

# HW vývojový systém pro robotiku

Bc. David Wunderlich

---

Diplomová práce  
2017



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. David Wunderlich**  
Osobní číslo: **A15171**  
Studijní program: **N3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**  
Forma studia: **prezenční**

Téma práce: **HW vývojový systém pro robotiku**  
Téma anglicky: **A HW System for Developing Robotic Applications**

## Zásady pro vypracování:

1. Analyzujte požadavky na architekturu centrální řídicí jednotky pro robotické aplikace.
2. Navrhněte schéma zapojení centrální jednotky a sensorových modulů.
3. Navrhněte, osadte a oživte plošné spoje jednotky a modulů.
4. Implementujte ukázkové příklady práce s jednotkou i moduly.
5. Vytvořte implementační a uživatelskou dokumentaci.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. PINKER, Jiří. Mikroprocesory a mikropočítače. Praha: BEN – technická literatura, 2004. ISBN 80-730-0110-1.
2. ZÁHLAVA, Vít. Návrh a konstrukce desek plošných spojů: principy a pravidla praktického návrhu. 1. vyd. Praha: BEN – technická literatura, 2010, 123 s. ISBN 978-80-7300-266-4.
3. PARK, Eun Jung. Lego Mindstorms EV3: stavíme a programujeme roboty. Brno: Computer Press, 2015. ISBN 978-80-251-4385-8.
4. CATSOULIS, John. Designing embedded hardware. 2nd ed. Sebastopol, CA: O'Reilly, c2005. ISBN 0596007558.
5. VÁŇA, Vladimír. ARM pro začátečníky. Praha: BEN – technická literatura, 2009. ISBN 978-80-7300-246-6.
6. ÚREDNÍČEK, Zdenek. Robotika. Zlín, 2012. ISBN 978-80-7454-223-7.
7. YIU, Joseph. The definitive guide to ARMCortex-M3 and Cortex-M4 processors. Third edition. ISBN 0124080820.
8. PEHLIVANIAN, Ara a Don NGUYEN. JavaScript okamžitě. Brno: Computer Press, 2014. ISBN 978-80-251-4163-2.

Vedoucí diplomové práce:

**Ing. Tomáš Dulík, Ph.D.**

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

**3. února 2017**

Termín odevzdání diplomové práce:

**16. května 2017**

Ve Zlíně dne 3. února 2017



doc. Mgr. Milan Adámek, Ph.D.

*děkan*



prof. Mgr. Roman Jasek, Ph.D.

*ředitel ústavu*

### **Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### **Prohlašuji,**

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 15.5.2017

  
.....  
podpis diplomanta

## **ABSTRAKT**

Tato práce se zabývá návrhem systému pro výuku programování a robotiky. Systém je využíván převážně pro studenty základních a středních škol, zájmových kroužků a letních táborů. V teoretické části jsou srovnány systémy, které lze využít pro výuku programování a robotiky. Následně je popsán software, který je používán pro výuku a vývoj na navrženém systému.

V praktické části je popsána konstrukce systému a modulů, které jsou používány pro výuku. Pro jednotlivé moduly jsou vytvořeny vzorové programy znázorňující funkci modulů a možnosti jejich ovládní.

Součástí práce je také uživatelská dokumentace popisující celý systém.

Klíčová slova: robotické systémy, Espruino, Lego, STMicroelectronics, hardware

## **ABSTRACT**

This thesis is focused on developing system for education of programming and robotics. System is mainly used for students of primary schools, programming and robotics clubs and summer camps. In theoretic section, there is a comparison of various robotic systems, which can be used for learning programming and robotics. Subsequently, software which is used for teaching and developing on designed system is described.

In practical part of thesis, construction of developed system and modules is described. For each developed module, an example program is created, explaining functions and possibilities of use.

Keywords: robotic systems, Espruino, Lego, STMicroelectronics, hardware

Tímto bych rád poděkoval vedoucímu mé diplomové práce panu Ing. Tomáši Dulíkovi Ph.D za jeho trpělivost, ochotu, rady a za veškerou pomoc při zpracování diplomové práce i za pomoc v průběhu celého studia.

Prohlašuji, že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

# OBSAH

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>11</b>
<b>1 ROBOTICKÉ SYSTÉMY</b> .....	<b>12</b>
1.1    LEGO® MINDSTORMS® .....	12
1.1.1    První generace Lego Mindstorms - RCX.....	13
1.1.2    Druhá generace Lego Mindstorms - NXT .....	14
1.1.3    Třetí generace Lego Mindstorms – EV3 .....	15
1.1.4    Vyhodnocení .....	16
1.2    ARDUINO.....	17
1.2.1    Typy desek .....	18
1.2.2    Arduino Nano.....	18
1.2.3    Arduino Uno.....	18
1.2.4    Arduino Robot.....	19
1.2.5    Vyhodnocení .....	20
1.3    ESPRUINO .....	21
1.3.1    Blockly .....	21
1.3.2    Espruino Board.....	22
1.3.3    STM Discovery .....	22
1.3.4    Vyhodnocení .....	23
1.4    MAKEBLOCK .....	23
1.4.1    Scratch.....	24
1.4.2    MakeblockOrion .....	24
1.4.3    Vyhodnocení .....	25
<b>2 VYBAVENÍ VÝVOJOVÉHO SYSTÉMU</b> .....	<b>26</b>
2.1    CENTRÁLNÍ JEDNOTKA .....	26
2.2    AKČNÍ ČLENY .....	26
2.2.1    Motory.....	26
2.2.1.1    Kartáčové stejnosměrné motory .....	27
2.2.1.2    Servomotory.....	27
2.2.1.3    Ovládání.....	28
2.3    SENZORY .....	29
2.3.1    Ultrazvukový sensor vzdálenosti .....	29
2.3.2    IR senzory .....	29
2.3.3    Potenciometry .....	29
2.3.4    Enkodéry .....	30
2.3.4.1    Konstrukce .....	30
2.3.4.2    Princip funkce .....	30
2.3.4.3    Vyhodnocení signálu .....	31
<b>II PRAKTICKÁ ČÁST</b> .....	<b>32</b>
<b>3 HARDWAROVÁ REALIZACE SYSTÉMU</b> .....	<b>33</b>
3.1    CENTRÁLNÍ JEDNOTKA .....	33
3.1.1    Blokové zapojení.....	33
3.1.2    Napájení .....	34
3.1.3    Mikropočítač .....	36

