední funkce, kterou ovládací program nabízí, je zobrazení aktuální hodnoty rychlosti otáčení pohyblivého modulu. Tato funkce funguje tak, že po stisknutí tlačítka „Rychlost otáčení [st/s]“ se zobrazí v okně pod tímto tlačítkem aktuální rychlost otáčení v jednotkách stupňů za sekundu.

4.5 Řešení - software

V této části mé práce se zaměřím na stručný popis řídicího programu pro mikrokontrolér ATmega32, který je umístěn v pohyblivé plastové trubce a zastává práci hlavní řídicí jednotky celého modulu. Popis programu budu provádět za pomoci vývojového diagramu.

Zdrojový kód řídicího programu jsem napsal v programovém prostředí CodeVision AVR studio v 2.60 od společnosti HP InfoTech. Výhodou tohoto vývojového prostředí je integrovaný modul C Compiler, který mi umožnil napsání zdrojového kódu v programovacím jazyce C, a následné přeložení tohoto kódu do programového jazyka Assembler.

Kompletní zdrojový kód je uveden v příloze (Příloha č. 6).

Řídicí program jsem rozdělil celkem do čtyř vývojových diagramů:

- obecný vývojový diagram
- vývojový diagram pro BLOK č. 1
- vývojový diagram pro BLOK č. 2
- vývojový diagram pro BLOK č. 3

4.5.1 Obecný vývojový diagram

V této části se budu věnovat obecnému popisu celého programu pro řídicí mikrokontrolér ATmega32. Vývojový diagram pro obecný popis programu ukazuje Obr. 32.



Obr. 32 Obecný vývojový diagram

Po startu programu se načtou všechny potřebné knihovny, které jsou nutné pro použití dalších funkcí v programu.

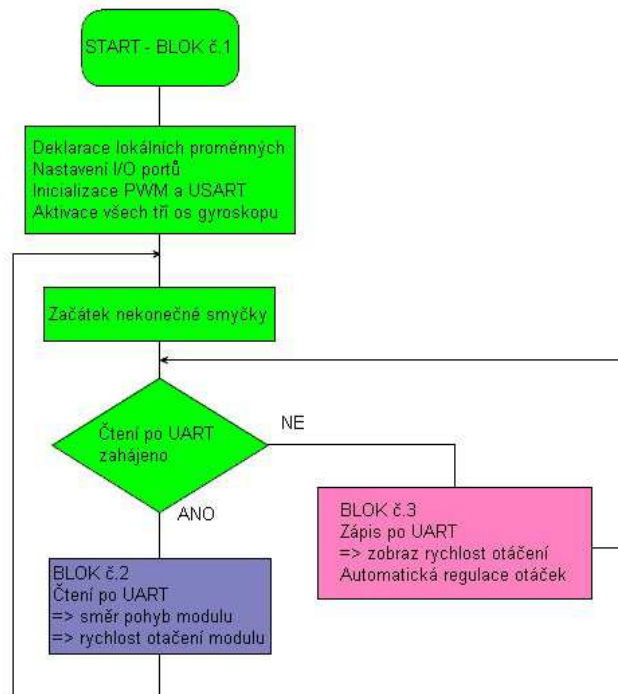
Následuje deklarace globálních proměnných, které jsou dosažitelné všemi funkcemi programu, tedy nejen funkcemi v hlavním těle programu.

Dále program generuje mnou nadefinovanou funkci pro I2C komunikaci, kterou později volám v hlavním těle programu. Tato funkce pracuje zvlášť pro horní a dolní byte. Začíná se horním bytem vysláním start bitu pro zahájení I2C komunikace, dále se inicializuje zápis a adresa SLAVE zařízení. Nyní začíná přenos dat, který je následně ukládán do proměnné „H“. Konec přenosu je dán stop bitem. Pro dolní byte funguje funkce stejně, jen se data ukládají do proměnné „L“.

Posledním blokem v obecném vývojovém diagramu je blok pro hlavní tělo programu, který budu popisovat podrobněji v následující části mé práce.

4.5.2 Vývojový diagram pro BLOK č. 1

V této části se budu věnovat podrobnějšímu popisu hlavního těla programu. Vývojový diagram pro hlavní tělo programu ukazuje Obr. 33. Pro lepší přehlednost a orientaci se tento vývojový diagram nazývá BLOK č. 1, protože se dále větví na další dvě části nazvané BLOK č. 2 a BLOK č. 3, které ale budu podrobněji popisovat až v následujících kapitolách.



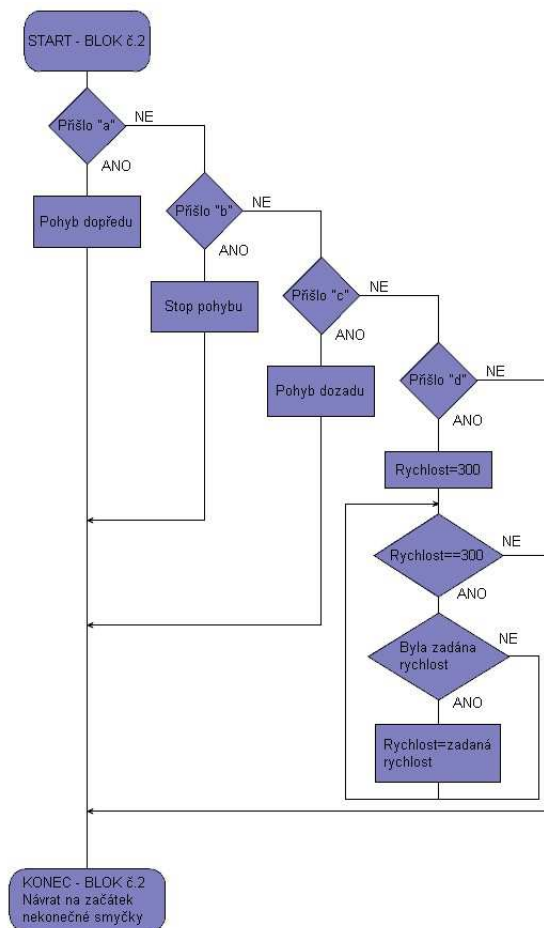
Obr. 33 Vývojový diagram pro BLOK č. 1

Začátek hlavního těla programu startuje deklarací lokálních proměnných, které využívá pouze toto hlavní tělo programu. Následuje nastavení stupně výstupních portů, které jsou nastaveny podle zapojení mikrokontroléru. Obecně jsou všechny piny mikrokontroléru nastaveny jako vstupní, ovšem piny PA0, PA1, PA2, PB1, PB2 a PD5 jsem přenastavil na výstupní piny. Dále program pokračuje inicializací vnitřní periférie čítač/časovač1 v režimu FAST PWM a inicializací komunikace USART, kterou taktéž nabízí mikrokontrolér ATmega jako svou vnitřní periférii. Na konci tohoto bloku ještě dochází k aktivaci všech tří os gyroskopu, které jsou obecně vypnuté, aby mohl zahájit snímání rotace.

Nyní běh programu spadá do nekonečné smyčky, která začíná podmínkou, zda bylo zahájeno čtení z UART, tedy jestli něco přišlo z počítače do mikrokontroléru. Pokud je podmínka splněna, přesouvá se program do BLOKu č. 2 (čtení po UART). V opačném případě se program dostává do BLOKu č. 3 (zápis po UART a automatická regulace).

4.5.3 Vývojový diagram pro BLOK č. 2

BLOK č. 2 (viz Obr. 34) slouží výhradně pro čtení dat po UART. V této části programu se zjišťuje, co přišlo po UART a podle toho se buď určuje směr otáčení motoru, nebo se zadává požadovaná rychlost otáčení modulu.



