

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

Katedra aplikované elektroniky a telekomunikací

BAKALÁŘSKÁ PRÁCE

Robotické motorové kity

Originál (kopie) zadání BP/DP

Abstrakt

Práce se zabývá ovládáním motorových modulů, které jsou k dispozici na Katedře aplikované elektroniky a telekomunikací, pomocí pulzně šířkové modulace. Je provedeno porovnání jednotlivých motorů, budičů nebo případně celých kompletů z hlediska řízení a dalších parametrů. Následně jsou napsány vzorové aplikace pro řízení jednotlivých budičů s motory a pojízdných celků. Pro všechny možnosti je pak vytvořena univerzální klikací aplikace pro operační systém Windows, pomocí které lze motory řídit. V závěru jsou popsány základní možnosti měření otáček

Klíčová slova

Pulzně šířková modulace, můstkový motorový budič, deska STM32 Nucleo

Abstract

The thesis deals with a control of motor modules available at the Department of Applied Electronic and Telecommunications using the pulse width modulation. Individual motors, motor drivers or whole assemblies are compared in terms of the control and other parameters. Subsequently, model applications for the control of individual drivers with motors and moving units are written. For all the options, a universal application for the Windows operating system is created to drive engines. At the end, the basic options of speed measurement are described.

Key words

Pulse width modulation, bridge driver, STM32 Nucleo board

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práci, je legální.

.....

podpis

V Plzni dne 4.6.2017

Marek Novák

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce Ing. Petru Weissarovi, Ph.D. za užitečné rady a poskytnutý hardware.

Obsah

OBSAH	7
ÚVOD	1
SEZNAM SYMBOLŮ A ZKRATEK	2
1 HARDWARE	3
1.1 MOTORY	3
1.1.1 Princip stejnosměrných motorů.....	3
1.1.2 Porovnání vybraných motorů	4
1.2 BUDIČE.....	4
1.2.1 Princip budičů	5
1.2.2 Porovnání vybraných budičů.....	5
1.3 OSTATNÍ VÝKONOVÉ OBVODY PRO ŘÍZENÍ MOTORŮ	13
1.3.1 6-step řízení	13
1.3.2 Krokování	14
1.3.3 Mikrokování	14
1.4 POROVNÁNÍ MECHANICKÝCH ŘEŠENÍ NA KAE.....	15
1.4.1 Sestava 1	15
1.4.2 Sestava 2.....	16
1.5 VÝVOJOVÁ DESKA STM32 NUCLEO.....	17
1.5.1 GPIO(General-purpose I/Os).....	17
1.5.2 RCC (Reset and clock control)	18
1.5.3 TIM (General-purpose timers)	18
1.5.4 USART (Universal synchronous asynchronous receiver transmitter).....	18
1.5.5 ADC (Analog-to-digital convertor)	19
2 SOFTWAREVÉ ŘÍZENÍ MOTORŮ	20
2.1 APLIKACE PRO OVLÁDÁNÍ MOTORŮ.....	20
2.2 OBSLUŽNÉ FUNKCE PRO ŘÍZENÍ BUDIČŮ	21
2.2.1 Knihovna pro AD převodník.....	21
2.2.2 Knihovna pro sériovou komunikaci	21
2.2.3 Hlavní knihovna pro řízení PWM.....	22
2.2.4 Obslužná knihovna	24
2.3 SOFTWAREVÉ ŘÍZENÍ PRO SHIELD S BUDIČEM VNH5019	24
2.4 SOFTWAREVÉ ŘÍZENÍ PRO SHIELD S BUDIČEM DVR-8833	26
2.5 SOFTWAREVÉ ŘÍZENÍ PRO ROBOTA ZUMO S BUDIČEM DVR-8835.....	28
2.6 SOFTWAREVÉ ŘÍZENÍ PRO SHIELD X-NUCLEO-IHM04A1 S BUDIČEM L6206	28
3 VÝHLED DALŠÍHO VYUŽITÍ	30
3.1 MĚŘENÍ OTÁČEK POMOCÍ MAGNETICKÝCH SENZORŮ	30
3.1.1 Princip Hallova jevu.....	30
3.2 MĚŘENÍ OTÁČEK POMOCÍ STROBOSKOPICKÉHO JEVU	30
3.2.1 Princip měření otáček pomocí stroboskopického jevu.....	30
3.3 MĚŘENÍ OTÁČEK POMOCÍ OPTICKÉ ZÁVORY	31
3.3.1 Princip optické závory.....	31
ZÁVĚR	32
SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ	34
PŘÍLOHY	35

Úvod

Cílem bakalářské práce je vyrobit vzorové aplikace s malými motory vhodné pro malé pohyblivé roboty s ohledem na dostupnost softwaru a hardwaru na katedře aplikované elektrotechniky a telekomunikací. Dále porovnat vybrané dostupné můstkové motorové budiče a motory v práci použité. Konkrétně se jedná o dva robotické celky obsahující v základu budič a motory. A další dva samostatné budiče s motorem. Celkově čtyři různé budiče a tři motory. Všechny budiče jsou pak řízeny pomocí vývojové desky STM32-nucleo.

Každý budič je napájen stejnosměrným napětím v rozsahu 5-30 V a dodává potřebný proud pro motory, které jsou k němu připojeny. Samotný budič je pak připojen na vývojovou desku, která je naprogramována příslušným programem pro jeho řízení. Pro každý budící můstek musí být naprogramována vlastní aplikace, protože každý budič je řízen jiným způsobem. Budiče umožňují chod motorů ve směru chodu hodinových ručiček nebo v protisměru a režim brždění. I zde jsou různé možnosti. Například režimu brždění lze dosáhnout přepólováním napětí na motoru nebo uzemněním motoru. Samotný čip budiče je pak vždy součástí vývojové desky, kterou lze připojit například na Arduino konektory nacházející se na desce STM32 Nucleo. Použité motory se pak liší v převodovce, maximálních dosažitelných otáček, momentu a v neposlední řadě ve tvaru a mechanickém řešení.

V následujících kapitolách budou porovnány jednotlivé použité můstkové budiče, motory a principy dalšího možného řízení motorů. Následně bude uveden popis vývojové desky STM32 Nucleo a popsání použitých periférií a robotických celků. Poté bude v práci věnován prostor samotnému popisu vzorových aplikací a univerzální aplikaci pro ovládání všech motorů. V závěru pak je teoretický rozbor snímačů, kterými lze měřit otáčky na použitých motorech, případně v jiných technických aplikacích.

Seznam symbolů a zkratek

PWM.....	Pulzně šířková modulace
AD převodník.....	Analogově digitální převodník
Usart.....	Synchronní/Asynchronní sériové rozhraní
STM32-Nucleo.....	Vývojová deska
STM32F411RE.....	Čip na desce STM32-Nucleo
KAE.....	Katedra aplikované elektrotechniky a telekomunikací
BLDC motor.....	Bezkartáčový stejnosměrný motor
6-step.....	Šesti-fázové řízení motorů

1 Hardware

V práci je použito několik různých motorů, budičů a mechanických řešení podvozků, ze kterých vznikají jednotlivé kombinace malých robotů. Každá kombinace je ovládána pomocí vývojové desky STM32-Nucleo osazená čipem STM32F411RE. V následujících kapitolách se budu zabývat jednotlivými použitými komponenty.

1.1 Motory

Motor přeměňuje elektrické energie na energii mechanickou. Většina dnešních motorů funguje na elektromagnetickém principu. Napájeny jsou ze stejnosměrného zdroje napětí zpravidla tužkových baterek, případně je možné připojit síťový adaptér mající na výstupu stejnosměrné napětí vhodné velikosti. Toho lze s výhodou využít hlavně při samotném testování a programování. V následující kapitole proberu princip stejnosměrných motorů.

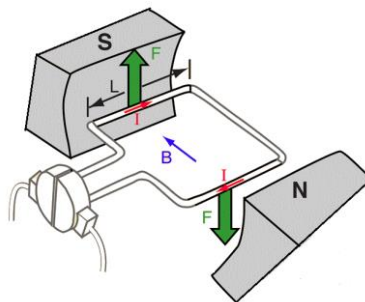
1.1.1 Princip stejnosměrných motorů

Pro vysvětlení použiji příklad se dvěma lamelami a jedním závitem. V reálných aplikacích se ale běžně používá daleko větší počet lamel i závitů.

Pokud závitem rotoru prochází elektrický proud, pak na tento vodič působí síla podle vztahu (1), kde F je působící síla, B je magnetická indukce a L je indukčnost vodiče.

$$F = B \times I \times L \quad (1)$$

Směr, kterým bude síla působit, lze zjistit podle Flemingova pravidla levé ruky. To říká, že položíme-li dlaň levé ruky tak, aby prsty ukazovaly směr protékajícího proudu a do dlaně vstupovaly indukční čáry, pak palec ukáže směr síly, kterou působí magnetické pole na vodič. Proud, který teče přes komutátor a dále přes vinutí, vybudí kolem rotorového vinutí magnetické pole. A vzájemným působením magnetického pole rotoru a statoru začne působit síla na rotor a ten se začne natáčet tak, aby vyrovnal severní pól statoru s jižním pólem rotoru. Aby docházelo k trvalému otáčení, tak je v obvodu připojen komutátor, který komutuje procházející proud cívkou a tím je docíleno stavu, kdy póly rotoru a statoru se nikdy nesrovnají a motor se bude trvale otáčet. Popsaný princip je vidět na *obrázku 1.1* [2],[3]



Obrázek 1.1 - princip stejnosměrného motoru

1.1.2 Porovnání vybraných motorů

V této kapitole se budu zabývat jednotlivými motory dostupnými na KAE. Jedná se o motory POLOLU-1124 a POLOLU-1118 od firmy Pololu a Micro Metal Gearmotor HP 6V také od firmy Pololu.

POLOLU-1124

Prvním použitým motorem je POLOLU-1124. Jedná se o stejnosměrný kartáčový motor s napájecím napětím v rozmezí 3-12 V. Maximální proud je 800 mA a pracovní 80 mA. Převodovka je plastová s oboustrannou hřídelí o průměru 3 mm a vytvarovaná do zářezu D. Převod převodovky je 120:1 s 120 ot./min. [6]

POLOLU-1118

Druhým motorem je motor POLOLU-1118. Jedná se o stejnosměrný motor s napájecím napětím v rozsahu 3-12 V. Maximální přípustný proud je 800 mA a pracovní 70mA. Převodovka je plastová s oboustrannou hřídelí o průměru 7 mm a zploštělá z obou stran. Převod převodovky je 228:1 s 45 ot./min. Na *obrázku 1.2* je motor vyfocen. [7]



Obrázek 1.2 - motor POLOLU-1118

Micro Metal Gearmotor HP 6V

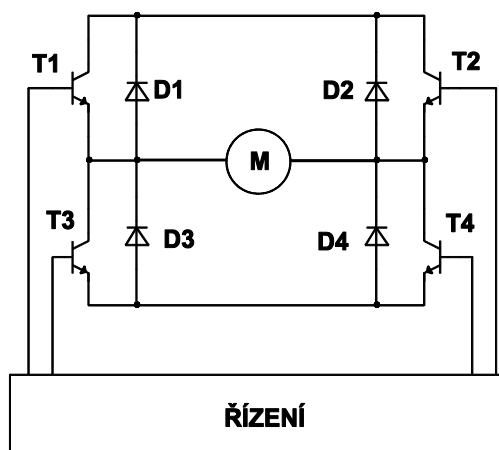
Třetím motorem je motor Micro Metal Gearmotor HP 6 V. Jedná se o převodový stejnosměrný motor s napájecím napětím 6 V. Při 6 V napájecího napětí bez zatížení má 400 ot/min a proudový odběr 120 mA. Maximální proud činí 1,6 A. Převodovka je kovová s převodovým poměrem 75:1 s 3 mm hřídelí do tvaru D délky 9 mm a průměru 3 mm. Maximální kroutící moment motoru je 0,16 N. m. [14].

1.2 Budiče

V této kapitole budu řešit použité budiče dostupné na KAE pro řízení DC motorů. Jedná se o budič VNH5019 od firmy STMicroelektronik, DVR8833 a DVR8833 od firmy Texas Instruments a X-NUCLEO-IHM041A1 od firmy STMicroelektronik.