3.1.4	Led panel .....	37
3.1.5	Tlačítkový panel .....	37
3.1.6	USB .....	38
3.1.7	Senzorové konektory .....	39
3.1.8	Výstupní konektory .....	40
3.1.9	Servomotorové konektory .....	41
3.1.10	Wifi konektor .....	42
3.1.11	Displej konektor .....	42
3.1.12	Debugger .....	42
3.1.13	Deska plošných spojů .....	43
3.2	BATERIOVÝ MODUL .....	44
3.2.1	Schéma .....	45
3.2.2	Deska plošných spojů .....	45
3.3	MIKROSPÍNAČ .....	46
3.3.1	Schéma .....	46
3.3.2	Deska plošných spojů .....	47
3.4	POTENCIOMETR .....	47
3.4.1	Schéma .....	47
3.4.2	Deska plošných spojů .....	48
3.5	ULTRAZVUKOVÝ SENZOR .....	48
3.5.1	Schéma .....	48
3.5.2	Deska plošných spojů .....	49
3.6	BZUČÁK .....	49
3.6.1	Schéma .....	49
3.6.2	Deska plošných spojů .....	50
3.7	IR SENZOR .....	50
3.7.1	Schéma .....	51
3.7.2	Deska plošných spojů .....	51
3.8	DC MOTOR .....	52
3.8.1	Schéma .....	52
3.8.2	Deska plošných spojů .....	52
3.9	BATERIOVÝ MODUL S NABÍJENÍM .....	53
<b>4</b>	<b>PROGRAMOVÁNÍ CENTRÁLNÍ JEDNOTKY .....</b>	<b>54</b>
4.1	NAHRÁNÍ FIRMWARE .....	54
4.2	INSTALACE A POPIS IDE .....	54
4.3	PROGRAMOVÁNÍ .....	55
4.3.1	Výpis do konzole a deklarace proměnných .....	55
4.3.2	Aritmetické operace .....	56
4.3.3	Čtení digitálního vstupu .....	57
4.3.4	Zápis digitálního výstupu .....	57
4.3.5	Čtení analogové hodnoty .....	57
4.3.6	Zápis analogové hodnoty – pulzně šířková modulace .....	58
4.3.7	Ovládání tlačítek .....	58
4.3.8	Periodicky spouštěný kód .....	59
4.3.9	Mikrospínač .....	59
4.3.10	Senzor čáry a servomotory .....	60

4.3.11	Ultrazvukový senzor .....	62
4.3.12	Stejnoseměrný motor s enkodérem .....	63
<b>5</b>	<b>DOKUMENTACE.....</b>	<b>66</b>
5.1	SENZOROVÉ KONEKTORY .....	67
5.2	VÝSTUPNÍ – MOTOROVÉ KONEKTORY .....	67
5.3	WIFI KONEKTOR .....	68
5.4	DEBUGGER / ST-LINK .....	69
5.5	OVLÁDACÍ A SIGNALIZAČNÍ PANEL.....	70
5.6	SERVOMOTOROVÉ KONEKTORY.....	72
	<b>ZÁVĚR .....</b>	<b>73</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>75</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>77</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>78</b>
	<b>SEZNAM TABULEK.....</b>	<b>81</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>82</b>

## ÚVOD

V současné době je nedostatek programátorů a absolventů technicky zaměřených vysokých škol. A tak je ideálním řešením zkombinovat zábavu společně s výukou nové generace mladých programátorů a techniků pomocí výuky kroužků robotiky. Díky výuce robotiky si děti osvojí základy konstrukce robotů, elektrotechniky a programování. Takto získané zkušenosti budou užitečné v průběhu celého studia nebo i déle.

Má práce se zabývá návrhem robotického systému, který je vhodný pro výuku základů robotiky. Před návrhem samotným je důležité analyzovat již existující robotické systémy. Zjistit jejich výhody a slabé stránky, jako je konstrukce robotů, možnosti programování a cena jednotlivých řešení. A následně tyto poznatky přesunout do návrhu nového systému. Pro systém musí být navržena centrální jednotka, která umožňuje jednoduché a bezpečné připojení sensorových a akčních modulů, které se budou dědi učit zabudovat do svých vlastních robotů a pomocí programování tak rozpohybují své roboty.

V druhé části teorie jsou popsány principy funkce a ovládání jednotlivých akčních a sensorových členů, které budou používány ve výuce robotiky.

První praktická část se zabývá návrhem schémat a desek plošných spojů centrální jednotky a modulů pomocí profesionálního návrhového programu Altium Designer. V tomto programu jsou navrženy všechny části systému od schématické značky až po panely navržených desek, z kterých jsou vygenerovány výrobní podklady. Při návrhu je kladen důraz na to, aby mohl být systém dále rozšiřován novými akčními a sensorovými moduly. Po výrobě prvních desek plošných spojů je nutné desky osadit a otestovat jejich správnou funkčnost a opravit případné problémy.

Po úspěšném oživení centrální jednotky a modulů následuje tvorba ukázkových programů, které prezentují a vysvětlují funkcionalitu systémů. Pro programování je použito vývojové prostředí Espruino, které podporuje programování pomocí javascriptu a grafického programovacího jazyka Blockly, proto jsou ukázkové programy napsány v obou dostupných jazycích. Systém také umožňuje nahrát vlastní firmware a jeho ladění.