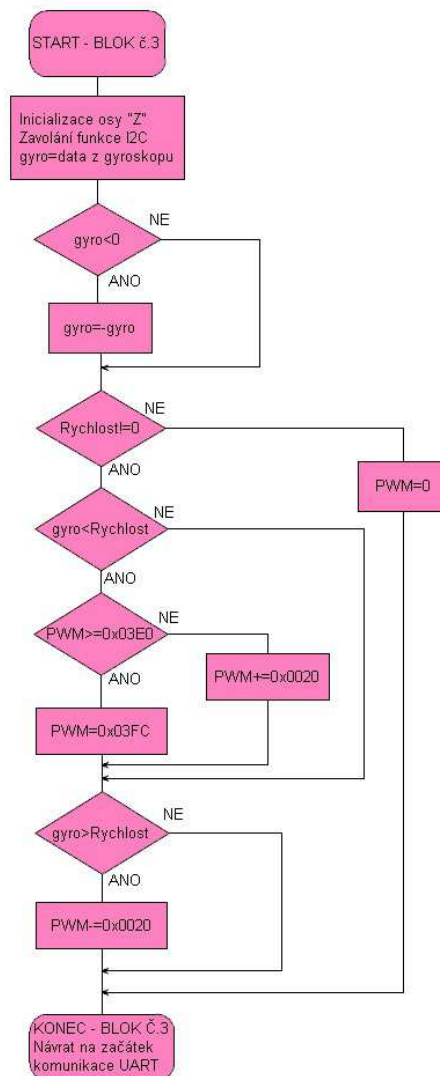
Obr. 34 Vývojový diagram pro BLOK č. 2

Tato část programu začíná postupnými podmínkami, zda přišlo písmeno a, b, c, nebo d. Pokud je některá z podmínek splněna, program se již dále neptá na zbylé podmínky a provede pouze úkony, které přísluší splněné podmínce a poté se vrátí na začátek nekonečné smyčky. Písmeno „a“ nastavuje pohyb modulu jedním směrem.

Písmeno „b“ zastavuje pohyblivý modul, písmeno „c“ nastavuje pohyb modulu opačným směrem než pohyb nastavený písmenem „a“ a konečně písmeno „d“ zajišťuje možnost odeslání požadované rychlosti otáčení modulu z počítače (resp. z ovládacího programu). V tomto módu program čeká do té doby, dokud není nějaká hodnota rychlosti otáčení (hodnota se musí pohybovat od 0 do 255) z počítače odeslána. Na začátku programu je obecně nastavena rychlost otáčení na nulu. Jakmile přijde z počítače hodnota (od 0 do 255), opustí program čekací smyčku a vrací se na začátek nekonečné smyčky, kde již do automatické regulace ukládá aktuálně přečtenou rychlost otáčení z počítače. Samozřejmě, pokud uživatel zadá rychlost otáčení nula, zaznamená program změnu rychlosti a opustí čekací smyčku, ale modul se nebude točit, protože byla zadána nulová rychlost otáčení.

4.5.4 Vývojový diagram pro BLOK č. 3

Na Obr. 35 je zobrazen vývojový diagram pro BLOK č. 3 popisující program, který zajišťuje zápis po UART, tedy odeslání rychlosti otáčení pohyblivého zařízení do počítače, kde může být následně tato hodnota zobrazena v ovládacím programu. Poté provádí automatickou regulaci rychlosti otáčení modulu.



Obr. 35 Vývojový diagram pro BLOK č. 3

Začátek této části programu startuje inicializací osy „Z“, ve které je snímána rotace modulu. Následně je volána I2C funkce, kterou jsem již popisoval dříve. V této funkci dochází k uložení aktuální hodnoty z gyroskopu do proměnné „gyro“. V proměnné „gyro“ je tak vždy uložena aktuální rychlost otáčení modulu, kterou snímá gyroskop.

Nyní se program dostává do podmínky, která se ptá, zda je hodnota v proměnné „gyro“ záporná. Pokud je podmínka splněna, hodnota „gyro“ se převede na kladné číslo. Jestliže podmínka splněna není, program dále pokračuje k další podmínce, kam se dostane i po převedení hodnoty „gyro“ na kladné číslo.

V další podmínce se zjišťuje, zda byla nějaká rychlost z počítače zadána, tedy pokud je proměnná „Rychlost“ jiné hodnoty než nula. Pokud není podmínka splněna, zůstává velikost PWM pro stejnosměrný motor nulová, tedy motor se nebude točit. Pokud je podmínka vyhodnocena pozitivně, dostává se program do části programu pro automatickou regulaci.

Automatická regulace pracuje tak, že se v každém cyklu zjišťuje, zda je hodnota proměnné „gyro“ větší, nebo menší než zadaná rychlost z počítače. Pokud je hodnota „gyro“ menší, dochází k načtení velikosti PWM o 0x0020h. Pokud tedy začíná regulace a modul se roztáčí od nulové rychlosti, tak se v každém cyklu načítá hodnota PWM o 0x0020h a tím se postupně navyšuje rychlost modulu až do požadované rychlosti. Pokud ovšem hodnota „gyro“ již převyšuje zadanou rychlost z počítače, dochází naopak k odečtení hodnoty PWM o stejnou velikost, jako při navyšování, tedy o 0x0020h. Tím se rychlost modulu stále reguluje a udržuje se tak požadovaná rychlost otáčení modulu. Po dokončení prvního kola regulace se program dostává na konec tohoto bloku a tím dochází k návratu programu hned pod začátek nekonečné smyčky, tedy do podmínky, zda bylo zahájeno čtení z UART či nikoliv.

Na závěr bych ještě rád zmínil vnořenou podmínku, kterou program provádí pouze tehdy, je-li hodnota „gyro“ menší, než zadaná rychlost z počítače. Je-li tato podmínka splněná, má v regulaci zásadní význam právě tehdy, když byla hodnota PWM zadána do téměř maximální hodnoty. Pokud tomu tak je, podmínka toto vyhodnotí a zajistí, aby už dále nedocházelo k načítání hodnoty PWM a naopak již nechá hodnotu PWM pevně nastavenou na hodnotu 0x03FC, tedy téměř na maximální hodnotu PWM (maximální hodnota PWM je 03FFh). Pokud by totiž tato podmínka nebyla v programu zaimplementována, došlo by při přetečení maximální hodnoty PWM k zastavení

modulu, protože PWM při přetečení maximální hodnoty, načítá opět od nuly (viz funkce PWM v kapitole 3.1.4 v části „*Rychlý režim PWM*“).

Jakmile v kapitole „Testování zařízení“ přesně určím, jakou maximální možnou hodnotu pro rychlost otáčení modulu lze zadat z počítače, aby se modul touto rychlostí skutečně točil, pak není vnořená podmínka, která hlídá přetečení hodnoty PWM, potřeba. Ovšem pro „ošetření“ pohybu modulu je lepší tuto podmínku zimplementovat do programu. Uživatelé jsou pořád pouze lidé, kteří se mohou splést a omylem zadat vyšší rychlost, než je maximální povolená a pak by se modul zastavoval a zrychloval a to by nebylo moc efektivní.

4.6 Konstrukční řešení

Konstrukční řešení celého modulu je zcela zásadní pro správný a plynulý pohyb plastové trubky. Na začátku své práce jsem věděl, že základem bude plastová trubka, stejnosměrný motor umístěný v ose trubky a nějaké závaží připevněné k ose motorku.

Při prvním přemýšlení nad konstrukcí modulu jsem ovšem dlouho zvažoval, jak umístit motor do středu trubky, jak těžké má být závaží, jak těžký může být modul, aby motor trubku vůbec roztočil, jak dlouhá má být trubka, kam a jak umístit elektroniku a mnohé další.