1.2.1 Princip budičů

Budiče používané v této bakalářské práci pracují na principu H-můstku, a proto bude princip vysvětlen právě na něm. Můstek může být v polomůstkovém zapojení, ale v praxi není příliš používáno, protože jím nelze řídit motor oběma směry a je třeba dvou zdrojů napájení. Výhodou jsou pouhé dva spínače. Další možností je plné můstkové spojení. Zde je již použit pouze jeden napájecí zdroj a lze motor řídit oběma směry.



Obrázek 1.3 – schéma plného H-můstku

Princip funkce je vysvětlen podle *obrázku 1.3*. Jednotlivé tranzistory jsou otevřené nebo zavřené. Při můstkovém řízení se tranzistory spínají takzvaně křížem. To znamená, že se nejdříve sepne tranzistor T1 a T4 a proud teče od kladného pólu přes T1, motor a T4 k pólu zápornému. T2 a T3 jsou v tomto kroku zavřeny. V druhé fázi jsou sepnuty tranzistory T2 a T3 a proud teče opět od kladného pólu zdroje přes T2, motor a T3 do pólu záporného. V tomto kroku jsou naopak uzavřeny tranzistory T1 a T4. Protože motor obsahuje vinutí, tak nemůže proud tekoucí v jednom taktu zaniknout okamžitě. Proto v době, kdy se přepíná z jedné dvojice tranzistorů na dvojici druhou, musí být proud veden vybíjecími diodami D1, D2, D3, D4. Po vypnutí T1 a T4 vedou diody D2 a D3 a při vypnutí T2 a T3 vedou diody D1 a D4. [1]

1.2.2 Porovnání vybraných budičů

VNH5019

Prvním použitým budičem je budič VNH2019 od firmy STMicroelektronik. Veškeré informace o funkci, zapojení a vlastnostech v této kapitole jsou čerpány z dokumentace výrobce [4]. V práci je použita dvojice těchto budičů na jednom shieldu od firmy Pololu, který má možnost být připojen na Arduino konektory, které jsou na desce STM32-Nucleo, čehož bylo při realizaci bakalářské práce využito. Lze ho provozovat i bez Arduino konektorů. Pro tyto účely jsou na desce vyvedené piny. Budič může pracovat v rozmezí napájecího napětí 5,5 – 24 V a dodat trvale 12 A, případně 30 A proudu špičkového. Další možností je připojení pouze jednoho motoru. V tomto případě je dvojice budičů schopna dodat 24 A trvalého

proudu a 60 A proudu špičkového. Maximální frekvence PWM na každý motor je bez rozdílu počtu připojených motorů 20 kHz a to s takzvaným tichým režimem (ultrasonic PWM).

Budič funguje na principu úplného čtyřkvadrátorového budiče, kde jsou spínány jednotlivé mosfet tranzistory pomocí signálů M1INA, M1INB a M1ENA/B pro motor 1. Rychlost je řízena pomocí PWM signálu M1PWM pro motor 1. Podrobný princip čtyřkvadrátorového budiče je uveden v kapitole 1.2.1.

Řízení probíhá skupinou řídicích signálů pro každý motor zvlášť. Ty jsou posílány pomocí mikrokontroleru na řídicí piny na desce. Jednotlivé signály jsou v *Tabulce 1.1*

Tabulka 1.1 - řídicí signály budiče VNH2019

M1INA	Vstupní řídicí signál A motoru 1
M1INB	Vstupní řídicí signál B motoru 1
M2INA	Vstupní řídicí signál A motoru 2
M2INB	Vstupní řídicí signál B motoru 2
M1EN/DIAG	Povolení vstupů A, B motoru 1
M2EN/DIAG	Povolení vstupů A, B motoru 2
M1PWM	Vstup pro PWM řízení rychlosti motoru 1
M2PWM	Vstup pro PWM řízení rychlosti motoru 2

Deska obsahuje i výstupní signály popsané v *Tabulce 1.2*. Signály MxOUTx jsou na desce připojené k LED diodám, které pomocí svitu zelené nebo červené LED diody signalizují směr chodu příslušného motoru. Zelená signalizuje chod motoru po směru hodinových ručiček a červená chod motoru v protisměru. Signály MxCS slouží jako analogový výstup a obsahují data o velikosti proudu, který budič dodává do motoru. Například tak lze indikovat zaseknutí kola a jiné nežádoucí stavy.

Tabulka 1.2 - výstupy budiče VNH2019

M1CS	Analogový výstup motoru 1
M2CS	Analogový výstup motoru 2
M1OUTA	Výstup A motoru 1
M1OUTB	Výstup B motoru 1
M2OUTA	Výstup A motoru 2
M2OUTB	Výstup B motoru 2

Samotný motor je pak kontaktován pomocí vodičů do šroubovací lišty na piny, které jsou popsány v *Tabulce 1.3*. Trojice pinů starajících se o napájení je popsána v *Tabulce 1.4*.

Tabulka 1.3 - kontakty motorů na budiči VNH2019

M1A	Vstup řídicího signálu A do motoru 1
M1B	Vstup řídicího signálu B do motoru 1
M2A	Vstup řídicího signálu A do motoru 2
M2B	Vstup řídicího signálu B do motoru 2
VIN	Napájecí vstup obou motorů
GND	Zem obou motorů

Tabulka 1.4 - napájecí piny pro budiče VNH2019

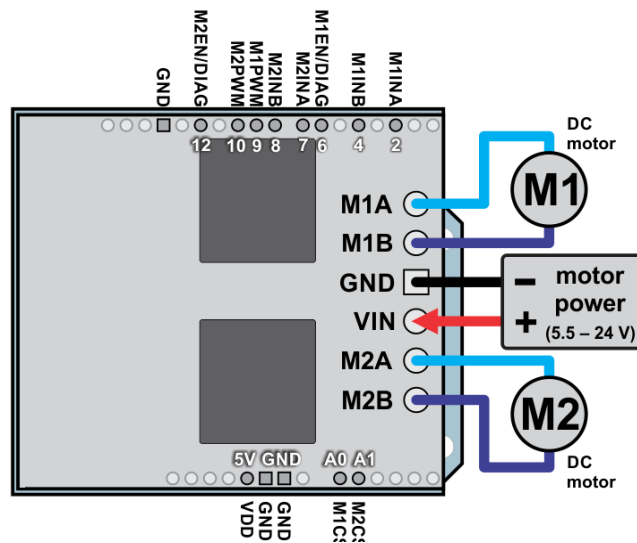
VDD	Napájecí vstup pro desku s budiči
GND	Zem pro desku s budiči
VOUT	Výstupní napětí za ochranným obvodem pro napájení dalších periférií

Motory se řídí kombinací vstupních řídicích signálů, které jsou popsány v *Tabulce 1.5*. Jednotlivými kombinacemi lze dosáhnout čtyř stavů, ve kterých se jednotlivé motory mohou nacházet. Jednotlivé stavy jsou: točení motoru po směru hodinových ručiček, točení motoru v protisměru hodinových ručiček a dva režimy brždění. Nastavení jednotlivých pinů pro dosažení jednotlivých režimů motorů ukazuje *Tabulka 1.5*. Tabulka je uvedena pro motor 1. Pro druhý motor by byla obdobná.

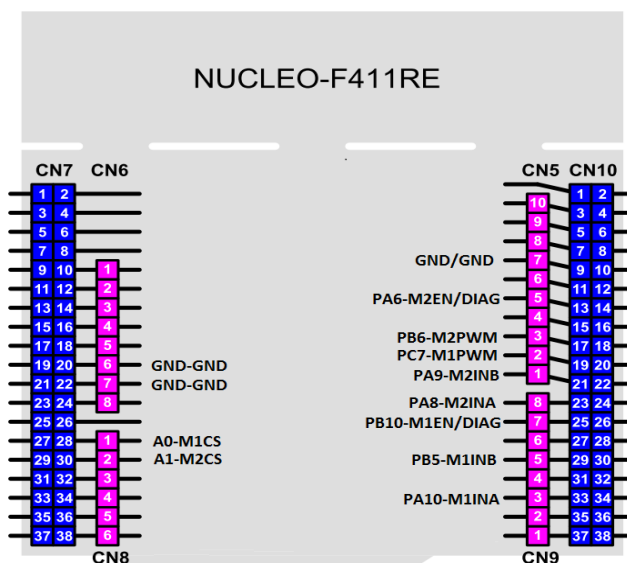
Tabulka 1.5 - nastavení režimů motoru

M1INA	M1INB	M1ENA/B	REŽIM
0	0	1	Brzda (to GND)
0	1	1	Točení motoru v protisměru hodinových ručiček
1	0	1	Točení motoru po směru hodinových ručiček
1	1	1	Brzda (to Vcc)

Samotné propojení motorů s deskou budičů je na *obrázku 1.4*. Propojení pinů budiče a desky STM32 Nucleo je na *obrázku 1.5*.



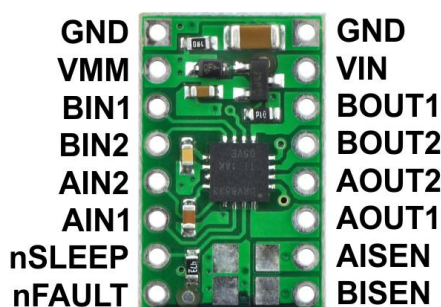
Obrázek 1.4 - připojení motorů k budiči



Obrázek 1.5 - propojení budiče s deskou STM32 Nucleo

DVR 8833

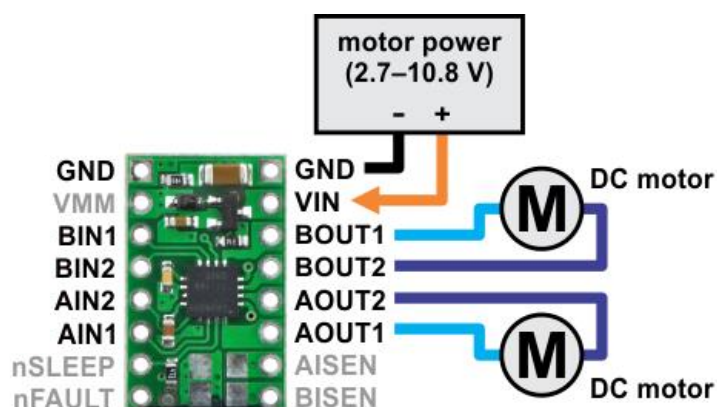
Dalším testovaným budičem je budič DVR 8833. Veškeré informace o funkci, zapojení a vlastnostech v této kapitole jsou čerpány z dokumentace výrobce [5]. Jedná se o zapojení s dvěma H můstky pro řízení dvou stejnosměrných kartáčových motorů v obou směrech otáčení. Nebo jeden bipolární krokový motor v napájecím rozmezí od 2,7 V do 10,8 V. Budič je schopen dodat trvale 1,2 A na jeden kanál nebo špičkově 2 A na kanál po dobu několika sekund. Případně 2,4 A při trvalém chodu a 3 A ve špičce při zapojení jednoho motoru. Maximální dosažitelná frekvence PWM je 50 KHz. Budič také obsahuje ochranu proti podpětí, nadměrnému proudu, přehřátí a zkratu. Tento budič obsahuje čtyři vstupní, čtyři výstupní piny, napájecí piny, pin pro uspání budiče a pin signalizující podpětí, velký proud a přehřátí. Na obrázku 1.6 je vidět rozmístění pinů budiče na desce.