Poslední z částí je dokumentace, která popisuje centrální jednotku. Součástí dokumentace jsou tabulkové popisy jednotlivých konektorů a veškerá využitelná funkcionalita,

což usnadní zapojení vlastních modulů nebo dokonce modulů jiných systémů, jako je například Lego Mindstorms. Součástí dokumentace jsou také popisy ovládacích a signalizačních prvků centrální jednotky.

## **I. TEORETICKÁ ČÁST**

## 1 ROBOTICKÉ SYSTÉMY

V současné době se na trhu vyskytuje mnoho robotických systémů. Většina těchto systémů je zaměřena na zkušenější uživatele, než jsou děti studující na základní škole, což může způsobit spoustu problémů při výuce základů robotiky. Jeden z hlavních důvodů je hardwarová konstrukce systému, kdy je používání systému příliš složité nebo může dojít k poškození systému špatným zapojením modulů. Například řídicí deska je osazena pouze pinovými lištami, což umožní špatné připojení modulu a tím dojde k poškození modulu nebo řídicí desky.

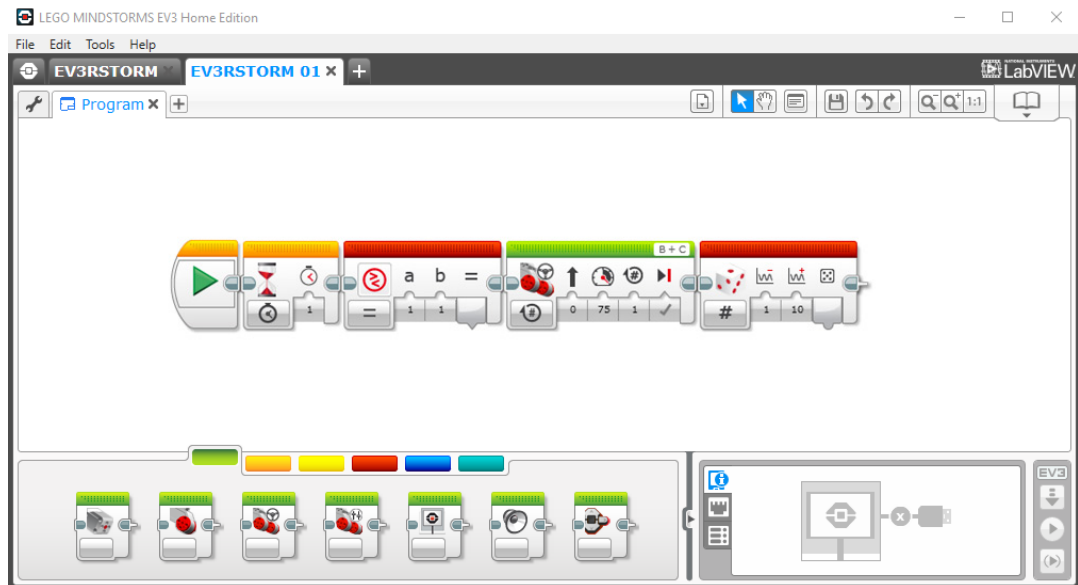
Další důležitou vlastností jsou také možnosti programování. Je důležité, aby vývojové prostředí bylo jednoduché a snadno ovladatelné. Textové programovací jazyky jsou příliš složité pro děti prvního stupně ZŠ, proto je důležitá podpora programování pomocí grafického editoru.

Aby mohly být správně vytvořeny požadavky na hardwarové vybavení navrhovaného systému, bylo nejdříve nutné analyzovat stávající systémy a zjistit jejich výhody a nevýhody a na základě zjištěných informací sestavit požadavky nového systému.

### 1.1 Lego® Mindstorms®

Tento robotický systém se skládá z různých řad robotických stavebnic vyráběných firmou Lego. Díky popularitě stavebnic Lego se tento systém rychle rozšířil a je velmi často používán pro výuku robotiky. Hlavní částí této stavebnice je centrální jednotka nazývána „brick“, ta se stará o komunikaci s jednotlivými moduly a provádí program vytvořený Uživatelem. Kromě dílů Technic obsahují stavebnice také rozšiřující moduly, jako jsou motory, senzory doteku, barvy, vzdálenosti, gyroskop a IR vyhledávač. Od uvedení systému na trh je v současné době prodávána již třetí generace. [3]

Samotné programování probíhá ve vývojovém prostředí LabVIEW s použitím programovacího jazyka „G“, který je založen na modelu „Dataflow programming“, který se velmi liší od sekvenčního modelu většiny populárních jazyků, jako je C/C++, Java. [3]



Obr. 1 Prostředí LabVIEW

### 1.1.1 První generace Lego Mindstorms - RCX

První generace byla nazývána RCX (prodávána od roku 1998). Srdcem této jednotky byl osmibitový mikročip značky Hitachi, který obsahoval 32K paměti pro uložení firmware a uživatelského programu. Samotné programování jednotky probíhalo bezdrátově pomocí IR rozhraní, které bylo možné využít i pro komunikaci mezi dalšími centrálními jednotkami. Pro ovládání modulů bylo možné použít tři vstupní porty pro senzory a tři výstupní motorové porty. K zobrazování informací o stavu baterie a modulů sloužil LCD displej. [13]

V současné době je vybavení tohoto systému již velmi zastaralé a bylo plně nahrazeno novějšími verzemi.