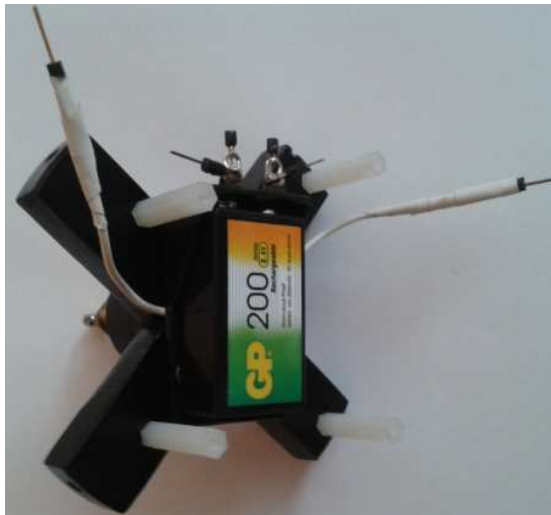
4.6.1 Kříž z pertinaxu

Po výrobě prvních prototypů modulu jsem došel k závěru, že motor umístím do osy trubky pomocí kotouče z pertinaxu, který se ukázal být jako dobrý materiál pro obrábění a zároveň je velmi lehký, což je pro mé zařízení velmi užitečné. Kotouč jsem nakonec ještě za pomoci soustruhu a přímočaré pily upravil tak, aby vytvořil pouze kříž, který je ještě lehčí, než celý kotouč. Uprostřed kříže jsem vysoustružil kulatý otvor pro upevnění motorku a na konci kříže jsem vytočil za pomoci závitníků závity pro šrouby, aby bylo možné upevnit kříž k trubce. Na závěr jsem celý kříž nastříkal černým sprejem. Finální vzhled a tvar kříže ukazuje Obr. 36.



Obr. 36 Kříž z pertinaxu pro upevnění motoru v ose trubky

Výhodou tohoto kříže je i to, že jsem k němu následně mohl velmi snadno přišroubovat celou DPS včetně 9 V baterie. Jak vypadá složený kříž s motorem, baterií a čtyřmi distančními sloupky pro upevnění DPS ukazuje Obr. 37.



Obr. 37 Sestavený kříž z pertinaxu

4.6.2 Plastová trubka

Po několika prvních zkouškách spojených s výběrem vhodné délky a průměru trubky jsem nakonec použil plastovou trubku o průměru 10 cm a délce 17 cm. V celé trubce jsou umístěny čtyři šrouby, kterými je k trubce přišroubován kříž z pertinaxu spolu s motorem, DPS a baterií. Dále jsou v trubce umístěny dva šrouby pro upevnění přídržného plátku pro hřídel motoru a pak je v trubce vyvrtána jedna větší díra, která slouží pro zašroubování drátku od závaží k ose motoru. Následně jsem trubku nastříkal stříbrným sprejem a na oba konce umístil dva gumové kroužky pro lepší pohyb trubky. Finální vzhled trubky je znázorněn na Obr. 38.



Obr. 38 Plastová trubka modulu

4.6.3 Olověné závaží pro udržení klidové polohy statoru

Při postupném vývoji nevhodnějšího tvaru závaží a jeho gramáže jsem sestavil opět několik různých prototypových tvarů, u kterých jsem následně testoval a zkoumal, které z nich má v trubce nejlepší vlastnosti z hlediska plynulosti a rychlosti pohybu modulu. Nakonec jsem se rozhodl odlít z olova zcela originální tvar závaží ve tvaru kotvy. Tento tvar se ukázal jako nejlepší řešení. Výroba probíhala tak, že jsem si nejprve ze dřeva vyřezal požadovaný tvar. Pomocí dřevěného modelu jsem si ze sádry odlil formu, do které jsem umístil drátek pro připevnění závaží k hřídeli motoru. Do této sádrové formy jsem v konečné fázi nalil olovo a tímto celý tvar závaží vyhotovil. Odlitek jsem zbrousil do zaobleného tvaru a následně nastříkal zeleným sprejem. Celé závaží nakonec váží 220 gramů. Obr. 39 ukazuje finální tvar olověného závaží.



Obr. 39 Finální tvar závaží pro pohyblivý modul

4.6.4 Celková konstrukce modulu

Celá konstrukce pohyblivého modulu má tu výhodu, že ji lze celou rozebrat, takže je případně možnost pohodlně modul opravovat, nebo měnit jednotlivé díly (viz Obr. 40).



Obr. 40 Všechny konstrukční a elektronické části modulu

1 – osazená DPS, 2 - přídržný plátek pro hřídel motoru, 3 - plastová trubka, 4 - držák baterie spolu s 9 V baterií, 5 - šroubky, matky a distanční sloupky, 6 - stejnosměrný motorek, 7 - olovené závaží, 8 - přípojné drátky mezi baterií a DPS, a mezi motorem a DPS, 9 - dva gumové kroužky, 10 - kříž z pertinaxu.

5. Testování zařízení

V rámci testování funkčního zařízení jsem si ověřil, že modul je schopen se pohybovat oběma směry a zároveň je možné řídit rychlost modulu, která je následně automaticky regulována.

Během testování jsem se nejvíce zaměřil na určení minimální a maximální rychlosti modulu, na maximální úhel stoupání, který je schopno zařízení vyjet a na ověření správné funkčnosti ovládacího programu.

Maximální a minimální rychlost modulu

Při roztočení plastové trubky jsem zjistil, že pohyb trubky je nejvíce plynulý tehdy, když se trubka točí maximální možnou rychlostí. Maximální možná rychlost trubky je cca 200 °/s. Naopak minimální rychlost, kterou je třeba zadat, aby se modul začal roztáčet, je 50 °/s. Při maximální rychlosti modulu (200 °/s) je automatická regulace nejvíce účinná, protože tím, že se modul pohybuje téměř plynule, je regulace nejvíce přesná. Při minimální rychlosti modulu (50 °/s) je naopak automatická regulace nejméně účinná, protože celý modul se točí „sekavě“ a tím se rychlost modulu skokově mění a do automatické regulace tak přichází značně zkreslená aktuální rychlost otáčení.

Maximální úhel stoupání, který je schopen modul překonat

Tato část testování přímo navazuje na kapitolu 3.2., kde jsem v části „Odvození pro maximální úhel stoupání“ odvodil výpočet pro maximální možný úhel stoupání, který modul vyjede. Při zjištění hodnot pro proměnné, které se ve výpočtu nachází, jsme tak mohl snadno tento výpočet provést a vytvořit si tak teoretickou představu, do jak velkého úhlu stoupání je možné modul vyslat, aby ho bez problému vyjel. Z rovnice (6) je vidět, že pro výpočet maximálního úhlu stoupání je potřeba znát hmotnost závaží a celého modulu a také poloměr modulu a ramena závaží. V mém případě mají tyto proměnné následující hodnoty:

- $R_1 = 5 \text{ cm}$
- $R_2 = 2,8 \text{ cm}$
- $m_1 = 620 \text{ g}$
- $m_2 = 220 \text{ g}$.

Když jsem tedy tyto proměnné dosadil do rovnice, vyšel mi maximální možný úhel stoupání 11,5 ° (viz Tab. 7). Při reálném měření maximálního úhlu stoupání jsem naměřil hodnotu cca 11° (viz Tab. 8). Ukazuje se tedy, že výpočet téměř odpovídá reálnému měření. Drobná odchylka může být způsobena chybou měření reálného úhlu α , nebo nepřesným určením proměnných, které jsem dosazoval do rovnice. V obou tabulkách je dále vidět výpočet a měření, kdy jsem hmotnost modulu postupně navyšoval o 50 g. Také zde je vidět pouze malý rozdíl mezi výpočtem a reálným měřením. Pro porovnání výpočtů a měření jsem hodnoty z tabulek vložil do názorného grafu (viz Obr. 41). Měření tedy ukázalo, že modul vyjede maximální úhel stoupání cca 11 °. Čím těžší je hmotnost modulu, tím menší bude maximální možný úhel stoupání, který bude modul schopen překonat.