Obrázek 1.6 - rozmístění pinů na budiči

Levá strana je v typických aplikacích použita pro řídicí signály z mikrokontroleru a strana pravá pro připojení motorů. Vstupy GNG jsou země. Výstup VMM slouží jako výstupní napětí pro další periférie obvodu. Napětí na tomto pinu je již za ochrannými obvody. BIN1 a BIN2 slouží k ovládání směru, rychlosti a brzdění prvního motoru podle kombinací popsané v Tabulce 1.6. AIN2 a AIN1 plní stejnou funkci jako BIN1 a BIN2, ale pro druhý motor. Vstup nSLEEP se používá, pokud chceme uvést čip do režimu spánku. To

znamená, že se vyresetují nastavené hodnoty, vypnou vnitřní hodiny, deaktivuje H můstek atd. NFAULT je výstupní pin, který signalizuje kritické stavy. V případě DRV8833 je to vysoký proud (Overcurrent) nebo vysoká teplota (Thermal Shutdown-TSD). Pokud jeden z vyjmenovaných stavů nastane, tak na jeho výstupu bude logická nula. AISEN má na zadní straně desky dvě pájecí plošky. Ty slouží k připojení smd 1206 rezistoru pro omezení proudu. Ve výchozím stavu je odpor nepřipojen a AISEN připojen na zem. Velikost odporu lze spočítat podle instrukcí v dokumentaci. BISEN má stejný účel jako AISEN, ale pro první motor. Vstup VIN slouží jako napájecí pin pro samotné motory. Výstupy BOUT1 a BOUT2 slouží k připojení prvního motoru. Výstupy AOUT2 a AOUT1 mají stejný účel jako BOUT1 a BOUT2, ale pro druhý motor. Připojení motorů k budiči je na *obrázku 1.7*. Praktické zapojení je pak vyfoceno na *obrázku 1.9*.

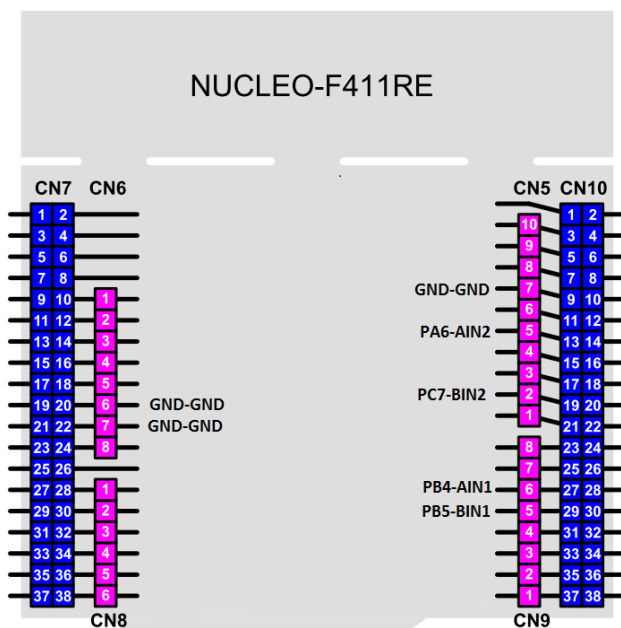


Obrázek 1.7 - připojení motorů k budiči

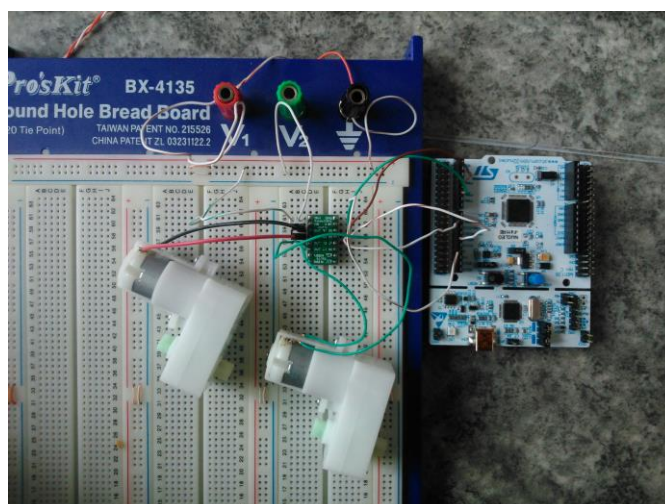
Tabulka 1.6 - kombinace pro řízení motorů

xIN1	xIN2	FUNKCE
PWM	0	Jízda dopředu (fast decay)
0	PWM	Jízda dozadu (slow decay)
1	PWM	Jízda dozadu (fast decay)
PWM	1	Jízda dopředu (slow decay)
1	1	Brzda

Propojení desky STM32 Nucleo a budiče je zobrazeno na *obrázku 1.8*. GPIO porty mikrokontroleru jsou nastaveny jako výstupní a propojeny se vstupními piny BINx a AINx určenými k řízení motorů a GND mikrokontroleru se propojí s GND budiče. Dále je ještě samozřejmě možné řízení nSLEEP a zjišťování chyby z nFAULT ale to není v této práci použito. Na výstupy BOUTx a AOUTx jsou připojeny motory a na piny GND a VIN je připojeno napájení vhodné velikosti pro motory.



Obrázek 1.8 - propojení desky budiče a STM32 Nucleo

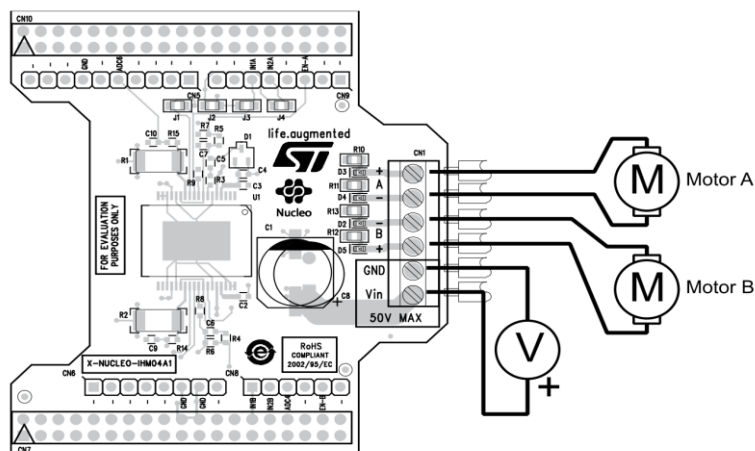


Obrázek 1.9-Realizace s budičem DVR 8833

L6206

Tento budič v pouzdře PowerSO36 od výrobce STMicroelektronik je v této bakalářské práci osazen na desce X-NUCLEO-IHM4A1 téhož výrobce. Veškeré informace o funkci, zapojení a vlastnostech v této kapitole jsou čerpány ze zdrojů [9][10]. Opět je založen na principu plného H-můstku. Je možné řídit stejnosměrné bipolární nebo čtyřpólové stejnosměrné motory. Napájecí napětí je v rozsahu 8-50 V. Maximální proud může dosáhnout hodnoty 2,8 A. Obsahuje blokování podpětí, nadproudovou a nadteplnou ochranu. Celý shield je kompatibilní s Arduino UNO R3 a deskou SMT32 Nucleo. Další možnosti použití je v kombinaci čtyř spojek, pomocí kterých lze dosáhnout až šesti různých režimů. Ty se liší v počtech připojených motorů nebo v druhu připojených motorů. V této práci je použit MOD1, který umožňuje bez

připojení spojek připojit dva stejnosměrné motory s možností řízení otáček v obou směrech otáčení. Schéma připojení motorů je na *obrázku 1.10*.



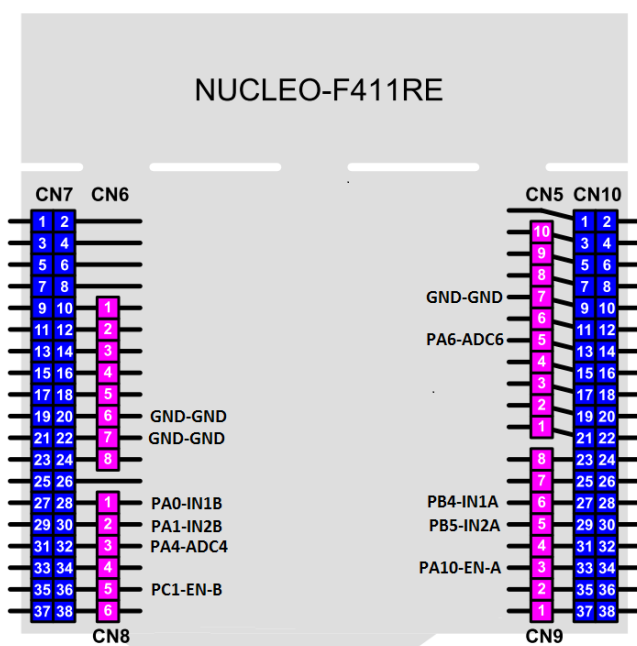
Obrázek 1.10 - připojení motorů k desce s budičem

Deska obsahuje šest řídicích pinů a dva piny jako analogový výstup pro snímání proudu. Dále jsou zde 4 svorky pro připojení motorů a dvě svorky napájení pro motory. Pin EN-B slouží k povolení můstku B a pin EN-A k povolení můstku A. Pin IN1A a IN2A slouží k ovládání směrů, brždění a rychlosti příslušného motoru pomocí kombinace nul, jedniček a PWM. Jednotlivé kombinace jsou v *Tabulce 1.7*. Pin ADC6 slouží jako analogový výstup pro měření proudu do motoru A a pin ADC4 slouží ke stejnému účelu pro motor B. Piny GND slouží jak zem a jsou připojeny na piny GND Nuclea. Na svorkovnici se nachází GND a VIN. Tyto svorky slouží k připojení napájení příslušné velikosti. V této práci jsem použil síťový adaptér s napájecím napětím 18 V. Dále je tu svorka A+ a A-, do které se připojuje samotný motor A. B+ a B- slouží ke stejnému účelu pro motor B.

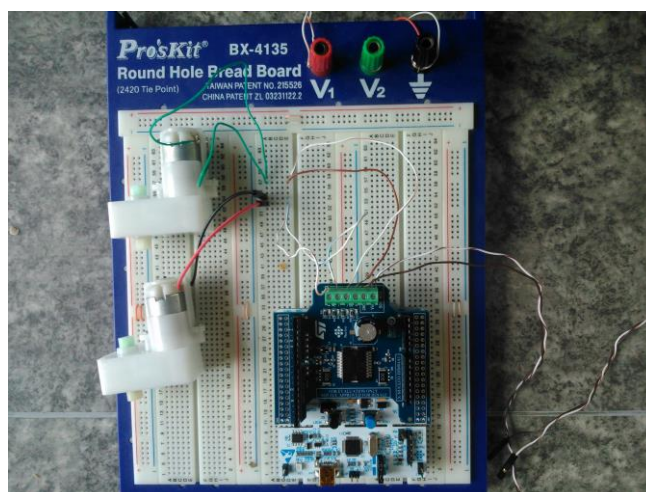
Tabulka 1.7 - kombinace pro řízení motorů

EN-x	IN1x	IN2x	funkce
1	PWM	0	Jízda vpřed
1	0	PWM	Jízda vzad
1	0	0	Brzda
0	X	X	vysokoimpedanční stav

Propojení budiče s deskou STM32 Nucleo je na *obrázku 1.11*. Praktická realizace je pak vyfocena na *obrázku 1.12*. GPIO je nastaven jako výstup pro vstupní piny budiče EN-x, IN1x a IN2x, aby mohly být motory řízeny. GPIO v nastavení jako vstup je pro výstupní piny budiče ADC4 a ADC6, pomocí kterých se odesílají do Nuclea hodnoty pro výpočet proudu. Všechny zemní svorky na Nucleu a budiči jsou propojeny.



Obrázek 1.11 - propojení desky STM32 Nucleo a budiče



Obrázek 1.12-realizace s budičem L6206

DVR-8835

Jedná se o dvojitý H-můstek pro řízení dvou stejnosměrných motorů nebo jednoho krokového motoru. Veškeré informace o budiči jsou ze zdroje [11]. Maximální napájecí napětí je 11 V. Na jeden motor je schopen nepřetržitě dodat 1,2 A nebo špičkově po dobu pár sekund až 1,5 A. Pokud se připojí výstupy paralelně k jednomu motoru, tak trvalý proud může dosahovat hodnoty 2,4 A nebo 3 A špičkově po dobu několika málo sekund. Maximální frekvence PWM na výstupu je 250 KHz. Budič může pracovat ve dvou provozních režimech. Režim IN/IN je složitější a umožňuje pokročilejší možnosti ovládání. Druhý režim se nazývá PHASE/ENABLE a je jednodušší a výrobcem doporučovaný.