Obr. 2 Lego RCX [13]

### 1.1.2 Druhá generace Lego Mindstorms - NXT

V roce 2006 byla vydána druhá generace systému nazývaná NXT. Tato generace získala mnoho podstatných změn a vylepšení. Centrální jednotka byla vybavena dvěma mikročipy společnosti Atmel. Hlavní AT91SAM7S256 obsahující 32-bit ARM® procesor pracující na frekvenci 48 MHz, s 256 KB FLASH a 64 KB RAM paměti. Druhý koprocesor 8-bit AVR ATmega48 pracující na s frekvenci 8 MHz, 4 KB FLASH a 512 Byte RAM. Komunikace přes IR rozhraní byla nahrazena za více spolehlivé USB a pro komunikaci mezi centrálními jednotkami bylo přidána bezdrátová komunikace Bluetooth. Čtyři vstupní porty umožňují použití analogových i digitálních rozhraní a k výstupním portům byla přidána podpora enkodérů. Pro možnosti zobrazení informací byl černobílý LCD displej nahrazen větším o rozlišení 100 x 64 pixelů. [11]

Tato generace se již nevyrábí, v současné době je výkon procesorů hodně omezující. Navzdory svým parametrům jsou tyto stavebnice stále hojně používány pro výuku robotiky na školách, což je pravděpodobně zapříčiněno vysokou pořizovací cenou nových stavebnic.

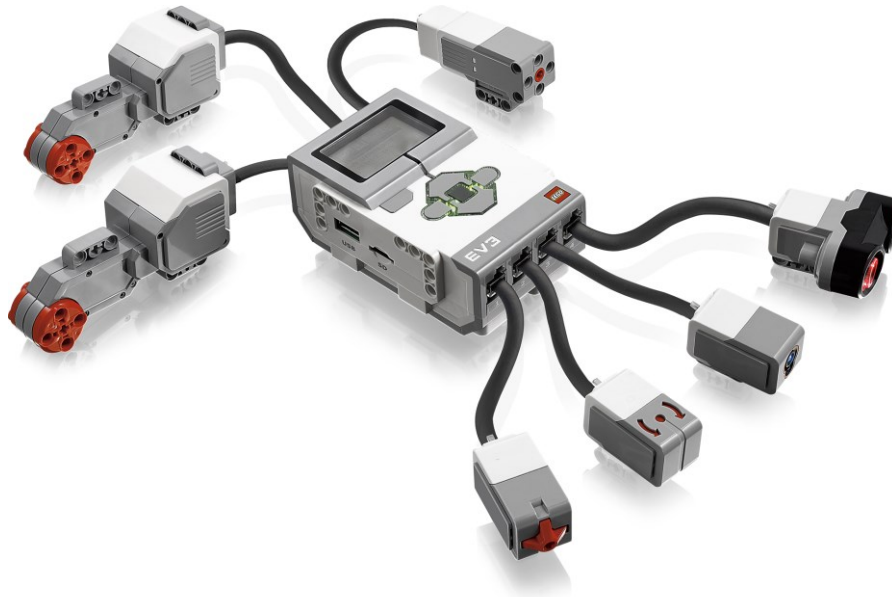


*Obr. 3 Lego NXT [11]*

### 1.1.3 Třetí generace Lego Mindstorms – EV3

Současná třetí generace je označována jako EV3. Opět došlo k výrazným změnám hardwarového vybavení. Předchozí mikropočítač byl nahrazen za 32-bit ARM9 procesor AM1808 od Texas Instrument, který poskytuje dostatek výkonu pro použití operačního systému Linux. Do systému byl také přidán slot pro kartu microSDHC, která zjednodušuje nahrání a spuštění vlastního softwaru a firmweru. Také byla rozšířena konektivita systému, u vstupních portů byla přidána UART komunikace, která je využita u nových senzorových modulů. [3]

Jednotlivé moduly byly vybaveny mikropočítačem STM8S103F3, který zajišťuje zpracování měřených hodnot a informace odesílá do centrální jednotky pomocí komunikace UART. Počet výstupných portů byl zvýšen na čtyři. Rozlišení LCD displeje bylo zvýšeno na 178 x 128 pixelů. [3]



*Obr. 4 Lego EV3 [22]*

#### 1.1.4 Vyhodnocení

Lego Mindstorms je kompletní systém, který je zavedený již dlouhou dobu a má spoustu svých příznivců. Hlavní výhodou tohoto řešení je kompatibilita s populárními stavebnicemi Lego Technic, která přináší rychlost a jednoduchost konstrukce mechaniky robotů. Stavebnice sice umožňuje programování v grafickém prostředí, ale použitý model Dataflow není vhodný pro výuku. Pokud by Lego Mindstorms mělo být použito pro výuku kroužků robotiky, pořizovací cena stavebnic by byla příliš vysoká. Dalším omezením jsou minimální možnosti vytvoření vlastních modulů.

#### Výhody

- Kompletní systém
- Grafický programovací jazyk
- Lze vytvořit vlastní firmware
- Odolná konstrukce

#### Nevýhody

- Programovací jazyk G, založený na Dataflow modelu<sup>1</sup>

---

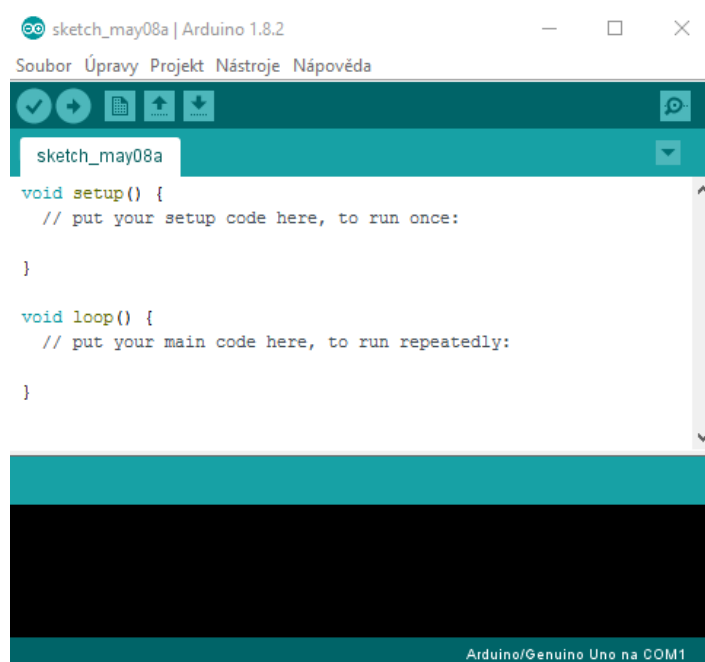
<sup>1</sup> Tuto nevýhodu eliminují projekty alternativních firmware, jako např. LEJOS, OpenRoberta, ev3dev aj.