Tab. 7 Výpočet maximálního úhlu stoupání pro různé hmotnosti modulu

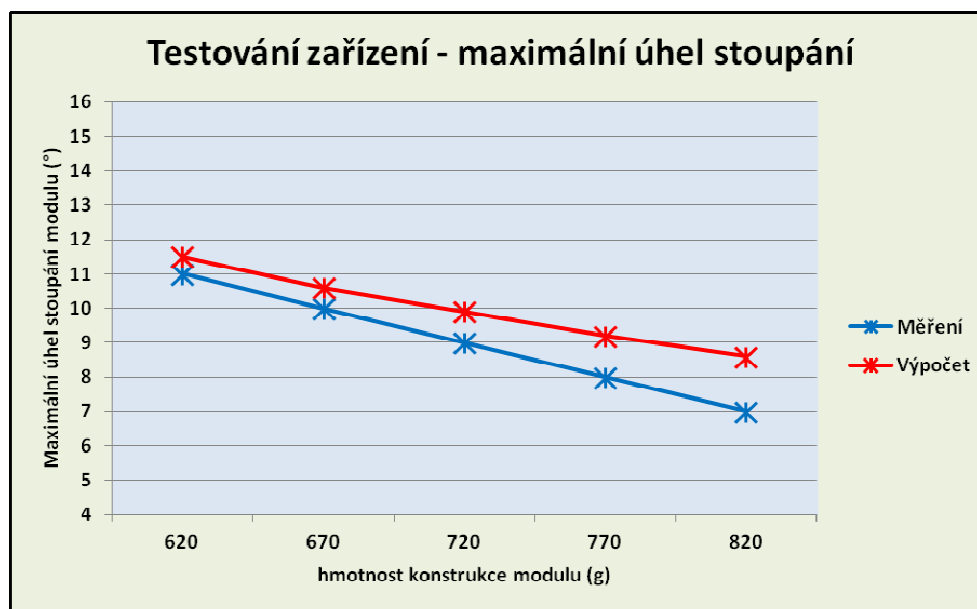
Výpočet

m1 (g)	620	670	720	770	820
max. úhel stoupání [°]	11,5	10,6	9,9	9,2	8,6

Tab. 8 Měření maximálního úhlu stoupání pro různé hmotnosti modulu

Měření

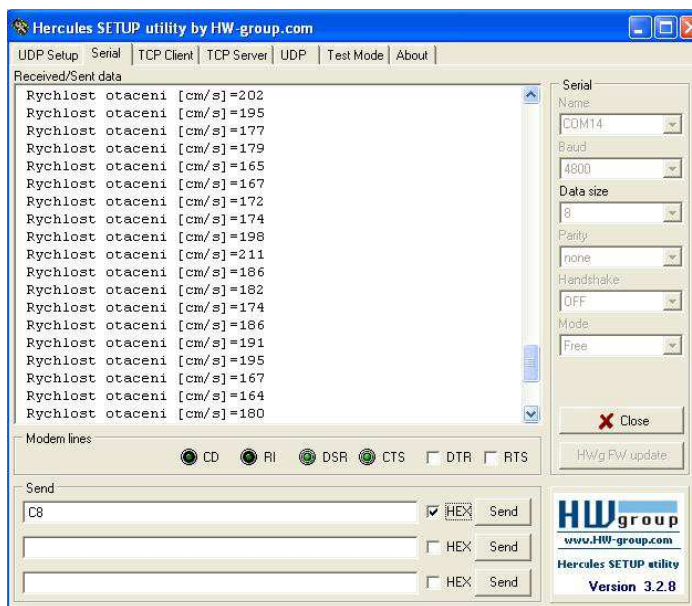
m1 (g)	620	670	720	770	820
max. úhel stoupání [°]	11	10	9	8	7



Obr. 41 Porovnání vypočítaných a naměřených hodnot pro maximální možný úhel stoupání

Ověření správné funkčnosti ovládacího programu

Při testování správné funkčnosti ovládacího programu jsem zjistil, že při párování modulu s počítačem je modul rušen a ne vždy se mi spárování podaří. Ve školní laboratoři se mi spárování pomocí ovládacího programu nepodařilo vůbec. Pro bezdrátové ovládání modulu je tedy lepší využít terminál. Já jsem použil terminál „Herkules 3.2.8.3“, který se s modulem spáruje téměř vždy. Ukázka správné funkčnosti bezdrátového přenosu pomocí terminálu je na Obr. 42.

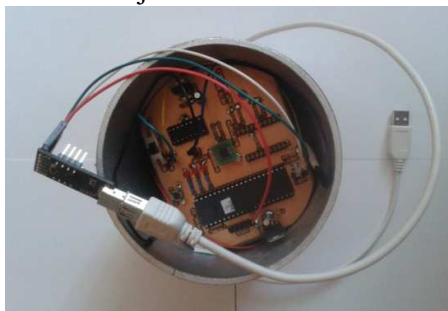


Obr. 42 Ukázka použití terminálu pro řízení modulu

Na terminálu je vidět, že jsem do okénka pro posílání znaku zadal hodnotu C8h, tedy 200 v dekadickém vyjádření. Modul by se tedy měl točit rychlostí 200 %/s. Z obrázku je patrné, že automatická regulace skutečně reguluje modul a drží jeho rychlost v rozmezí od 160 do 211 %/s. Zde je vidět i menší nedokonalost regulace, která je nepřesná z důvodu ne úplně plynulého pohybu trubky.

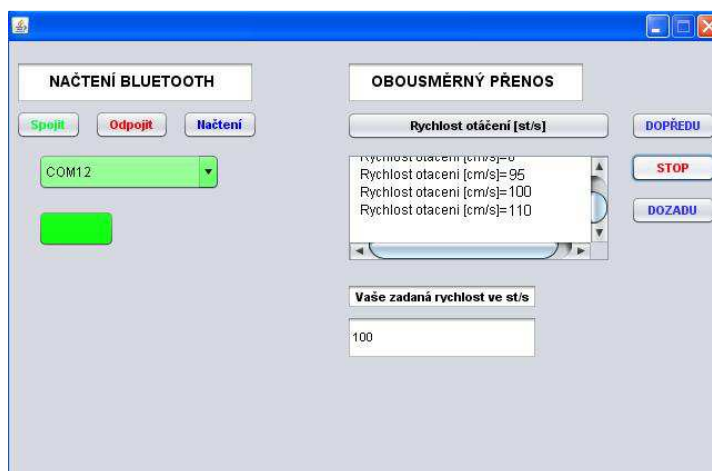
Abych ovšem skutečně ověřil správnou funkci ovládacího programu, připojil jsem celý modul k počítači za pomoci převodníku USB na UART TTL – CP2102, který slouží pro drátovou komunikaci přes sériovou linku UART s počítačem. Díky drátovému spojení modulu s trubkou se mi modul vždy spáruje s ovládacím programem a mohu tak spolehlivě ověřit funkčnost ovládacího programu. Převodník se k modulu připojí pouze pomocí dvou vodičů RX a TX. Pro správnou funkčnost je samozřejmě nutné propojit vývod GND mezi převodníkem a modulem.

Zapojený převodník k modulu ukazuje Obr. 43.



Obr. 43 Zapojený převodník USB/UART k modulu

Za pomoci drátového spojení modulu s počítačem mi ovládací program funguje velmi spolehlivě. Na Obr. 43 je vidět ověření funkčnosti. Po spárování modulu v levé části ovládacího programu jsem zadal do okénka pro rychlost otáčení hodnotu 100 a následně vybral směr otáčení modulu (klinutím na okénko „DOPŘEDU“ NEBO „DOZADU“). Tím jsem očekával, že se modul bude otáčet rychlostí kolem 100 °/s. Hodnoty v okénku pro zobrazení aktuální rychlosti modulu dokazují, že modul je skutečně regulován a rychlost otáčení se aktuálně pohybuje od 95 do 110 °/s. Jakmile jsem kliknul na tlačítko „STOP“, modul se ihned přestal točit.



Obr. 44 Ukázka funkčnosti ovládacího programu

6. Možná vylepšení

Při otestování zařízení mě napadlo několik možných vylepšení, které by vedly k lepší funkčnosti celého zařízení.

Pro lepší plynulost pohybu modulu a jeho rychlosti by se mohl použít lepší motor, který by měl kvalitnější převody pro větší zátěž. Další možností pro větší plynulost pohyblivého modulu by mohlo být použití účinnější regulace (např.: PID regulace).