Budič obsahuje pin VCC určený pro napájení samotného čipu a zemnicí pin GND. Jeden motor je připojen na výstupní piny AOUT1 a AOUT2 a druhý motor na BOUT1 a BOUT2. Pro řízení motoru A je

určen AIN1/APHASE a AIN2/AENBL, pro druhý motor pak BIN1/BHASE a BIN2/BENBL. Poslední pin se jmenuje MODE. Tento pin určuje, jaký mód budiče bude použit. V základním neošetřeném stavu je pin v logické 0 a to odpovídá modu IN/IN. Toto nastavení používá označení xINx pro ovládací piny. Pomocí jejich kombinace lze dosáhnout čtyř stavů na výstupu. Jednotlivé kombinace a stavy na výstupu popisuje *Tabulka 1.8*. Druhým možným režimem je režim Phase/Enable. Tento mód využívá stejné řídicí piny jako IN/IN, ale má je pojmenované xPHASE a xENBL. Zde je možné dosáhnout jen tří základních stavů na výstupu budiče. Jednotlivé kombinace vstupů a jim odpovídající výstupy jsou v *Tabulce 1.9*. Jednotlivé propojení jsou popsány v kapitole 1.4.2, protože budič je zapojen do robotického celku o kterém kapitola pojednává.

Tabulka 1.8 - tabulka modu IN/IN

xIN1	xIN2	funkce
0	0	Coast
PWM	0	Jízda dopředu
0	PWM	Jízda dozadu
1	1	Brzda

Tabulka 1.9 - tabulka modu PHASE/ENABLE

xPHASE	xENBL	Funkce
0	PWM	Jízda dopředu
1	PWM	Jízda dozadu
X	0	Brzda

1.3 Ostatní výkonové obvody pro řízení motorů

V této kapitole budu řešit další možnosti řízení motorů, které se nachází například na vývojových deskách STM32 Nucleo. Konkrétně se bude jednat o 6-step řízení pro BLDC motory a standardní krokování s mikrokrokováním pro krokové motory.

1.3.1 6-step řízení

Šestifázová komutace nebo také 6-step řízení je nejčastější způsob řízení BLCD motorů. Metoda je založena na principu komutace stejnosměrného proudu cívkou. Ale aby mohlo dojít ke komutaci, je třeba znát polohu rotoru. Ta se zjišťuje pomocí Hallovy sondy, která je umístěna na každé fázi motoru a snímá velikost magnetického pole permanentních magnetů rotoru. Tato metoda je ale nákladná, a proto se často přistupuje k další možnosti, která je realizována bez pomoci Hallových sond.

Řízení bez použití Halovy sondy spočívá ve sledování průběhu indukovaného napětí. Při této metodě jsou připojeny jen dvě fáze motoru ze tří. Třetí fáze je použita právě pro měření indukovaného napětí, ze kterého lze odvodit polohu rotoru. Odvození polohy rotoru může fungovat například na principu, kdy se detekuje průchod indukovaného napětí v odpojené fázi nulou (tzv. Zero-cross detection). Při průchodu nulou

se rotor nachází v polovině komutačního kroku a v tento okamžik se odměří další půlka kroku. Tím je zjištěn vhodný okamžik pro komutaci. [16]

1.3.2 Krokování

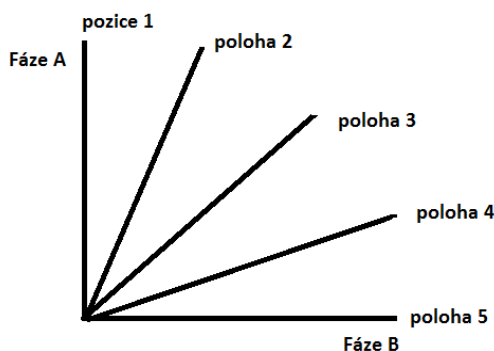
Popíši zde tři metody řízení motorů pomocí krokování. První metoda je čtyřtaktní řízení s magnetizací jedné fáze. Při této metodě jsou vždy napájeny jednotlivé fáze A, B, C, D. Pokud je vybuzena jedna z cívek, tak vzniká elektromagnetické pole v ose pólových nástavců, na kterém je namotáno příslušné vinutí cívky. Následně dojde k přitažení pólového nástavce na rotoru. Při stálém střídání buzení fází v sledu A – B – C – D – A, případně ve sledu opačném pro otáčení motoru opačným směrem.

Další možností je čtyřtaktní řízení s magnetizací dvou fází. Oproti předchozí metodě zde vždy dochází ke spínání dvou sousedních fází. Touto technikou je docíleno vychýlení rotoru přesně do polohy mezi tyto dvě fáze. Sled fází pro jeden směr je následující: AB – BC – CD – DA.

Poslední metodou je osmitaktní řízení krokového motoru. Jedná se o kombinaci předchozích dvou metod. V této metodě je postupně střídáno buzení jedné fáze a buzení dvou sousedních fází. To znamená kombinaci A – AB – B – BC – C – CD – D – AD. Oproti dvěma předchozím metodám je zde dosaženo dvounásobného počtu kroků na otáčku. Nevýhodou je rozdílná velikost momentu. Moment se liší, pokud je buzena jedna fáze nebo fáze dvě. Nedostatek lze vyřešit přizpůsobením velikostí proudů s ohledem na to, kolik fází je buzeno. Tato řešení ale vyžaduje složitější budící obvody. [15]

1.3.3 Mikro krokování

Tato metoda je využívána pro velmi přesné nastavení polohy krokového motoru. Oproti klasickému krokování lze jednotlivé kroky ještě zjemnit a to až na zlomky stupňů. Toho je potřeba například u tiskáren, robotů atd. Velikost kroku lze snížit počtem fází, ale to je velmi nákladné a řízení mnoha fází je složité. Proto se přistoupilo k řešení, kdy jsou jednotlivé fáze napájeny odlišným napětím a tím dochází k dosažení libovolné stabilní polohy mezi dvěma základními kroky. Jednoduchý příklad je na *obrázku 1.13*. Ve výchozí poloze je fáze A buzena proudem I a fáze B je nebuzena. V poloze 2 je fáze A buzena proudem I a fáze B je buzena proudem I/k . V poloze 3 je fáze A a B buzena proudem I . V poloze 4 je fáze A buzena proudem I/k a fáze B proudem I . V poledním kroku je fáze A nebuzena a fáze B je buzena proudem I . [15].



Obrázek 1.13-příklad pro mikro krokování

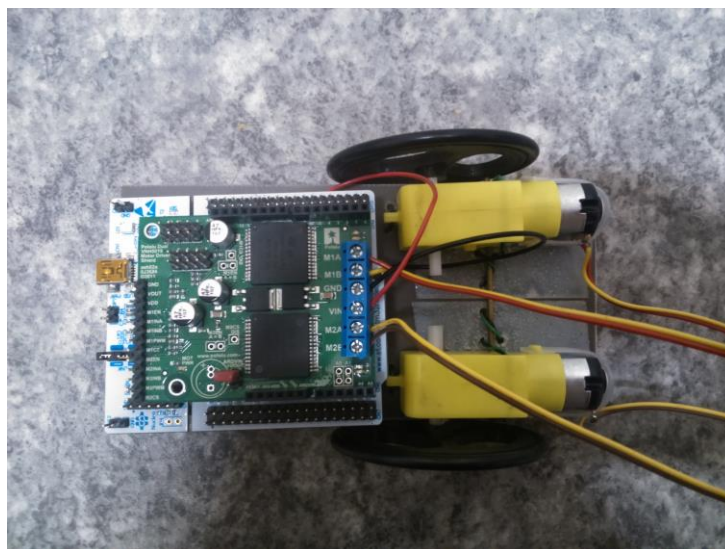
1.4 Porovnání mechanických řešení na KAE

Na KAE jsou k dispozici různé robotické sestavy. V této práci se budu zabývat dvěma z nich. Každý obsahuje v základu motory, budič a kola. Někdy je robot doplněn o další prvky. Například kompas, akcelerometr atd.

1.4.1 Sestava 1

Jedná o sestavu dvou motorů POLOLU-1124 s příslušnými koly, které jsou popsány v kapitole 1.1.2. Dále budič VNH5019, který je popsán v kapitole 1.2.1. a mikrokontroler NUCLEO-64 STM32F411RE popsán v kapitole 1.5. Pouzdro na baterie je klasické plastové pouzdro na 4 tužkové AA 1,5 V baterie. Samotný podvozek pak nese všechny komponenty desky. Na jednom konci podvozku se nachází dva axiální motory přidělané na třech místech pevně k podvozku vozítka. Nejprve jsou přidělané pomocí železné tyčky vedoucí skrz technické otvory obou motorů a zakončené pomocí zahnutí podél hrany podvozku na druhou stranu. Dále jsou skrz mechanické otvory v motoru a skrz otvor v podvozku provlečeny vodiče a tam svázané. Tento stejný princip je použit na středu motoru. Na hřídele motorů jsou následně nasazena plastová kola. Na druhé straně podvozku je přilepené pouzdro na baterie a na pouzdru je usazen mikrokontroler spolu s budičem. Z druhé strany je pak ještě přiděláno pomocné kolo, které zabraňuje, aby byl podvozek z jedné strany tažen po zemi. Celá sestava je vyfocena na *obrázku 1.14*.

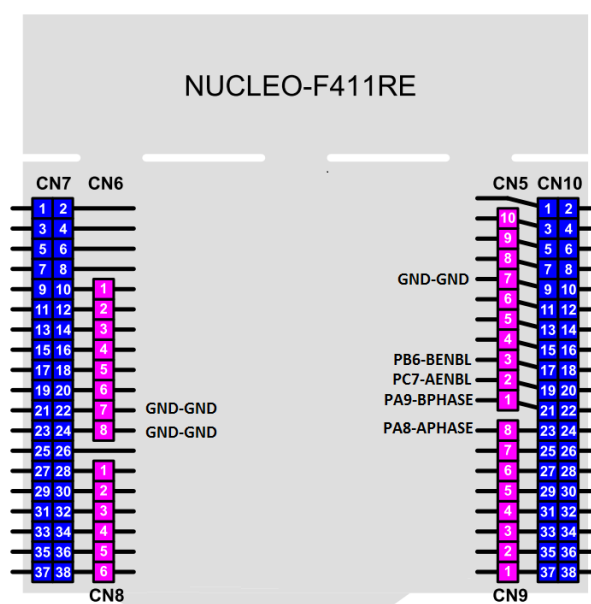
Elektrické propojení je realizováno pomocí vodičů. Z každého motoru vede dvoulinka, která je připojena do příslušných kontaktů na svorkovnici budiče. Dále je do stejné svorkovnice ještě připojeno napájení. To je realizováno již zmíněným pouzdrům s bateriemi. Případně je možné připojit například síťový adaptér. Budič samotný je připojen pomocí Arduino kontaktů na desku s mikrokontrolerem, kterým je napájen i plně řízen. Samotný mikrokontroler je napájen a programován přes USB kabel.



Obrázek 1.14 - sestava 1

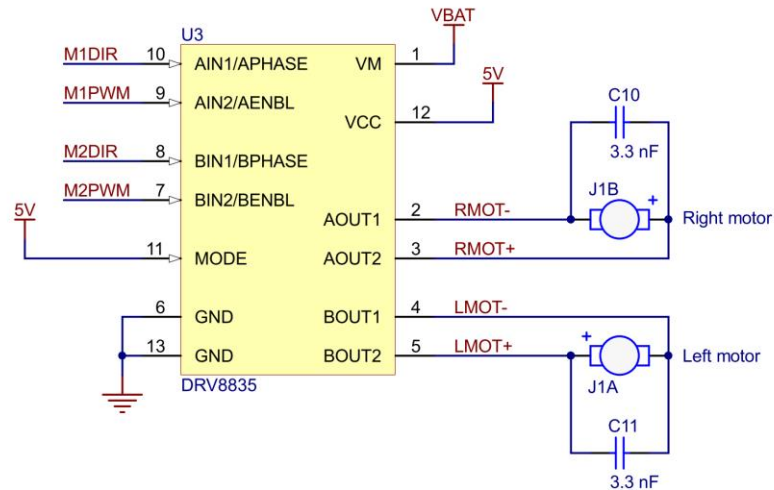
1.4.2 Sestava 2

Druhá sestava je Zumo Robot od firmy Pololu. Jedná se o pásové vozítko s přední radlicí. Veškeré informace o robotu jsou ze zdrojů [12][13]. Sestava obsahuje čtveřici kol, pro každou stranu vozítko dvě. Obrázek sestavy včetně připojení desky STM32 Nucleo je na *obrázku 1.17*. Přední kolo je vždy poháněno a zadním kolem je otáčeno pomocí pásu, který je nasazen přes obě kola. V přední části je zabudovaná radlice. Dále robot disponuje modulem s šesticí infračervených LED/fototranzistorových párů pro detekci čáry nebo hrany. Je zde integrovaný i akcelerometr a magnetometr, každý se třemi osami. Akcelerometr lze využít k detekci nárazů. Kompas není vzhledem k rušení z motorů, baterií a dalších komponent moc přesný, a proto se i při dobré kalibraci nehodí k navigaci. Lze ho použít jen pro přibližnou orientaci v prostředí. Ve spodku zařízení se nachází kryt pro čtyři 1,5 voltové AA baterie, které dodávají vozítku potřebnou energii. Na horní straně pouzdra je potom namontována řídicí deska, na které je připojen budič, obslužná tlačítka, identifikační diody, kompas a další obvody. Deska obsahuje také piny pro připojení řídicího mikrokontroleru, v našem případě opět deska STM32 Nucleo a další volné pájecí plošky pro případné doplnění dalších periférií. Robot je osazen budičem 8835, který je popsán v kapitole 1.2.2. Jeho propojení s deskou STM32 Nucleo, která budič řídí, je na *obrázku 1.15*.