- Malý počet portů pro moduly
- Cena

## 1.2 Arduino

Vývoj prvního Arduina započal v roce 2005, když se lidé z italského Interaction Design Institute ve městě Ivrea rozhodli vytvořit jednoduchý a levný vývojový set pro studenty, kteří si nechtěli pořizovat v té době rozšířené a drahé desky BASIC Stamp. Mezi studenty se Arduino uchytilo, a tak se tvůrci rozhodli poskytnout ho celému světu. A to nejenom prodejem vlastních desek, ale i sdílením všech schémat a návodů (jedná se o open-source projekt). Programová část Arduina byla založena na Processing, což je programovací prostředí s vlastním editorem, určený k výuce programování. Za osm let vývoje vznikla spousta různých typů desek Arduina. Jelikož se jedná o open-source projekt, vznikla také spousta dalších neoficiálních typů, takzvaných klonů. [14]

Arduino IDE je napsané v jazyce Java a jedná se o software vzniklý z výukového prostředí Processing. To bylo mírně upraveno, byly přidány nové funkce a neposlední řadě podpora knihovny Wiring. Samotné Arduino lze programovat v jazyce C nebo C++. Nejjednodušší variantou je však použití knihovny Wiring. Ta je v současné době pro programování Arduina velmi rozšířená. [14]



Obr. 5 Arduino IDE

### 1.2.1 Typy desek

Srdcem každého Arduina je procesor od firmy Atmel, který je obklopen dalšími elektronickými komponenty. Pro celou řadu desek je typické jednotné grafické zpracování s převažující modrou barvou. Na většině desek je osazen převodník, který umožňuje komunikaci mezi PC (USB) a čipem. [14]

Jelikož je velké množství Arduino desek je v následujících kapitolách popsány pouze jedny z nejrozšířenějších desek.

### 1.2.2 Arduino Nano

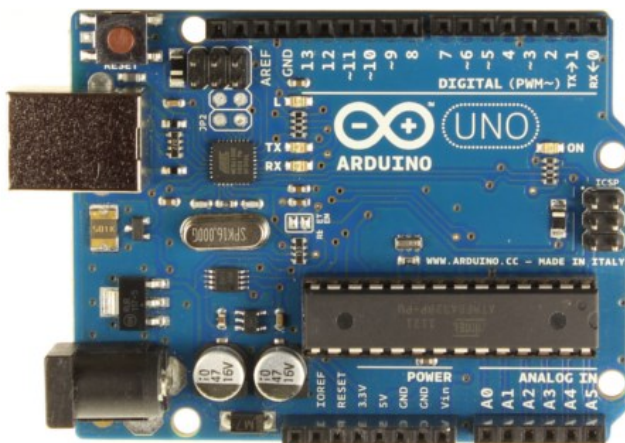
Arduino Nano je malá, kompletní a přátelská deska založená na procesoru ATmega328. Jedná se o nejmenší použitelné řešení, kdy se na desce nachází pouze procesor společně s USB konektorem. Veškerou další funkcionalitu desky je nutno řešit přídatnými moduly. Deska samotná se stará pouze o zpracovávání informací ze senzorů a vysílání řídicích signálů do připojených modulů. [14]



*Obr. 6 Arduino Nano [14]*

### 1.2.3 Arduino Uno

Arduino Uno je v současné době asi nejčastěji používaný typ desky. Je přímým pokračovatelem hlavní vývojové linie, která započala prvním Arduinem se sériovým portem místo USB, pokračující přes Arduino Extreme, NG, Diecimila a Duemilanove až k dnešnímu Uno. Na desce najdeme procesor ATmega328 a již klasické USB. [14]

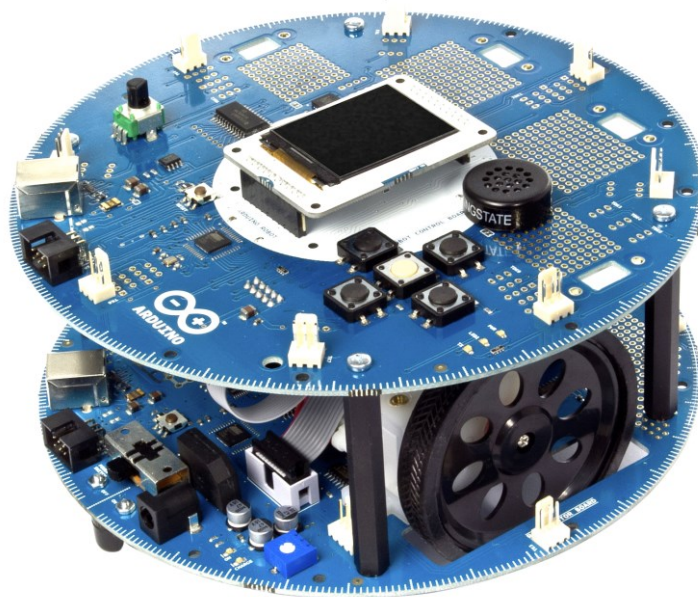


*Obr. 7 Arduino Uno [14]*

V levém horním rohu se nachází tlačítko pro restartování, následně se pod tlačítkem nachází USB konektor typu B, který slouží k připojení desky k PC. Novější modely desek jsou již vybaveny konektorem micro USB. Některé desky tento konektor společně s USB převodníkem postrádají z důvodu úspor. Pokud není přítomný USB konektor je potřeba použít externí převodník, který je zapojen do ICSP hlavičky pro externí programování. Deska je také osazena konektorem pro externí napájení, pokud není napájena přímo z USB. Napájení je signalizováno pomocí LED diody. Po stranách jsou osazeny pinové lišty pro připojení modulů. [14]