Maximální možný úhel stoupání, který je modul schopen vyjet, by mohl být ještě větší. Pokud bych například počítal hmotnost baterie, motorku, elektronických součástek a konstrukce z extrudovaného polystyrenu, tak hmotnost modulu bez závaží se bude pohybovat okolo 40 g. Hmotnost samotného závaží bych nechal 220 g. Délku poloměru trubky (5 cm) a délku ramene závaží (2,8 cm) bych nechal také stejnou. Takto bych maximální možný úhel, který by byl modul schopný vyjet, zvětšil na hodnotu cca 28 °. Pokud bych navíc zauvažoval zvětšit poloměr trubky na 15 cm, délka ramene pro závaží by se tak pohybovala okolo 12 cm, maximální možný úhel stoupání by se ještě o něco zvětšil, protože podíl mezi oběma hmotnostmi by najednou nebyl tak malý. V tomto případě by byl 0,8. Maximální možný úhel stoupání, který modul překoná, by nyní byl cca 43 °.

Z hlediska funkčnosti ovládacího programu by bylo dobré doladit ovládací program tak, aby bezpečně fungoval v každém počítači a nebylo nutné pro bezdrátové spojení používat pomocný terminál. Bohužel jsem nedošel k žádnému korektnímu výsledku, který by tento problém vyjasnil. Je možné, že pokud bych ovládací program nepsal v jazyce Java, ale v nějakém pokročilejším programovacím jazyce, problém při párování by se odstranil.

7. Závěr

V první části mé práce jsem se seznámil s problematikou řízení a regulace stejnosměrných motorů, kde jsem se nejprve věnoval obecnému principu činnosti a konstrukci motoru a dále jsem se již zaměřil na řízení a možnou regulaci otáček stejnosměrného motoru. V další kapitole jsem se zabýval principem mechanismu přenosu momentu síly. Zde jsem také následně odvodil vzorec pro výpočet maximálního úhlu stoupání, který jsem později využil při testování funkčního zařízení. Následně jsem se zabýval problematikou snímání otáček za pomoci gyroskopu. Konkrétně jsem se zde zaměřil na funkčnost gyroskopu MEMS. První část jsem uzavřel rozбором problematiky o sériovém komunikačním rozhraní, u kterého jsem se blíže zaměřil na bezdrátové spojení přes sériové rozhraní UART.

Ve druhé části mé práce jsem se již věnoval praktickému využití získaných poznatků v první části a na základě těchto jsem navrhnul a dále zkonstruoval pohyblivý modul na principu mechanismu přenosu momentu síly. V této části práce jsem blíže popsal postupné kroky návrhu elektroniky pro pohyblivý modul a ovládacího programu pro funkční zařízení, kde jsem i na závěr uvedl konkrétní postup při návrhu konstrukčního provedení modulu.

V rámci testování funkčnosti zařízení jsem si ověřil, že modul je schopen se pohybovat oběma směry a zároveň je možné řídit jeho rychlost otáčení, na kterou je modul následně automaticky regulován. Maximální rychlost modulu je 200 °/s a maximální možný úhel stoupání, který je schopen modul překonat je cca 11 °. Modul se řídí bezdrátovým přenosem za pomoci externího bluetooth modulu ovládacím programem v počítači. Během testování správné funkčnosti ovládacího programu jsem zjistil, že při párování modulu s počítačem je modul rušen a pomocí ovládacího programu se mi ne vždy spárování podaří. Pro bezdrátové spojení modulu s počítačem je tak lepší využít terminál, který se s modulem spáruje téměř vždy.

8. Seznam literatury

- [1] Feynman, R. P., Leighton, R. B., Sands M., Feynmanovy přednášky z fyziky – revidované vydání, 1. díl, Fragment 2013, ISBN 978-80-253-1642-9.
- [2] Matoušek, D., *Práce s mikrokontroléry ATMEML AVR AT90S*, 3. díl, Praha 2006, Nakladatelství BEN – technická literatura, ISBN 80-7300-209-4.
- [3] Matoušek, D., *Práce s mikrokontroléry ATMEML AVR AT mega 16*, 4. díl, Praha 2006, Nakladatelství BEN – technická literatura, ISBN 80-7300-174-8.
- [4] Mann, B., *C pro mikrokontroléry*, 1. Vydání, Praha 2003, Nakladatelství BEN - technická literatura, ISBN 80-7300-077-6.
- [5] Vedral, J., Fischer, J., *Elektronické obvody pro měřicí techniku*, Praha 2004, Vydavatelství ČVUT, ISBN 80-01-02966-2.
- [6] Váňa, V., *Mikrokontroléry ATMEML AVR - programování v jazyce C*. 1. vydání. Praha 2003, Nakladatelství BEN – technická literatura, ISBN 80-7300-102-0.
- [7] Vacek, V., *Učebnice programování ATMEML s jádrem 8051*. 1. vydání. Praha 2001, Nakladatelství BEN – technická literatura, ISBN 80-7300-043-1.
- [8] Hrázský, J., *Mikropočítače a počítače 2: Učebnice výpočetní techniky pro SPŠ*. Praha 2003, Nakladatelství Střední průmyslová škola dopravní.
- [9] Kolomazník, V., *Návrh a realizace řídicích jednotek pro semiautonomní konvoj* [online]. Brno: VUT 2014. [přístup 14. 3. 2015]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=86957. Bakalářská práce. VUT Fakulta strojíního inženýrství. Ústav mechaniky těles, mechatroniky a biomechaniky.
- [10] Krůžela, M., *Regulace otáček elektromotoru* [online]. Zlín: UTB 2006. [přístup 20. 1. 2015]. Dostupné z: http://digilib.k.utb.cz/bitstream/handle/10563/905/kr%C5%AF%C5%BEela_2006_bp.pdf?sequence=1. Bakalářská práce. UTB Fakulta aplikované informatiky.
- [11] Mihalko, J., *MEMS inerciální snímače* [online]. Brno: VUT 2012. [přístup 4.2.2015]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=52098. Diplomová práce. VUT Fakulta elektrotechniky a komunikačních technologií. Ústav automatizace a měřicí techniky.
- [12] Atmel Corporation, Inc. *Datasheet, ATmega32 8bit AVR mikroprocesor* [online]. 2011, [přístup 29. 12. 2014]. Dostupné z: <http://www.atmel.com/images/doc2503.pdf>

- [13] STMicroelectronics, Inc. *Datasheet, A3G4250D MEMS gyroskop* [online]. 2012, [přístup 29. 12. 2014]. Dostupné z: <http://www.st.com/web/en/resource/technical/document/datasheet/DM00047823.pdf>
- [14] STMicroelectronics, Inc. *Datasheet, LF00 stabilizátor napětí* [online]. 2003, [přístup 29. 12. 2014]. Dostupné z: <http://www.farnell.com/datasheets/93506.pdf>
- [15] Roving Networks, Inc. *Datasheet, RN41XV & RN42XV Bluetooth Module* [online]. 2012, [přístup 20.1.2015]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/RN41XV-RN42XV-ds-v1.0r.pdf>
- [16] Roving Networks, Inc. *Uživatelská příručka, Bluetooth Data Module Command Reference & Advanced Information* [online]. 2013, [přístup 25.1.2015]. Dostupné z: http://ww1.microchip.com/downloads/en/DeviceDoc/bluetooth_cr_UG-v1.0r.pdf
- [17] Texas Instruments, Inc. *Datasheet, L293D Quadruple half-H-driver* [online]. 1998, [přístup 25.1.2015]. Dostupné z: <http://www.gme.cz/img/cache/doc/399/017/l293d-datasheet-1.pdf>
- [18] HUSÁK, M., *MEMS a mikrosystémové technologie*. Automa [online]. 2008, roč. 14, č. 11 [přístup 29.12. 2015], str. 7-11. Dostupný z: <http://www.odbornecasopisy.cz/res/pdf/38122.pdf>
- [19] Vojáček, A., *Integrované MEMS GYROSKOPY* [online]. 2009, [přístup 29. 12. 2014]. Dostupné z: <http://automatizace.hw.cz/integrované-mems-gyroskopy>
- [20] Novák, I., *Stejnoseměrný elektromotor* [online]. 2008, [přístup 3.3.2015]. Dostupné z: <http://www.emotor.cz/stejnosemny-elektromotor.htm>
- [21] AVR BaudCalc, *AVR Baud Rate Calculator* [online]. 2005, [přístup 8.4.2015]. Dostupné z: <http://wormfood.net/avrbaudcalc.php?postbitrate=9600&postclock=1>
- [22] RoboWeb, *H-můstek s tranzistory* [online]. 2011, [přístup 3.3.2015]. Dostupné z: <http://robodoupe.cz/2011/robotem-sem-robotem-tam-ii-%E2%80%93-elektronika-take-neni-k-zahozeni/>