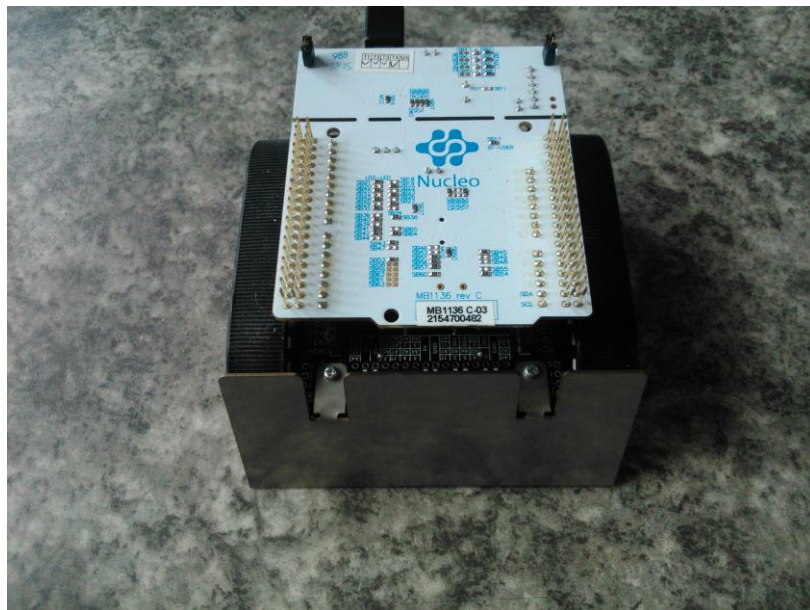


Obrázek 1.15 - propojení desky STM32 Nucleo s budičem

Budič je na řídicí desku připojen už od výrobce Zumo robota tak, aby fungoval v modu 1. To znamená, že pin mode je připojen na logickou jedničku. K budiči jsou již připojeny i motory. Na uživateli zůstává už jen obsluha pinů pro řízení. V modu 1 se jedná o signály xPHASE a xENBL. Připojení budiče a motorů k budiči je na *obrázku 1.16*. Dvojice použitých motorů obstarávající ve vozítku pohon je popsána v kapitole 1.1.2. Jedná se o motory Micro Metal Gearmotor HP 6V.



Obrázek 1.16 - propojení řídicí desky s budičem a budiče s motory.



Obrázek 1.17-Zumo Robot

1.5 Vývojová deska STM32 Nucleo

V práci je použita vývojová deska STM32 Nucleo s procesorem STM32F411RE od firmy STMicroelektronik, pomocí které jsou řízeny všechny použité měniče a motory. V této kapitole budou popsány, pro velké množství periférií, pouze periférie a příslušné registry v práci použité.

1.5.1 GPIO(General-purpose I/Os)

Tato periférie se stará o vstupy a výstupy. Tím lze pomocí mikrokontroleru řídit výstupní obvody jako například v této práci budič, který dále řídí motory, ale například i různé LED diody a jiné obvody.

Vstupní brána slouží pro získání dat z vnějších senzorů atd. STM32 Nucleo obsahuje 6 bran GPIOx, kde x je A, B, C, D, E nebo H. Brána se ovládá pomocí následujících registrů.

Dvoubitový registr MODER řídí režim brány. Ten může být podle binární kombinace ve stavu input (počáteční stav), output, alternativní režim nebo analogový vstup. OTYPER je jednobitový registr, který nastavuje výstup do push-pull nebo open-drain režimu. OSPEEDR nastavuje rychlost hrany na rychlosti Low, Medium, Fast a High speed. ODR je jednobitový registr, který přímo nastavuje hodnotu výstupu do stavu 1 nebo 0. IDR registr slouží ke čtení stavu vstupních obvodů. PUPDR slouží k nastavení Pull-up nebo Pull-down výstupu. BSRR má stejný účel jako ODR, ale je vnitřně úspornější. AFRL a AFRH je pro svoji velikost rozdělen do dvou registrů (low a high). Slouží k nastavování alternativní funkce portu. Například pro používání kanálů PWM na výstupu nebo Usart komunikaci a další alternativní režimy.

1.5.2 RCC (Reset and clock control)

Jedná se o klíčovou periférii rozvádějící hodiny do všech částí mikrokontroleru. Každý port musí mít přívod hodin. Periférie je ovládána registry. Každý registr je pojmenován podle jména sběrnice, například AHB1, APB2 a další. Použití registrů vysvětlím na sběrnici AHB1, na které jsou připojeny všechny GPIO porty. AHB1RSTR dělá resetovací pulz do periférie a AHB1ENR povoluje hodiny do příslušné periférie.

1.5.3 TIM (General-purpose timers)

V této podkapitole budu řešit skupinu TIM3-TIM5 pro generování PWM. Tato skupina je připojena na sběrnici APB1. Periférie slouží k čítání a časování pro mnoho aplikací. Například pro generování pulzně šířkové modulace. Každý Timer má k dispozici 4 kanály, kde každý může samostatně řídit vlastní výstup.

CR1 obsahuje bit *CEN* který spouští čítání a *ARPE* starající se o změnu ARR, který obsahuje periodu, až po přetečení. Pokud by se hodnota ARR změnila při compare, tak by mohlo dojít k chybě v taktu. CCER má bit *CCxE*, kde x je číslo kanálu. Tímto bitem se povoluje výstup. CCMR1(CCMR2), kde CCMR1 je pro kanály 1 a 2 a CCMR3 je pro kanály 3 a 4. Registr obsahuje trojici bitů. Bity nastavují pro PWM režim PWM1 nebo PWM2. V režimu PWM1 je aktivní úroveň pro CCRx <=CNT. Pro PWM2 inverzně. *CCRx*, kde x je zvolený kanál, obsahuje komparační hodnotu příslušného kanálu. ARR obsahuje bity s navolenou hodnotou, do které čítač komparuje.

1.5.4 USART (Universal synchronous asynchronous receiver transmitter)

Usart je využíván pro sériovou komunikaci. V této práci je ale na straně počítače sériová komunikace pomocí ovladače převedená z virtuálního COM portu na USB. Na straně druhé převádí ST-Link USB data na sériovou komunikaci. To znamená, že ve skutečnosti jsou data posílána pomocí USB paketů, protože STM32 Nucleo není přímo propojeno přes RS232 s PC. V této práci je Usartu použito pro posílání řídicích signálů z aplikace v Pc do Nuclea. A opačným směrem například hodnoty proudu z motorů zpět do aplikace. Celé řízení probíhá pomocí následujících registrů.

SR obsahuje bit *TXE*, který oznamuje prázdný vysílací registr. To znamená, že data byla přesunuta z registru TDR do vysílacího shift registru. Nulování probíhá automaticky po zapsání dalších dat. Bit *TC* oznamuje dokončení vysílání. *RXNE* oznamuje nově přijatá data v registru RDR. Řídicí registr CR1 obsahuje bit *RE*, který povoluje příjem, bit *TE*, který povoluje vysílání a *UE*, který povoluje použití periférie. Pomocí BRR registru se nastavuje vysílací rychlost. Registr DR pracuje z daty. Při čtení jsou data vyčítána z RDR registru a při vysílání jsou data zapsána do TDR registru.

1.5.5 ADC (Analog-to-digital convertor)

Tato periférie je určená k převodu analogových dat na data digitální pro jejich další zpracování nebo jen jejich zobrazení. V této práci je ADC použit pro převod dat obsahující velikost proudu, který budič dodává do motoru. Pro ovládání ADC jsou určeny následující registry.

CR1 obsahuje dva bity *RES*, které nastavují rozlišení převodníku. Lze nastavit 6, 8, 10, 12 bitový převod. Dalším bitem je bit *SCAN*, který povoluje práci s SQR kanály. CR2 registr obsahuje bit *ALIGH* určený pro nastavení, jestli budou bity v datovém registru zarovnány doleva nebo doprava, bit *ADON*, který zapíná AD převodník a bit *SWSTART*, který umožňuje programové spuštění převodu. To znamená, že spuštění si uživatel dělá ručně v programu. V SMPR1 a SMPR2 registru má každý kanál 3 bity, které řeší impedanční přizpůsobení pomocí doby převodu, která se dá předem spočítat. Registr SQR1 obsahuje 4 bity *L*, které říkají, kolik konverzí se provede v jedné sekvenci. Ve zbytku registru SQR1 a v registrech SQR2 a SQR3 je pak vždy 5 bitů *SQxx* pro každý použitý kanál. V těchto pěti bitech je zapsáno z jakých kanálů se v dané sekvenci bude převádět. Statut registr SR obsahuje bit EOC, který oznamuje dokončení převodu. V posledním registru DR je patnáct bitů *DATA*, které obsahují převedená data.

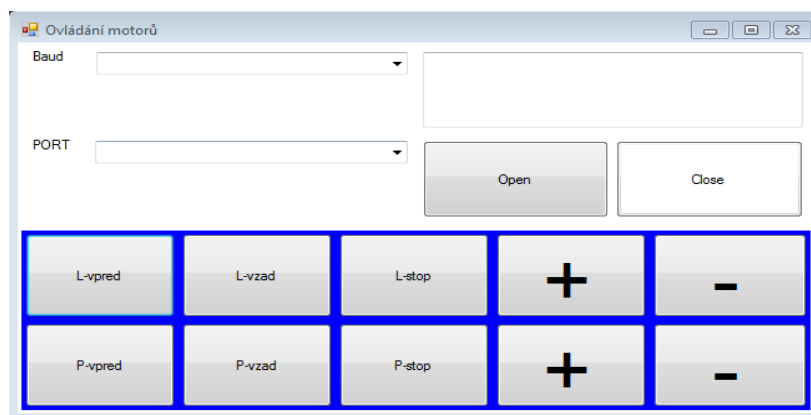
Tyto popsané bity v registrech se používají, pokud chceme převádět jen z jednoho kanálu. Pokud bychom chtěli provádět více převodů naráz, tak se v místo SQRx registrů použije registr JSQR, který je obdobou SQR registrů. Ale místo dvou bitů *L* obsahuje dva bity *JL*, které plní stejnou funkci a čtyři x čtyři bity *JSQx*, kde *x* je číslo kanálu od 1 do 4. Samotný start je pak proveden pomocí bitu *JSWSTAR*, který se nachází v registru CR2 a převedená data jsou ukládána pro každý kanál do vlastního registru JDRx. Zde je potřeba si dát pozor, který datový registr patří kterému kanálu. Protože neplatí, že například pro kanál 4 je vyhrazen registr JDR4.

2 Softwarové řízení motorů

V bakalářské práci jsou použity čtyři odlišné budiče a pro každý z nich je realizován vlastní řídicí program. Samotné ovládání je pak realizováno pomocí univerzální aplikace pro tyto účely naprogramované. Ta bude popsána v následující kapitole a další kapitoly se budou věnovat programům pro jednotlivé budiče. Jednotlivé programy v budičích obsahují několik knihoven. V kapitole 1.5 jsou popsány knihovny, které přímo realizují funkci budiče. Při části programování budičů byly využity nebo upraveny části knihoven z prezentací a výukových videí pro předmět KAE/MPP.