#### **1.2.4 Arduino Robot**

Robot je první oficiální Arduino na kolech. Robot má dva procesory, jeden na každé ze svých dvou desek. Motorová deska řídí motory a řídicí panel čte snímače a rozhoduje o tom, jak pracovat. Každá deska je plnohodnotná deska Arduino programovatelná pomocí IDE Arduino. Motorová i řídicí deska jsou ovládány mikropočítačem ATmega32u4. Mikropočítač má mnoho z jeho pinů přímo namapovaných na čidla a pohony. Programování robota je stejné jako pro ostatní desky Arduino. Oba procesory mají vestavěnou komunikaci USB, což eliminuje potřebu externího převodníku. [14]



*Obr. 8 Arduino Robot [14]*

### 1.2.5 Vyhodnocení

Použití desek Arduino je vhodné řešení pro výuku studentů, kteří mají již základní znalosti robotiky a elektrotechniky. Pro studenty základní školy je toto řešení příliš komplikované, připojení modulů není jednoznačné a často by docházelo k poškození zařízení z důvodu špatného zapojení. Další obtíží je nemožnost vyvíjet pomocí programovacího grafického editoru.

#### Výhody

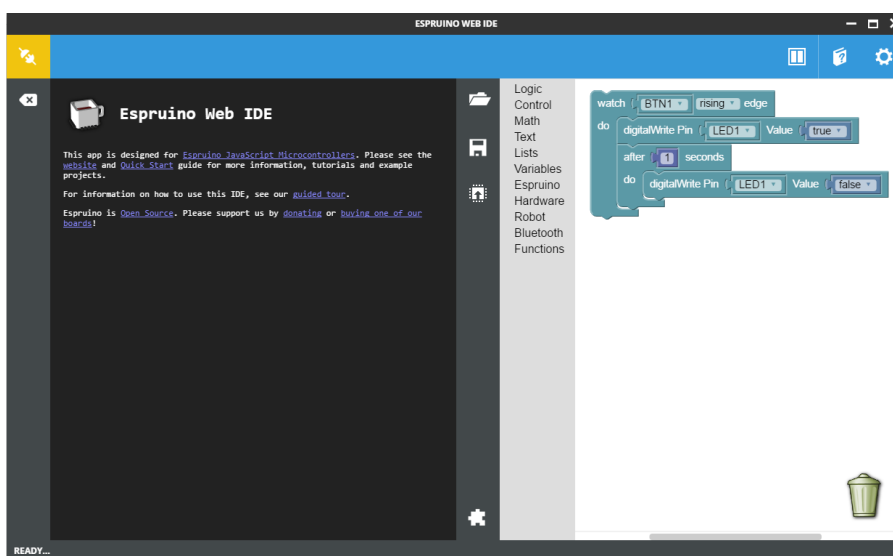
- Cena hlavních desek
- Více druhů hlavních desek
- Dokumentace
- Open-source

#### Nevýhody

- Pouze textové programovací jazyky
- Lze koupit pouze hlavní desky nebo kompletní robot
- Cena kompletního robota
- Nejsou dostupné moduly
- Složitě připojení vlastních modulů
- Konstrukce

## 1.3 Espruino

Jedná se o poměrně mladý systém, který byl financován pomocí crowdfundingové platformy Kickstarter. Espruino se zaměřuje na zjednodušení a zrychlení vývoje na mikropočítačích. Samotný firmware na deskách Espruino pracuje jako Javascriptový interpret, který využívá minimum paměti a výkonu. Vývoj probíhá ve vlastním webovém IDE, které lze jednoduše stáhnout jako doplněk do webového prohlížeče chrome. Velkou výhodou tohoto IDE je možnost programování pomocí Javascriptu a knihovny Blockly. [15]



Obr. 9 Espruino web IDE

### 1.3.1 Blockly

Blockly je knihovna, která přidává editor vizuálních kódů do webových aplikací a aplikací pro Android. Editor Blockly používá grafické bloky, které reprezentují koncepty kódu, jako jsou proměnné, logické výrazy, smyčky a další. Umožňuje uživatelům aplikovat principy programování, aniž by se museli starat o syntaxi nebo zastrašování blikajícího kurzoru na příkazovém řádku. [10]

Z pohledu uživatele je Blockly intuitivní a vizuální způsob, jak vytvořit kód. Z pohledu vývojáře je Blockly v podstatě textové pole, které obsahuje syntakticky správný kód generovaný uživatelem. [10]

### 1.3.2 Espruino Board

Tato vývojová deska je navržena především s ohledem na pořizovací cenu. Srdcem desky je mikropočítač STM32F103RCT6 pracující na frekvenci 72MHz vybavený 256KB flash paměti a 48KB RAM. Pro komunikaci s vývojovým prostředím je deska vybavena micro USB konektorem. Desku lze napájet pomocí bateriového konektoru, aby nebylo nutné napájení pomocí USB konektoru. Vybavením desky jsou pouze tři led diody a dvě tlačítka. Pro další funkcionalitu desky je nutné vytvořit vlastní moduly, které je nutné k desce připájet nebo desku vybavit potřebnými konektory. [15]



*Obr. 10 Espruino Board [15]*

### 1.3.3 STM Discovery

Jelikož je možné firmware Espruino použít i na jiné než oficiální desky, může být použita celá řada vývojových sad STM Discovery. Hlavní podmínkou pro použití firmware je dostatečná velikost paměti mikropočítače.