9. Seznam obrázků

Obr. 1 Schéma stejnosměrného elektromotoru [20].....	4
Obr. 2 Schéma zapojení H-můstku [22].....	5
Obr. 3 Příklady průběhů PWM modulace [10]	6
Obr. 4 Časový diagram režimu rychlého PWM [12]	7
Obr. 5 Časový diagram fázově korigovaného režimu PWM [12].....	8
Obr. 6 Moment síly [1]	9
Obr. 7 Příklad maximálního úhlu stoupání	10
Obr. 8 Náhled na osy gyroskopu [19]	12
Obr. 9 Příklad působení Coriolisovy síly (modrá šipka) na osobu pohybující se od středu k okraji rotujícího kruhu [19].....	12
Obr. 10 Působení Coriolisovy síly na mechanický gyroskop tvořený hmotným objektem periodicky se pohybující (kmitající) k okraji kruhu (vlevo) a ke středu (vpravo) [19].....	13
Obr. 11 Zjednodušená struktura snímače MEMS gyroskopu a příklad funkce struktury snímače gyroskopu při rotaci [19].....	13
Obr. 12 Blokové schéma MEMS gyroskopu [13].....	14
Obr. 13 Tuning fork gyroskop; a) fyzikální princip; b) struktura [11].....	15
Obr. 14 Princip I2C komunikace [2].....	15
Obr. 15 I2C komunikace - START a STOP [2].....	16
Obr. 16 Průběh vysílání po I2C sběrnici [2]	16
Obr. 17 Asynchronní osmibitový přenos [9].....	17
Obr. 18 Blokové schéma modulu USART pro mikrokontroler ATmega32 [12].....	19
Obr. 19 Blokové schéma - obecný návrh	21
Obr. 20 Celkové schéma elektroniky pro pohyblivý modul	22
Obr. 21 Rozložení vývodů mikrokontroléru ATmega32 na pouzdrě PDIP [12]	23
Obr. 22 Schéma zapojení gyroskopu A3G4250D [13]	26
Obr. 23 Ukázka měření gyroskopem ve třech osách [19]	27
Obr. 24 Ukázka konstrukce uložení gyroskopu do plastové trubky.....	27
Obr. 25 Rozložení vývodů Integrovaného H-můstku L293D [17]	28
Obr. 26 Blokové schéma integrovaného H-můstku L293D [17]	29
Obr. 27 Stejnosměrný elektrický motor s převodovkou 050SHV-250-R	30
Obr. 28 Rozložení vývodů bluetooth modulu RN42-I/RM [15].....	31
Obr. 29 Ukázka výpisu řídicího módu terminálu.....	33
Obr. 30 Schéma zapojení stabilizátoru LF50CDT a LF33CDT [14].....	34
Obr. 31 Ukázka ovládacího programu pro řízení celého modulu	35
Obr. 32 Obecný vývojový diagram.....	38
Obr. 33 Vývojový diagram pro BLOK č. 1	39
Obr. 34 Vývojový diagram pro BLOK č. 2	40
Obr. 35 Vývojový diagram pro BLOK č. 3	42
Obr. 36 Kříž z pertinaxu pro upevnění motoru v ose trubky	45
Obr. 37 Sestavený kříž z pertinaxu	45
Obr. 38 Plastová trubka modulu.....	46
Obr. 39 Finální tvar závaží pro pohyblivý modul	46
Obr. 40 Všechny konstrukční a elektronické části modulu.....	47

Obr. 41 Porovnání vypočítaných a naměřených hodnot pro maximální možný úhel stoupání...	49
Obr. 42 Ukázka použití terminálu pro řízení modulu	50
Obr. 43 Zapojený převodník USB/UART k modulu	51
Obr. 44 Ukázka funkčnosti ovládacího programu	51
Obr. 45 Schéma zapojení gyroskopu A3G4250D.....	i
Obr. 46 Návrh desky plošných spojů pro gyroskop A3G4250D; a) pohled shora; b) pohled zdola	i
Obr. 47 Schéma zapojení adaptéru pro bluetooth modul RN42.....	ii
Obr. 48 Návrh desky plošných spojů pro adaptér bluetooth modulu RN42	ii
Obr. 49 Schéma zapojení elektroniky pro pohyblivý modul	iii
Obr. 50 Návrh desky plošných spojů pro elektroniku pohyblivého modulu; a) pohled shora; b) pohled zdola	iii
Obr. 51 Finální pohyblivý modul a) pohled z jedné strany trubky; b) pohled z druhé strany trubky	iv

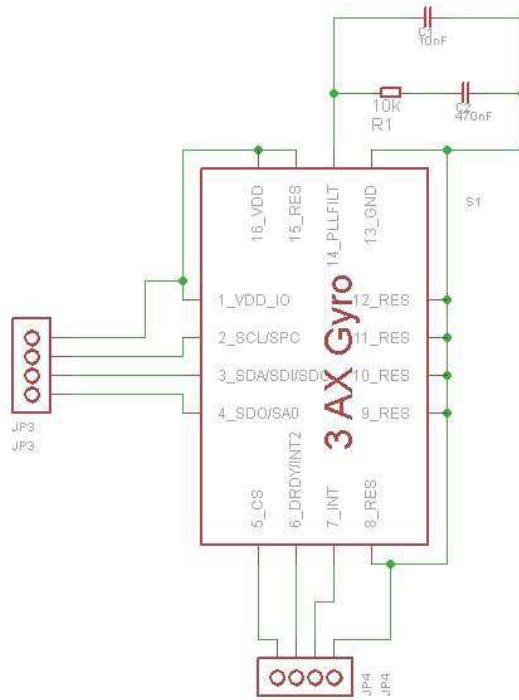
10. Seznam tabulek

Tab. 1 Základní parametry ATmega32 [12].....	23
Tab. 2 Základní parametry gyroskopu A3G4250D [13]	25
Tab. 3 Základní parametry prvně testovaného stejnosměrného motoru	30
Tab. 4 Základní parametry stejnosměrného motoru s převodovkou 050SHV-250-R.....	30
Tab. 5 Základní parametry bluetooth modulu RN42-I/RM [15].....	31
Tab. 6 Hodnota chybovosti při hodinovém kmitočtu 1MHz [21].....	32
Tab. 7 Výpočet maximálního úhlu stoupání pro různé hmotnosti modulu.....	49
Tab. 8 Měření maximálního úhlu stoupání pro různé hmotnosti modulu.....	49

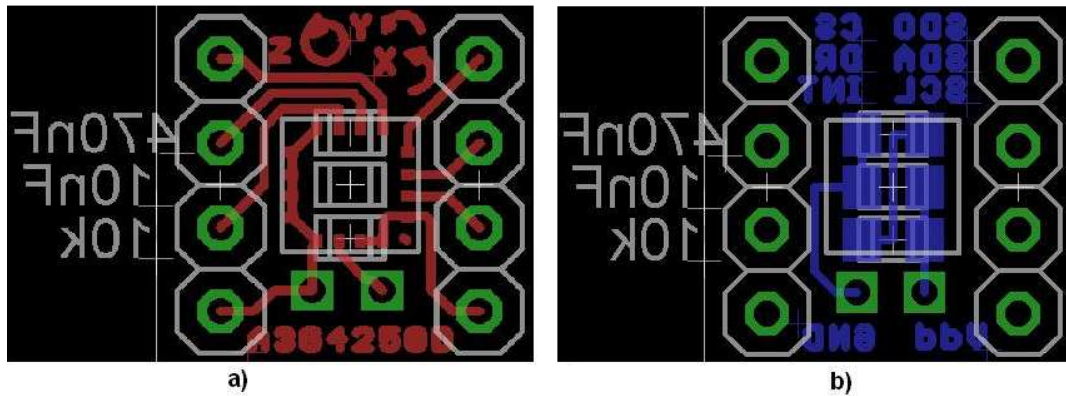
11. Seznam příloh

Příloha č. 1	Schéma zapojení gyroskopu A3G4250D	i
Příloha č. 2	Schéma zapojení adaptéru pro bluetooth modul RN42.....	ii
Příloha č. 3	Schéma zapojení elektroniky pro pohyblivý modul.....	iii
Příloha č. 4	Vlastní zařízení.....	iv
Příloha č. 5	Seznam součástek.....	v
Příloha č. 6	Zdrojový kód programu pro řídicí jednotku celého modulu	vi