2.1 Aplikace pro ovládání motorů

Všechny použité budiče jsou ovládány pomocí sériové komunikace USART. Aby nemuselo být ovládání realizováno nepohodlně, například pomocí terminálu Hercules, tak jsem naprogramoval klikací aplikaci pro systém Windows v C# .NET. Hlavní okno aplikace je na *obrázku 2.1*. Je složeno z tlačítek pro samotné řízení budičů respektive motorů, tlačítek pro otevření a zavření sériové komunikace a dvou rozklikávacích polí se seznamem pro nastavení komunikační rychlosti a kanálu, po kterém bude komunikace probíhat. Poslední komponent je okno, které zobrazuje data z jednotlivých motorů, pokud je tato možnost pro daný budič dostupná a naprogramovaná. Každý motor má svojí sadu tlačítek. Tlačítko „L-vpřed“ zajišťuje otáčení levého motoru po směru chodu hodinových ručiček. „L-vzad“ v protisměru chodu hodinových ručiček. Tlačítko „L-stop“ zastavuje otáčení levého motoru. Možnost „+“ po zmáčknutí přidá rychlost příslušného motoru a „-“ naopak rychlost snižuje. Celek je nastaven tak, aby bylo dosaženo neskokového nárůstu rychlosti přibližně po 20 krocích. Tlačítka s označením „P“ jsou analogií pro pravý motor. Tlačítka „Close“ a „Open“ slouží k otevření nebo zavření celé sériové komunikace. Rozklikávací pole „baud“ umožňuje výběr komunikační rychlosti a „PORT“ po rozkliknutí nabídne aktuálně dostupné sériové porty pro komunikaci. Pokud není k Pc připojeno více zařízení, tak tato možnost umožní rychlé odhalení správného komunikačního portu. V pravém horním rohu je pak okno, kde se zobrazují informace, které budič umožňuje pomocí AD převodníku získat. Jedná se například o velikost proudu, který budič aktuálně dodává do motorů nebo signalizace nadměrné teploty, proudu a jiné nebezpečné stavy. Celý kód je v příloze na CD „1.Ovládací_program“.



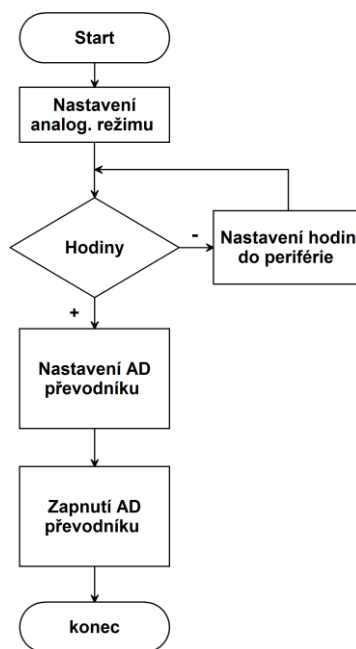
Obrázek 2.1 - hlavní obrazovka obslužného programu

2.2 Obslužné funkce pro řízení budičů

Jednotlivé programy v budičích obsahují podle možností budiče až 5 knihoven. Jedná se o knihovny, které obstarávají funkci AD převodníku, sériové komunikace, nastavení portů a hlavně knihovna „obsluha_PWM“, která obsahuje nastavení základního řízení pro konkrétní budič.

2.2.1 Knihovna pro AD převodník

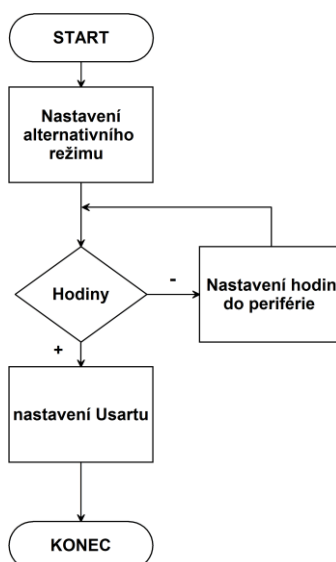
Tato knihovna realizuje AD převodník, který umožňuje převádět analogová data v budiči na data číslicová, která lze dále zpracovávat. Vývojový diagram je na *obrázku 2.2*. Nejprve se na správné piny nastaví režim analogový vstup. Následně se testuje, jestli jsou přivedeny hodiny do periférie AD převodníku. Pokud tomu tak není, tak se hodiny nastaví. Pokud jsou hodiny nastaveny, tak se musí provést nastavení příslušných registrů. Jednotlivé registry jsou popsány v kapitole 1.5.5. Poté je povoleno zapnutí samotného AD převodníku.



Obrázek 2.2 - vývojový diagram AD převodníku

2.2.2 Knihovna pro sériovou komunikaci

Knihovna „Nucleo_Usart“ se stará o sériovou komunikaci mezi budičem, ovládací aplikací a přenáší data do budiče a z budiče. Obsahuje několik funkcí pro příjem dat, odesílání dat, kontrolu, jestli jsou přijmutá nějaká data, možnost stringového posílání dat a inicializaci. Vývojový diagram inicializace Usartu je na *obrázku 2.3*. Samotné funkce Usartu budou popsány v kapitole o softwarové realizaci jednotlivých budičů.



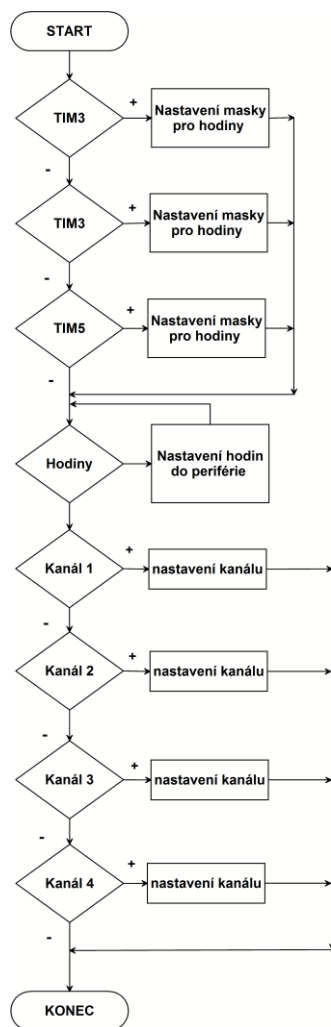
Obrázek 2.3 - inicializace sériové komunikace

2.2.3 Hlavní knihovna pro řízení PWM

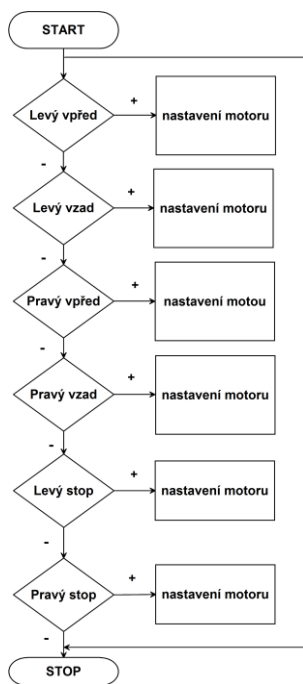
Tato knihovna obsahuje dvě funkce. Funkci „nastav_motor“ a „motor_smer“.

První jmenovaná funkce je funkcí obslužnou. Povoluje hodiny do periférie a pro každý motor nastavuje vlastní kanál. Funkce očekává čtyři parametry, pomocí kterých se dozví, jaký kanál se má nastavit, na jaké sběrnici se nacházejí hodiny, které musí být spuštěny a hlavně hodnotu ARR a CCRx registrů, pomocí kterých se řídí výsledná frekvence PWM do motoru a tím pádem i rychlost samotného motoru. Tato funkce je univerzální a lze s ní pohodlně pomocí čtyř parametrů rychle nastavit všechny používané kanály. Vývojový diagram funkce je na *obrázku 2.4*.

Druhá funkce je vlastně převedená pravdivostní tabulka z datasheetu konkrétního budiče, realizovaná pomocí switche a uvádí motor do příslušného režimu. Režimy jsou pojmenované „Levy_vpřed“, „Levy_vzad“, „pravy_vpřed“, „pravy_vzad“, „pravy_stop“, „Levy_stop“. Z názvů je jasné, která větev switche co dělá. Jednotlivé režimy jsou nastavovány pomocí kombinace signálů na příslušné piny. Příslušný stav, do kterého si přejeme motor uvést je předáván funkci pomocí parametru. Funkce musí být pokaždé upravená, protože každý budič vyžaduje jiný způsob řízení. Vývojový diagram této funkce je na *obrázku 2.5*. Jedná se však o obecný diagram. V kapitole, kde jsou porovnávány jednotlivé budiče, je vývojový diagram této funkce rozpracován vždy pro konkrétní budič.



Obrázek 2.4 –vývojový diagram funkce nastav_motor



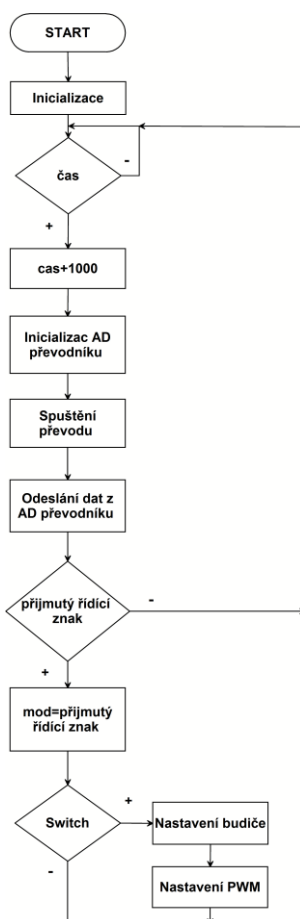
Obrázek 2.5 - vývojový diagram funkce motor_smer

2.2.4 Obslužná knihovna

Obslužná knihovna „stm_core“ obsahuje tři funkce, které se starají o nastavení hodin do příslušného GPIO portu, konfiguraci do příslušného režimu (vstup, výstup, a jiné), čtení ze stavu v příslušném pinu a zápis do něj. Funkce je univerzální pro všechny dostupné porty. Její funkce jsou použity v každé předešlé knihovně.

2.3 Softwarové řízení pro shield s budičem VNH5019

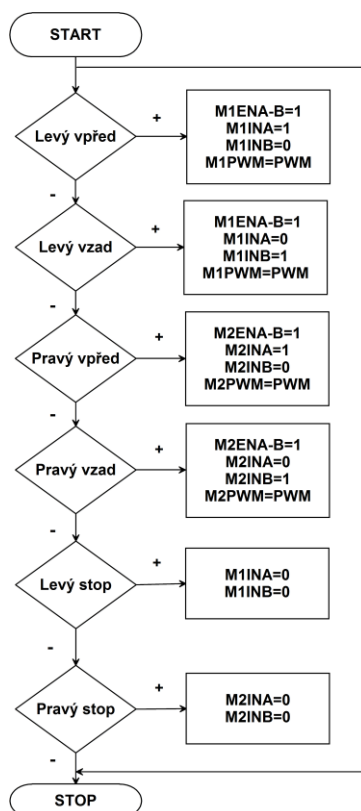
Celý kód je k dispozici v příloze „2.budič_VNH5019“. Main se skládá z pěti částí. Inicializace, časovače, Usartu, AD převodníku a switche, který určuje, jakou funkci bude motor dělat. Hlavní struktura funkce main je na *obrázku 2.6*



Obrázek 2.6 - základní struktura programu pro budič VNH5019 a L6206

Každou vteřinu přeteče časovač a spustí AD převodník, který vyčte z analogových pinů PA0 a PA1 na budiči informaci o velikosti proudu a pomocí Usartu ji odešle do ovládací aplikace, kde si uživatel může hodnotu přechíst. Následně se testuje, jestli nebyl pomocí Usartu z ovládací aplikace vyslán znak pro obsluhu

budiče. Pokud se tak stane, tak se znak ze vstupního registru uloží do proměnné „mod“ a pomocí switche se pak zapne příslušný režim budiče pomocí funkce „motor_smer“, která je na *obrázku 2.7*.