Vývojové desky Discovery poskytují rychlý a snadný způsob vývoje pro mikropočítače STMicroelectronics. Každá deska obsahuje vestavěný ST-LINK pro programování a ladění pomocí konektoru Mini-B USB. Desku lze napájet přímo z USB nebo pomocí externího zdroje. Všechny desky obsahují také regulátor napětí, tlačítko pro resetování, uživatelské tlačítko, led diody a SWD komunikaci. Volné piny mikropočítače jsou vyvedeny na pinové lišty

### 1.3.4 Vyhodnocení

System Espruino poskytuje ideální programové vybavení pro výuku robotiky. Webové vývojové prostředí je lehce dostupné, jednoduché a intuitivní. Hlavní nevýhodou jsou dostupné desky, které jsou vybaveny pouze základními prvky pro programování mikropočítače, proto je nutné vytvořit vlastní moduly.

#### Výhody

- Cena hlavních desek
- Více druhů hlavních desek
- Dokumentace
- Open-source
- Možnost použití grafického programování

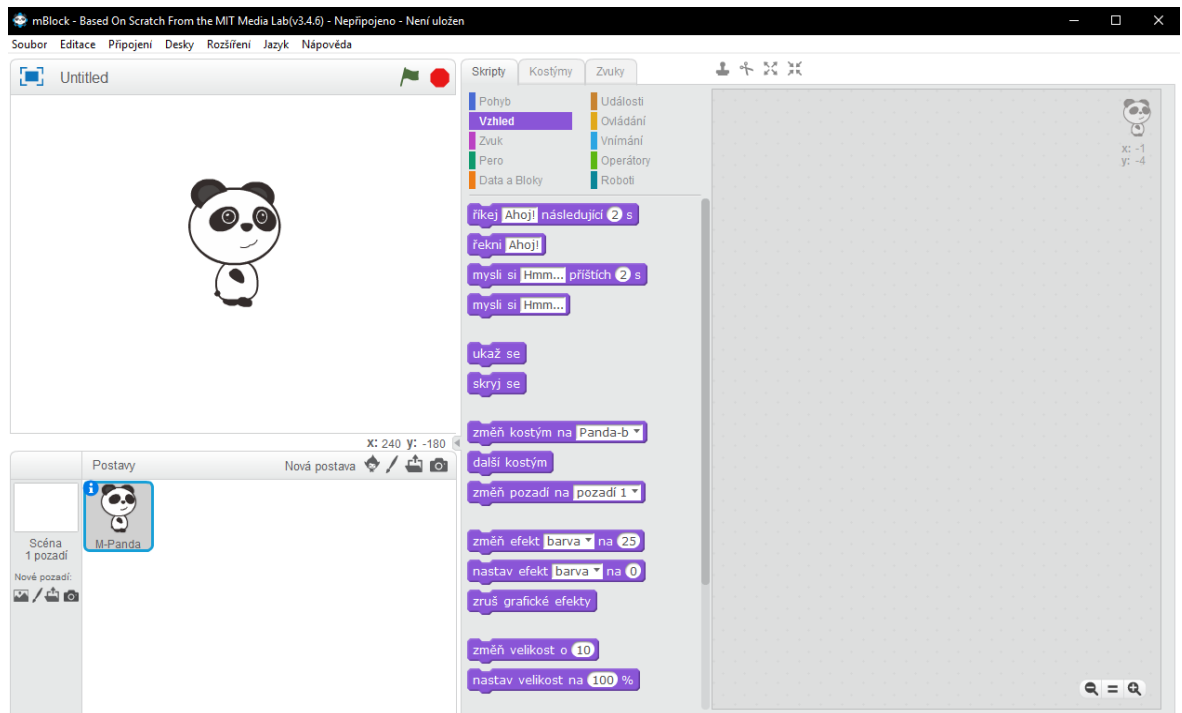
#### Nevýhody

- Nejsou dostupné moduly
- Složitě připojení vlastních modulů
- Konstrukce

## 1.4 Makeblock

Makeblock je robotický systém využívající hliníkovou konstrukční platformu. Díky Makeblock lze vytvářet profesionální a stabilní roboty a stroje. U platformy Makeblock je nezbytným nástrojem pro sestavení robota pouze šroubovák, který je potřebný ke spojování jednotlivých dílů stavebnice. Není potřeba žádné pájení nebo speciální dovednosti. Centrální jednotka je založena na open-source platformě Arduino. Design desek byl přepracován, aby konstrukce byla odolnější a jednodušší k zapojení modulů. Pro celý systém je dostupná řada senzorů, které komunikují s centrálními jednotkami pomocí UART komunikace. [16]

Pro programování lze využít standardní Arduino IDE, ale pro výuku méně zdatných studentů je možnost využití mBlock IDE, což je grafické programovací prostředí založené na open-source programovacím jazyku Scratch 2.0, které umožňuje snadné programování projektů. IDE mBlock také podporuje český jazyk. [16]



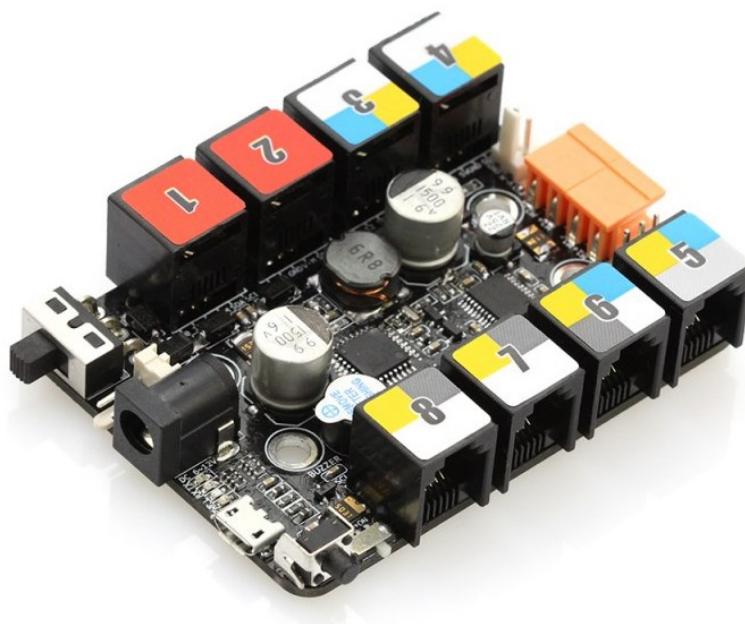
Obr. 11 mBlock IDE

### 1.4.1 Scratch

Scratch je programovací jazyk a online komunita, kde děti mohou programovat a sdílet interaktivní média, jako jsou příběhy, hry a animace s lidmi z celého světa. Jak děti vytvářejí se systémem Scratch, naučí se tvořit, spolupracovat a systematicky rozumět. Scratch je navržen zejména pro věk 8 až 16 ale je používán lidmi každé věkové kategorie. Scratch je navržen a udržován skupinou celoživotních škol v MIT Media Lab. [9]