Příloha č. 1 Schéma zapojení gyroskopu A3G4250D

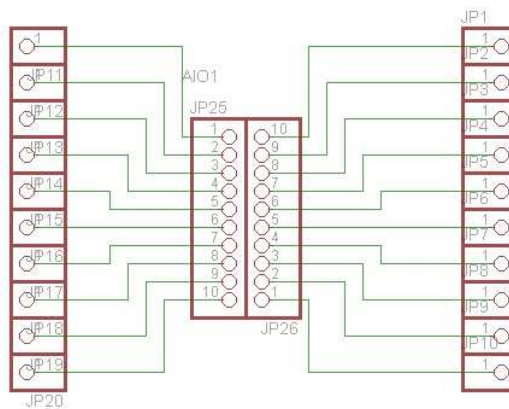


Obr. 45 Schéma zapojení gyroskopu A3G4250D

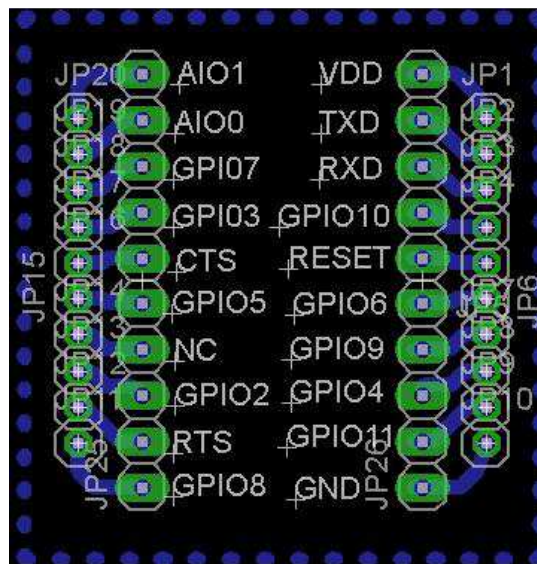


**Obr. 46 Návrh desky plošných spojů pro gyroskop A3G4250D;
a) pohled shora; b) pohled zdola**

Příloha č. 2 Schéma zapojení adaptéru pro bluetooth modul RN42

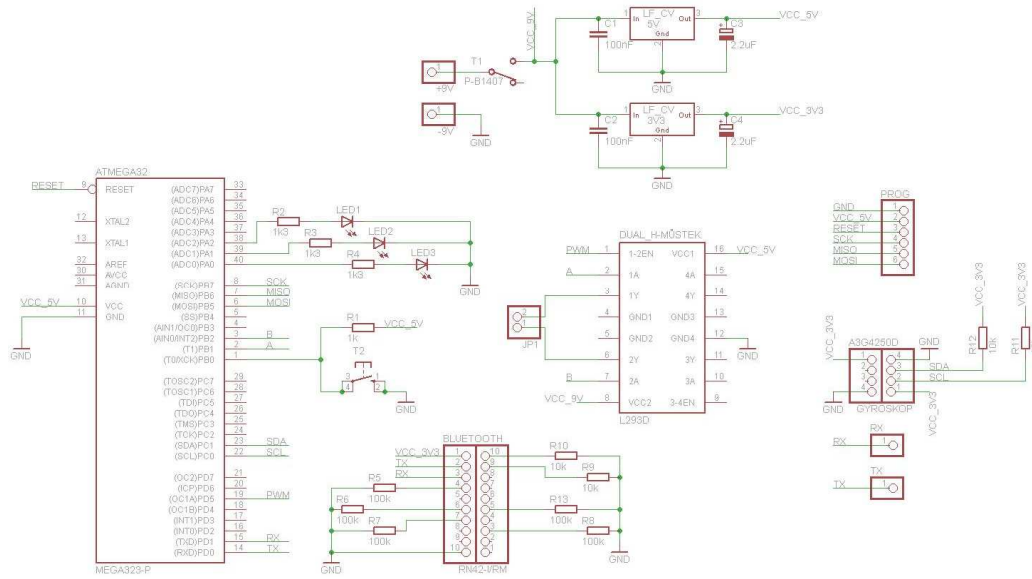


Obr. 47 Schéma zapojení adaptéru pro bluetooth modul RN42

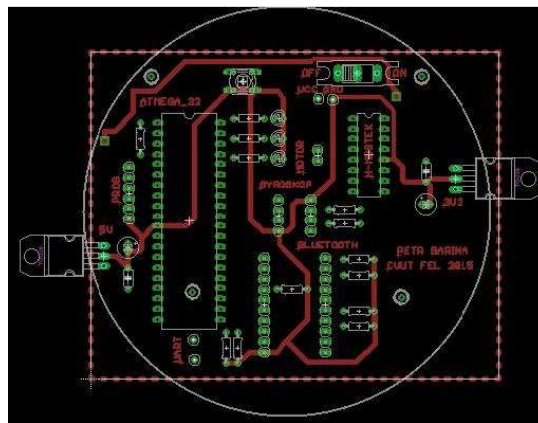


Obr. 48 Návrh desky plošných spojů pro adaptér bluetooth modulu RN42

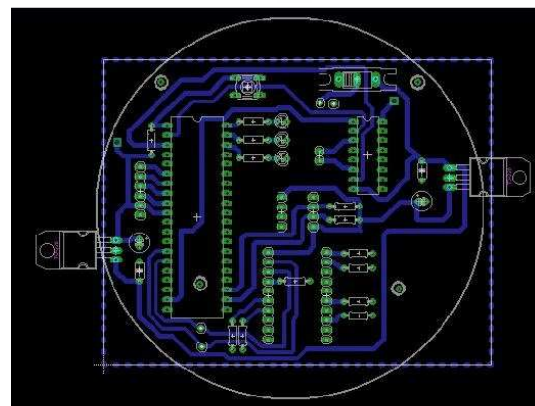
Příloha č. 3 Schéma zapojení elektroniky pro pohyblivý modul



Obr. 49 Schéma zapojení elektroniky pro pohyblivý modul



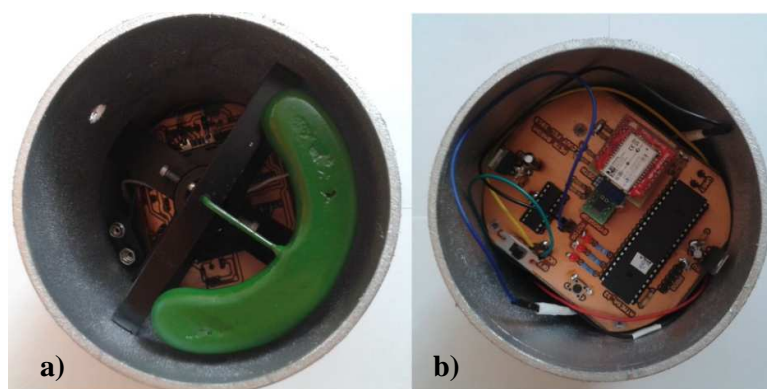
a)



b)

Obr. 50 Návrh desky plošných spojů pro elektroniku pohyblivého modulu; a) pohled shora; b) pohled zdola