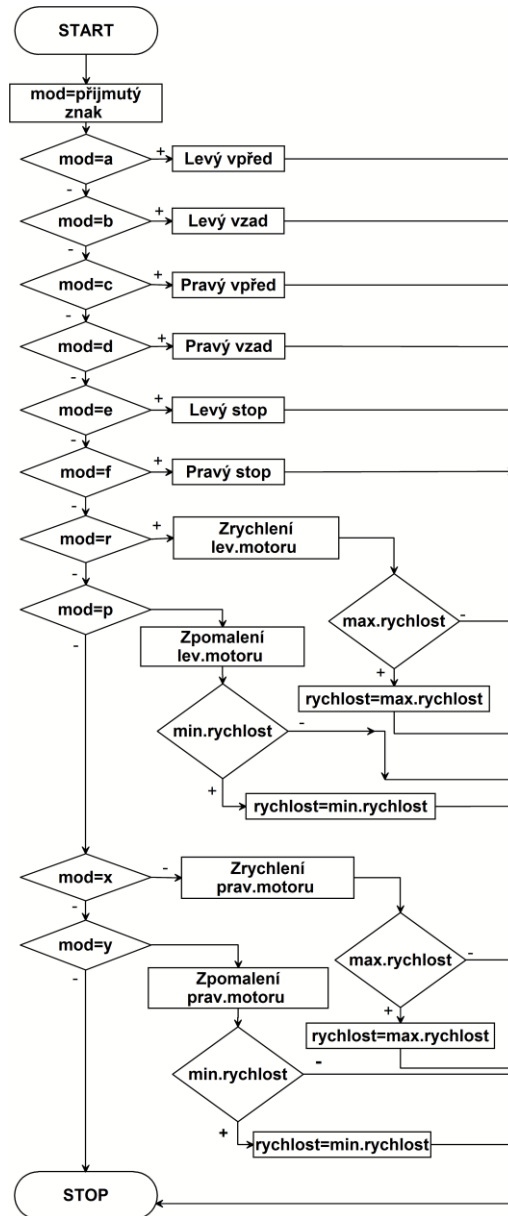


Obrázek 2.7 - funkce motor_smer pro budič VNH5019

Samotný switch je potom na vývojovém diagramu na *obrázku 2.8*. Příslušný case switche si volá funkci „motor_smer“ a ta nastaví příslušný režim. Zrychlování a zpomalování je potom realizováno přímo v příslušné větvi. Pokud je požadavek na zrychlení nebo naopak na zpomalení motorů, tak se přičte nebo odečte vhodně zvolená konstanta (v tomto případě 40) do pomocné proměnné „CCR“. Pokud je dosaženo maximální nebo minimální rychlosti, pak při požadavku o další zrychlování nebo zpomalování zůstane nastavená původní minimální nebo maximální hodnota. Tento switch je použit ve všech dalších budičích a liší se pouze v konstantě zapisované do CCR registru. V závěru funkce main se volá funkce „nastav_motor“, které se předá informace o zvoleném kanálu, časovači, nastavené rychlosti z pomocné proměnné, která se ve funkci přepíše do registru CCRx a zvolená konstanta do registru ARR. Rychlost motoru, respektive frekvenci PWM lze spočítat pomocí vzorce 2.1 a 2.2.

$$ARR = \frac{\text{maximální frekvence mikroprocesoru}}{\text{maximální frekvence budiču}} [-] \quad (2.1)$$

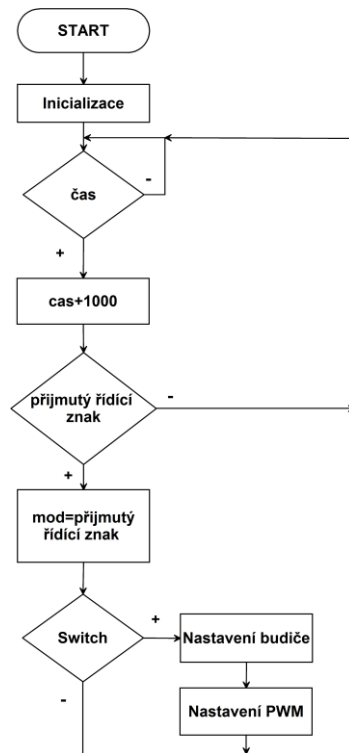
$$CCR_x = \frac{ARR \times (\text{číslo v rozmezí } 0 - 100)}{100} [\%] \quad (2.2)$$



Obrázek 2.8 - switch funkce main pro budič VNH5019

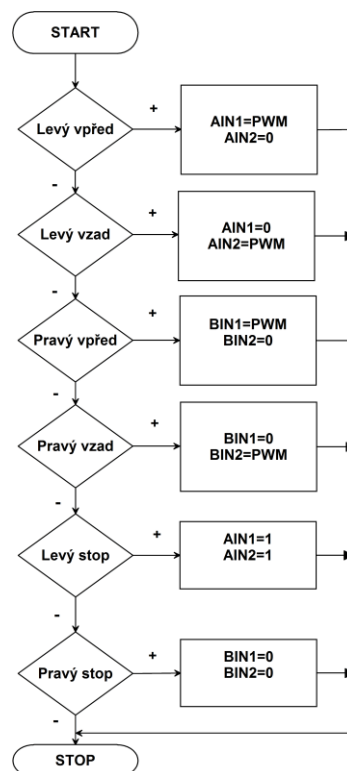
2.4 Softwarové řízení pro shield s budičem DVR-8833

Celý kód pro tento budič je v příloze „3.budič_DVR-8833“. Kód je složen ze tří základních bloků. Vývojový diagram je na *obrázku 2.9*.



Obrázek 2.9 - základní kostra programu pro budič DVR- 8833 a DVR-8835

V nekonečné while smyčce je nejprve testováno, jestli nedošlo k přijmutí znaku z ovládací aplikace. Pokud tak nastane, tak se pomocí switche vybere příslušná větev a spustí se příslušný režim z funkce „motor_smer“. Tato funkce je zobrazena *na obrázku 2.10*.

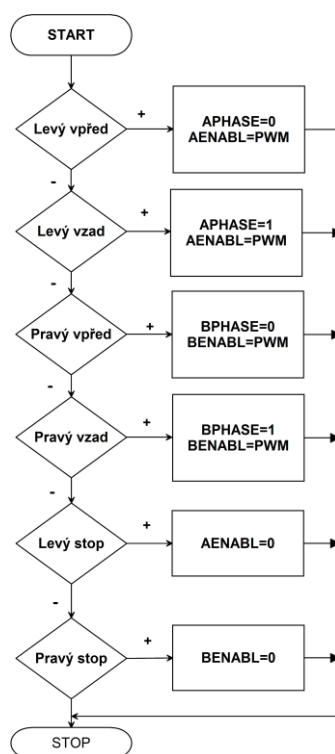


Obrázek 2.10 - funkce motor_smer pro budič DVR-8833

Oproti předchozímu budiči je zde v části pro změnu rychlosti nastavována jiná vhodná konstanta, aby opět bylo dosaženo přibližně 20 stupňů rychlosti. Výpočet hodnot jednotlivých konstant je v rovnicích 2.1 a 2.2. Konstanta je odlišná z důvodu jiné maximální frekvence, kterou budič dokáže na PWM výstup dodat. Po spuštění příslušného režimu motoru se na závěr volá funkce „nastav_motor“, pomocí které se nastaví opět potřebné registry pro řízení PWM.

2.5 Softwarové řízení pro robota ZUMO s budičem DVR-8835

Kompletní kód programu je v příloze „4.budič_DVR-8835“. Hlavní kostra programu je znázorněna vývojovým diagramem na *obrázku 2.9*. Nejprve je testováno, jestli nepřišel řídicí znak po sériové lince. Pokud k této události dojde, tak se znak přepíše do pomocné proměnné a dále se pomocí něho větví program do větví switche. Každá větev následně zavolá obslužnou funkci „motor_smer“. Funkce je na *obrázku 2.11*. Jako v předchozím budiči je zde opět pro změnu rychlosti upravená konstanta, která se pomocí funkce „nastav_motor“ předává do CCRx registru tak, aby při maximální využití frekvence budiče bylo možné využít přibližně 20 rychlostí motorů. V závěru se vyvolá funkce „nastav_motor“, která vykoná nastavení všech potřebných registrů.

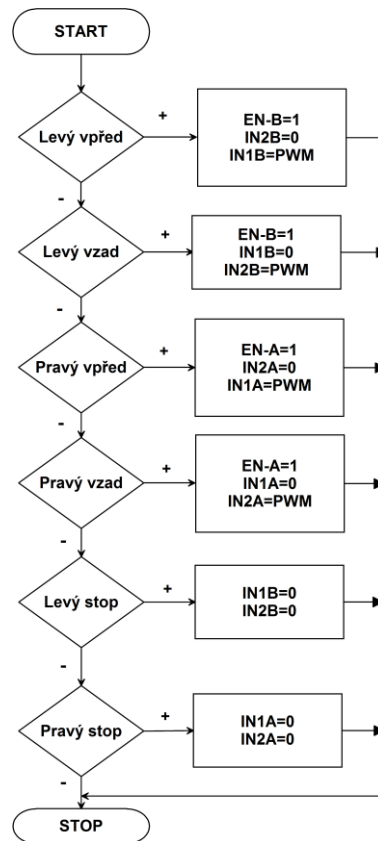


Obrázek 2.11 - funkce motor_smer pro budič DVR-8835

2.6 Softwarové řízení pro shield X-NUCLEO-IHM04A1 s budičem L6206

Kompletní kód programu budiče je v příloze „5.budič_L6206“. Hlavní struktura programu je vidět na vývojovém diagramu na *obrázku 2.6*. Pokud dojde k uplynutí doby jedné sekundy, dojde k vyčtení

analogových pinů budiče a získané informace se pomocí sériové linky pošlou do ovládací aplikace, kde se uživateli zobrazí na display. Následně probíhá testování na přijetí znaku z ovládací aplikace. Pokud k přijetí znaku dojde, tak se přijatý znak uloží do pomocné proměnné „mod“ a pomocí switch se vybere příslušná větev, která spustí pomocí funkce „motor_smer“ příslušný režim budiče. Switch je zobrazen na **obrázku 2.12**. Opět je zde upravena konstanta pro řízení rychlosti a v závěru jsou všechny potřebné hodnoty předány funkci „nastav_motor“, která nastaví kanál, bity ARR a CCRx a další.



Obrázek 2.12 - funkce motor_smer pro budič L6206

3 Výhled dalšího využití

V této kapitole budu řešit další možné rozšíření, které by šlo použít v této bakalářské práci. Konkrétně jsem se rozhodl pro možnost měření otáček. Popíši zde několik způsobů jako je například měření otáček založených na různých principech. Změřené otáčky by se daly pomocí převodu přepočítat na rychlost anebo použít pro řízení rychlosti pomocí PID regulátoru.

3.1 Měření otáček pomocí magnetických senzorů

Metoda zde popsána bude založena na principu Hallova jevu. V této bakalářské práci je velmi vhodná pro svoji možnost být namontovaná přímo na hřídeli motoru. Takovéto motory vyrábí například firma Pololu jako motory s prodlouženou hřídelí nebo také motory s dvojitou hřídelí. Hřídel je prodloužená o vhodnou délku. Na prodlouženou část je možné nasadit kompatibilní magnetický senzor (Hallova sonda), který firma také dodává. Se senzoru pak vedou vodiče, které lze připojit k řídicímu mikrokontroleru a získávat data, která se následně dále zpracují.

3.1.1 Princip Hallova jevu

Pokud umístíme vodivou destičku z kovu nebo polovodivého materiálu do homogenního magnetického pole (vektory magnetického pole jsou navzájem rovnoběžné) tak, aby vektor magnetického byl kolmý na destičku, tak začne destičkou procházet elektrický proud. Pokud připojíme voltmetr na boční strany destičky, tak naměříme takzvané Hallovo napětí. Toto napětí vzniká vlivem magnetické síly, která působí na volné nosiče náboje tvořící proud a vychyluje je z původního směru. Tím zajistí nadbytek nosičů na jedné straně a nedostatek na straně druhé. Tím vzniká elektrické pole o určité intenzitě. Velikost Hallova napětí je dána rovnováhou mezi magnetickou silou působící na elektrony a elektrickou silou. [8].

Tohoto principu využívá Hallova sonda. Ta je přidělena třeba na již zmíněné hřídeli a vyhodnocuje průchod kovového materiálu. Při každém průchodu například vyšle impuls do řídicí jednotky, kde se impulzy inkrementují a tím získáváme údaje o měřených otáčkách.

3.2 Měření otáček pomocí stroboskopického jevu

Tohoto bezkontaktního měření využívají s výhodou modeláři pro měření rychlosti otáček vrtulí. Princip je založen na detekci rotující reflexní značky, která je umístěná například na vrtuli.

3.2.1 Princip měření otáček pomocí stroboskopického jevu

Stroboskop je zařízení, které vysílá vysoko intenzivní impulzy světla na reflexní značku nebo i pruhu barvy, která je umístěná na měřeném otáčejícím se objektu. Frekvence, kterou stroboskop osvětluje reflexní prvek, je známá. Reflexní prvek odráží vyslané světlo a to se lidskému oku jeví jako pohybující se bod. Pokud je frekvence stroboskopu větší než frekvence měřené části, tak se zdá, že se bod točí v protisměru hodinových ručiček. Pokud je frekvence stroboskopu menší než frekvence měřené části, tak se naopak zdá,

že osvětlovaný prvek se pohybuje ve směru hodinových ručiček. Cílem měření je pohybující se bod zastavit. To znamená, že se snažíme nastavit frekvenci na stroboskopu tak, aby se osvětlovaný prvek již nepohyboval. Tím jsme dosáhli stavu, kdy se měřená frekvence a známá frekvence stroboskopu rovnají. Ze zjištěné frekvence lze pak dopočítat otáčky.

3.3 Měření otáček pomocí optické závory

Jedna z metod optické závory se hodí pro kola s více otvory, a proto je tato metoda pro tuto práci vhodná. Například v kapitole 1.4.1 se píše o sestavě 1 a ta má na hřídeli motoru připojené právě takové kolo. Optická závora se nastaví, aby snímala otvory v kole a pomocí přepočtu se získá počet otáček a následně rychlost.

3.3.1 Princip optické závory

Optická závora může fungovat na různých principech. Jednou z možností je přijímač a vysílač v jednom pouzdře. Vysílač vyšle paprsek světla a ten se odrazí od materiálu s dobrou odrazivostí a přijímač ho detekuje. Pokud se do dráhy paprsku dostane cizí objekt, tak je zamezeno odrazu paprsku. Přijímač v tomto případě nedetekuje odražený paprsek a aktivuje výstup. V případě již zmíněného otáčejícího se kola s více otvory odrazení paprsku zabraňují jednotlivé části, které oddělují otvory v kole.

Další možností může být jednocestná závora. V tomto případě jsou vysílač a přijímač samostatné součástky a jsou umístěné naproti sobě. Vysílač vyšle světelný paprsek a ten je přijmut v přijímači. Pokud se do cesty paprsku dostane cizí objekt, tak je paprsek přerušen a nedojde k přijmutí světelného paprsku a přijímač aktivuje výstup.

Závěr

Cílem bakalářské práce bylo porovnat dostupné budiče a malé pohyblivé roboty dostupné na katedře a sestavit pro ně vzorové aplikace. Na začátku práce je popsán základní princip stejnosměrných motorů, můstkového budiče a dalších možností řízení motorů kromě použití můstkového budiče a PWM. Následně jsou porovnány použité motory. Poté se již bakalářská práce věnuje hlavnímu cíli zadání.

Výsledkem práce jsou dvě robotická vozítka a dva samostatné budiče, ke kterým byly připojeny motory, aby byla otestována funkčnost aplikace. Jednotlivé sestavy jsou vždy řízeny pomocí vývojové desky STM32 Nucleo s procesorem STM32F411RE. Pro pohodlné řízení motorů jsem napsal ještě aplikaci pro systém Windows, pomocí které lze jednotlivé motory řídit. Ve výsledku je vždy postavená sestava skládající se z vývojové desky, která obsahuje Arduino piny, do kterých je připojen budič. K samotnému budiči jsou pak následně připojeny motory. Jednotlivé budiče se od sebe liší maximální frekvencí, kterou dokáží dodat na výstup PWM, napájecím napětím a maximálním proudem schopným dodat do motorů. Některé budiče umožňují vedle řízení motorů i měření dodaného proudu nebo signalizaci chybových stavů. Jediná nevýhoda je propojení vývojové desky pomocí USB kabelu s PC. A tím pádem nemožnost jízdy vozítka dál než je dosah USB kabelu. Případné vylepšení by mohlo být v bezdrátovém propojení, například pomocí BlueTooth.

Prvním robotickým vozítkem je sestava podrobně popsána v kapitole 1.4.1. Je zde použit budič VNH5019 a dva motory POLOLU-1124. Napájení budiče je realizováno pomocí čtyř tužkových AA 1,5 V baterií. Budič je řízen programem, který je na CD v příloze „2.budič_VNH5019“. Samotné ovládání je potom realizováno pomocí aplikace, která je na CD v příloze „1.Ovládací_program“. Tou je možno vozítku měnit směr jízdy dopředu, dozadu a měnit rychlost. Motory nejsou ale dostatečné kvality a proto i když je nastavena pro oba motory stejná rychlost, tak se vozítko stáčí jedním směrem a nejede rovně. Budič má i možnost zjištění velikosti proudu, který je dodáván budičem do příslušného motoru. To je realizováno pomocí AD převodníku, který vyčte data z příslušného pinu budiče, zpracuje a zobrazí v aplikaci.

Druhým robotickým vozítkem je sestava podrobně popsána v kapitole 1.4.2. Jedná se o robota ZUMO od výrobce Pololu. V sestavě je použit budič DVR-8835. Budič je řízen programem „4.budič_DVR-8835“. Robot opět je ovládán pomocí univerzální aplikace, která je na CD v příloze „1.Ovládací_program“ a opět má možnost jízdy dopředu, dozadu a změnit rychlost jízdy. Je možné ho i rozšiřovat o další chytré moduly nebo použít moduly již připojené. Například je zde kompas a akcelerometr. Funkčnost těchto prvků ale není cílem této práce. U tohoto robota jsou připojeny motory Micro Metal Gearmotor HP 6V. Jedná se již o lepší motory než v předchozí sestavě a robot je již schopen přímé jízdy bez stáčení se k jedné straně.

Dalším použitým budičem tentokrát již bez robotického vozítka je budič L6206 na desce X-NUCLEO-IHM04A1. Celá deska s budičem je připojena na vývojovou desku STM32 Nucleo a k budiči jsou pak připojeny dva motory POLOLU-1118. Budič je řízen programem na CD v příloze „5.budič_L6206“. Samotné ovládání směru točení motorů a rychlosti je opět realizováno pomocí univerzální aplikace.

Posledním budičem je budič DVR-8833. Samotná deska budiče není o moc větší, než budič samotný. Z toho důvodu není ani možnost, oproti předchozím budičům, přímého připojení budiče k vývojové desce STM32 Nucleo. Proto bylo při práci s tímto budičem použito nepájivé kontaktní pole. Budič se připojil do pole a pomocí vodičů se propojí s Arduino konektory na desce STM32 Nucleo. Motory jsou pak také přes nepájivé pole připojeny k budiči. Program pro řízení budiče je na CD v příloze „3.budič_DVR-8833“.

Ovládání je realizováno pak opět jako v předchozích případech pomocí univerzální aplikace. Opět je umožněná jízda dopředu, dozadu a změna rychlosti.

V závěru bakalářské práce jsou ještě krátce popsány možnosti, kterými lze měřit otáčky motorů. Nejvhodnější z metod pro použité motory v této práci se ukázala být metoda měření otáček pomocí Hallovy sondy. Výrobce dodává takzvané motory s dvojitou nebo prodlouženou hřídelí a k tomu i kompatibilní sondu. Sonda se lehce přidělá na hřídel motoru a pomocí vodičů se připojí k desce STM32 Nucleo. Tímto způsobem lze snadno bez velkých zásahů do konstrukce získat informace o otáčkách motoru. Tyto údaje se následně dají použít pro výpočet rychlosti a řízení otáček motoru pomocí PID regulátoru.

Seznam literatury a informačních zdrojů

- [1] Kůs, Václav. Elektrické pohony a výkonová elektronika. Druhé vydání. Plzeň: Západočeská univerzita v Plzni, 2016. 181 stran. ISBN 978-80-261-0639-5.
- [2] Bartoš, Václav. Elektrické stroje. 2. vyd. Plzeň: Západočeská univerzita, 2004. 221 s. ISBN 80-7043-332-9.
- [3] MARTIN, Dundáček. Mikroprocesorový modul řízení SS motoru se zpětnou vazbou [online]. Brno, 2008 [cit. 2017-04-20]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=7982. Diplomová práce.
- [4] Pololu Robotics and Electronics [online]. Copyright © [cit. 20.04.2017]. Dostupné z: https://www.pololu.com/file/download/vnh5019.pdf?file_id=0J504
- [5] Pololu Robotics and Electronics [online]. Copyright © [cit. 20.04.2017]. Dostupné z: https://www.pololu.com/file/download/drv8833.pdf?file_id=0J534
- [6] Pololu - 120:1 Mini Plastic Gearmotor, 90° 3mm D-Shaft Output. Pololu Robotics and Electronics [online]. Copyright © 2001 [cit. 20.04.2017]. Dostupné z: <https://www.pololu.com/product/1124>
- [7] Pololu - 228:1 Plastic Gearmotor, Offset Output. Pololu Robotics and Electronics [online]. Copyright © 2001 [cit. 20.04.2017]. Dostupné z: <https://www.pololu.com/product/1118>
- [8] Hallův jev :: MEF. Fyzika :: MEF [online]. Copyright © 2006 [cit. 04.05.2017]. Dostupné z: <http://fyzika.jreichl.com/main.article/view/294-halluv-jev>
- [9] STMicroelectronics [online]. Copyright © [cit. 04.05.2017]. Dostupné z: <http://www.st.com/content/ccc/resource/technical/document/datasheet/6e/e7/d6/58/65/53/48/2a/CD0002346.pdf/files/CD00002346.pdf/jcr:content/translations/en.CD00002346.pdf>
- [10] STMicroelectronics [online]. Copyright ©y [cit. 04.05.2017]. Dostupné z: http://www.st.com/content/ccc/resource/technical/layouts_and_diagrams/schematic_pack/c2/c4/1f/5c/41/ba/4e/f3/x-nucleo-ihm04a1_schematic.pdf/files/x-nucleo-ihm04a1_schematic.pdf/jcr:content/translations/en.x-nucleo-ihm04a1_schematic.pdf
- [11] Pololu Robotics and Electronics [online]. Copyright ©p [cit. 04.05.2017]. Dostupné z: https://www.pololu.com/file/download/drv8835.pdf?file_id=0J570
- [12] Pololu - Zumo Robot for Arduino (Assembled with 75:1 HP Motors). Pololu Robotics and Electronics [online]. Copyright © 2001 [cit. 04.05.2017]. Dostupné z: <https://www.pololu.com/product/2506>
- [13] Pololu - Zumo Reflectance Sensor Array. Pololu Robotics and Electronics [online]. Copyright © 2001 [cit. 04.05.2017]. Dostupné z: <https://www.pololu.com/product/1419>
- [14] Pololu - 75:1 Micro Metal Gearmotor HP 6V. Pololu Robotics and Electronics [online]. Copyright © 2001 [cit. 04.05.2017]. Dostupné z: <https://www.pololu.com/product/2361>
- [15] BĚLÍŠEK, David. Řízení krokového motoru mikrokontrolérem. Brno, 2009. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Ing. Ladislav Káňa.
- [16] OTRUBA, Jiří. Implementace řízení BLDC motoru. V Plzni, 2015. Diplomová práce (Ing.). Západočeská univerzita v Plzni, Fakulta elektrotechnická. Vedoucí práce David Vošmik.

Přílohy

Elektrické přílohy na CD

Příloha 1- aplikace pro ovládání motorů

1.Ovládací_program

Příloha 2- program pro budič VNH-5019

2.budič_VNH5019

Příloha 3- program pro budič DVR-8833

3.budič_DVR-8833

Příloha 4- program pro budič DVR-8835

4.budič_DVR-8835

Příloha 5- program pro budič L6206

5.budič_L6206