Příloha č. 4 Vlastní zařízení



Obr. 51 Finální pohyblivý modul
a) pohled z jedné strany trubky; b) pohled z druhé strany trubky

Příloha č. 5 Seznam součástek

Označení	Hodnota/typ	Pouzdro	Popis
ATMEGA32	MEGA323-P	DIL40	mikrokontrolér AVR
T1	P-B1407	-	posuvný spínač: 1 - pólový ON - ON
T2	TC-0103-T	B3F-10XX	mikrospínač: 1 - pólový spínací ON - OFF
5V	LF50CV	TO220	stabilizátor napětí 5V
3V3	LF33CV	TO220	stabilizátor napětí 3.3V
C1	100nF	C050-024X044	keramický kondenzátor
C2	100nF	C050-024X044	keramický kondenzátor
C3	2.2μF/50V	E2-5	elektrolytický kondenzátor
C4	2.2μF/50V	E2-5	elektrolytický kondenzátor
R1	100k	0204/7	rezistor
R2	100k	0204/7	rezistor
R3	100k	0204/7	rezistor
R4	10k	0204/7	rezistor
R5	10k	0204/7	rezistor
R6	100k	0204/7	rezistor
R7	100k	0204/7	rezistor
R8	1k	0204/7	rezistor
R9	1k3	0204/7	rezistor
R10	1k3	0204/7	rezistor
R11	1k3	0204/7	rezistor
R12	10k	0204/7	rezistor
R13	10k	0204/7	rezistor
LED1	-	LED 3mm	LED červená
LED2	-	LED 3mm	LED červená
LED3	-	LED 3mm	LED červená
DUAL_H-MŮSTEK	L293D	DIP 16	integrováný dvojitý H-můstek
BLUETOOTH	-	-	2x 10 pinů - dutinová lišta 2.54 mm
PROG	-	-	4 - pinový sokl 2.54 mm pro programátor
A3G4250D	-	-	sokl 8 pro MEMS gyroskop
JP1	-	-	2x oboustranný kolík pro DC motor
RX	-	-	oboustranný kolík pro RX UART
TX	-	-	oboustranný kolík pro TX UART
9V	-	-	oboustranný kolík pro baterii 9V
-9V	-	-	oboustranný kolík pro baterii 9V

Součástky, které nejsou přímou součástí řídicí elektroniky

Bluetooth modul RN42

MEMS gyroskop A3G4250D

pouzdro na baterii 9V

1x 9V baterie GP Rechargeable 200mAh, Ni-MH

DC motor 050SHV-250-R

2x propojovací drátek mezi baterií a DPS

2x propojovací drátek mezi DC motorem a DPS

Příloha č. 6 Zdrojový kód programu pro řídicí jednotku celého modulu

```
// Nacteni knihoven
#include <mega32.h>
#include <i2c.h>
#include <stdio.h>
#include <delay.h>

// Deklarace globalnich promennych = jsou dosazitelne vsemi funkcemi programu
int L = 0, H = 0, a_H=0, a_L=0;

// Funkce I2C sbernice
void I2C (void)
{
    //horni byte
    i2c_start();    // start bit
    i2c_write(0xD0); // inicializace zápisu
    i2c_write(a_H); // adresa registru, který se bude číst
    i2c_start();    // začátek I2C přenosu
    i2c_write(0xD1); // inicializace čtení - přečte hodnotu z registru a_H
    H = i2c_read(0); // uloží přečtenou hodnotu do proměnné H
    i2c_stop();     // stop bit

    //dolni byte
    i2c_start();
    i2c_write(0xD0);
    i2c_write(a_L);
    i2c_start();
    i2c_write(0xD1);
    L = i2c_read(0);
    i2c_stop();
}

// Zacatek hlavniho tela programu
void main(void)
{
    // Deklarace lokalnich promennych = jsou dosazitelne pouze v hlavnim tele programu
    int mod, rychlost=0, data_z = 0, gyro=0;

    // Inicializace vstupnich a vystupnich portu
    // Port A
    // Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=Out Bit1=Out Bit0=Out
    DDRA=0x07;
    PORTA=0x00;
    // Port B
    // Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=Out Bit1=Out Bit0=In
    DDRB=0x06;
    PORTB=0x00;
```

```

// Port C
// Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRC=0x00;
PORTC=0x00;
// Port D
// Bit7=In Bit6=In Bit5=Out Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRD=0x20;
PORTD=0x00;

// Timer/Counter 1 inicializace
TCCR1A=(1<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (1<<WGM11) |
(1<<WGM10);
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (1<<WGM12) | (0<<CS12) | (0<<CS11) |
(1<<CS10);
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// USART inicializace
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 4800
UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR) | (0<<UPE) | (0<<U2X) |
(0<<MPCM);
UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (1<<RXEN) | (1<<TXEN) | (0<<UCSZ2) |
(0<<RXB8) | (0<<TXB8);
UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) | (0<<USBS) | (1<<UCSZ1) |
(1<<UCSZ0) | (0<<UCPOL);
UBRRH=0x00;
UBRRL=0x0C;

i2c_init();
// aktivace vsech os gyra
i2c_start(); // start bit
i2c_write(0xD0); // inicializace zápisu
i2c_write(0x20); // inicializace registru, do kterého se bude zapisovat
i2c_write(0x4F); // zapis hodnoty 01001111 do registru REG1 = aktivace os
i2c_stop(); // stop bit
// Probliknuti tri LED - kontrola behu programu
PORTA.0=1;
PORTA.1=1;
PORTA.2=1;
delay_ms(1000);
PORTA.0=0;
PORTA.1=0;
PORTA.2=0;

```

```

while (1)
{
// Cteni po UART z PC
while (!(UCSRA &(1<<RXC))) // když není nic poslano z terminalu, tak se tento if provede
{
a_H=0x2D; // přiřazení adresy registru pro aktivaci osy Z do proměnné a_H
a_L=0x2C; // přiřazení adresy registru pro aktivaci osy Z do proměnné a_L
I2C (); // volání funkce I2C
data_z = ((H<<8)|(L)); // posun vlevo pro H a logický součin mezi H a L
gyro=data_z/134;

// Vložení hodnoty v gyru do kladné
if (gyro<0)
{
gyro=-gyro;
}
printf("\n Rychlost otáčení [cm/s]=%d", gyro);
delay_ms(100);

// Automatická regulace otáček dle zadané rychlosti
//Následující regulace se začne provádět, jakmile se zadá nějaká rychlost
if(rychlost!=0)
{
// Malá rychlost = nacti PWM
if (gyro<rychlost)
{
// Pokud je PWM větší než 03E0, tak ji už nenacítej nech ji hodnotu 03FC
if(OCR1A>=0x03E0)
{
OCR1A=0x03FC;
}
else
{
OCR1A+=0x0020;
}
}
//Velká rychlost = odečti PWM
if (gyro>rychlost)
{
OCR1A-=0x0020;
delay_ms(200);
}
}
else
{
OCR1A=0;
}
}
}

```

```

// Zapis modu do promenne mod
mod=UDR;
// POHYB MOTORU DOPREDU
if(mod== 'a')
{
    PORTA.0=1;
    delay_ms(500);
    PORTA.0=0;
    // nastaveni smeru
    PORTB.1=1;
    PORTB.2=0;
    OCR1A=0;
}
// STOP MOTORU
else if (mod== 'b')
{
    PORTA.1=1;
    delay_ms(500);
    PORTA.1=0;
    // stop
    PORTB.1=0;
    PORTB.2=0;
    OCR1A=0;
}
// POHYB MOTORU DOZADU
else if(mod== 'c')
{
    PORTA.2=1;
    delay_ms(500);
    PORTA.2=0;
    // nastaveni smeru
    PORTB.1=0;
    PORTB.2=1;
    OCR1A=0;
}
// Mod pro zadani rychlosti
else if (mod=='d')
{
    rychlost=300;
    while (rychlost==300)
    {
        if (UCSRA &(1<<RXC))// Když je něco poslano z terminalu, tak se tento if provede
        {
            rychlost=UDR;
        }
    }
}
}
}

```