### 1.4.2 MakeblockOrion

Makeblock Orion je hlavní ovládací panel vylepšený a zdokonalovaný pro výuku. Ovládací panel je založen na bázi Arduino Uno, proto je zde použit čip Atmel ATmega328-AU. Nejedná se o příliš výkonný čip, ale tento nedostatek nahrazují čipy osazená na sensorových deskách, které se starají o zpracování měřených a akčních veličin. S kvalitním napájecím zdrojem o maximálním výkonu 36W (3A) může pohánět současně dva stejnosměrné motory. Řízení motorů obstarává driver TB6612, který lze také použít pro řízení krokového motoru. Dobře navržený barevný systém konektorů je jednoduše rozeznatelný, pro použití se senzory jsou moduly. K připojení slouží uživatelsky přívětivé porty RJ25 implementující jednoduché spojení. [16]



Obr. 12 Makeblock Orion [16]

### 1.4.3 Vyhodnocení

Makeblock je kompletní robotický systém, ale hliníková konstrukce stavebnic zvyšuje cenu. Konstrukci některých robotů může také ztížit nutnost šroubových spojů, které jsou náchylné k snadnému povolání, zejména pokud uvažujeme, že roboty budou stavět děti ze základní školy. Integrovaní napájecího zdroje zjednodušilo možnosti napájení desky, ale prostor, který byl pro zdroj využit, znemožnil osazení ovládacích prvků a signalizace, proto je nutné používat další moduly. Barevný systém pro připojení modulů je jednoduše rozeznatelný.

#### Výhody

- Více druhů hlavních desek
- Odolná konstrukce
- Možnost použití grafického programování

#### Nevýhody

- Složitá konstrukce modulů
- Chybějící základní signalizační a ovládací prvky na hlavní desce
- Náročnost vytvoření vlastních modulů
- Hliníková konstrukce stavebnice

## 2 VYBAVENÍ VÝVOJOVÉHO SYSTÉMU

V této kapitole jsou popsány komponenty navrhovaného systému. Na základě předchozí analýzy robotických systému byly vybrány komponenty a softwarové vybavení, z kterých je nový systém sestaven. Pro konstrukci robotů byla zvolena stavebnice Lego Technic.

Aby bylo možné jednoduše programovat centrální jednotku, byl vybrán open-source systém Espruino, kde bude využito vývojové prostředí, které umožňuje jednoduché psaní kódu pomocí javascriptu nebo grafického jazyka Blockly.

Mikropočítač pro jednotku byl zvolen od výrobce STMicroelectronics, konkrétně z řady F407, která používá výkonné jádro Cortex-M4 a obsahuje velké množství integrovaných periférií. [1]

### 2.1 Centrální jednotka

Kontrolér nebo řídicí jednotka má tři úlohy:

- **Informační úloha** spočívá ve sběru a zpracování informace poskytované senzorním systémem robota.
- **Rozhodovací úloha** spočívá v plánování geometrického pohybu robotické struktury.
- **Komunikační úloha** spočívá v organizaci informace mezi robotem a jeho okolím.

Řídicí jednotka je složena z **procesoru** a **softwaru**. [6]

### 2.2 Akční členy

Akční členy jsou hnací stroje pracující jako svaly robota, aby změnilly jeho konfiguraci. Akční členy poskytují výkon pro činnost mechanické struktury proti gravitaci, setrvačnosti a vnějším silám modifikujícím geometrické umístění robotické ruky. Akční členy mohou být elektrické, hydraulické nebo pneumatické a musí být říditelné. [6]

#### 2.2.1 Motory

Obvykle se celkový řídicí řetězec skládá z řídicí jednotky (procesor se vstupy a PWM výstupy připojené na spínací tranzistorové můstky), stejnosměrného nebo střídavého motoru a případně zpětné regulační vazby. V závislosti na tom, zda se zpětně sleduje výsledný vytvořený pohyb (otáčení) motoru a využívá se zpětně k řízení a regulaci motorů. [17]

Typy regulace rozdělujeme na:

- **Řízení v otevřené smyčce** – používá se pro nenáročnou regulaci motorů, protože se nevyužívá zpětné vazby, která zajišťuje zpětnou vazbu pro regulaci běhu elektromotoru a informuje o tom, zda je motor ve skluzu nebo zablokován, či zda se otáčí požadovanou rychlostí. Dá se říct, že řízení probíhá "naslepo". [17]
- **Řízení v uzavřené smyčce** - průběh regulace motoru je monitorován, tím je získána zpětná vazba, která umožňuje řídit rychlost otáčení, rozběh i doběh motoru apod. [17]

### 2.2.1.1 Kartáčové stejnosměrné motory

Princip kartáčových motorů je založen na periodickém střídavém přepínání polarit napájecího stejnosměrného proudu pomocí komutátoru při každém otočení rotoru o  $180^\circ$ . Rychlost otáčení se řídí napětím, nejjednodušeji jeho spínáním (PWM modulace napájecího napětí), kdy průměrná hodnota napájecího napětí je dána vzájemným poměrem četností stavů zapnuto / vypnuto. V závislosti na tom, zda se používá zpětná vazba či nikoli, lze algoritmus řízení rozdělit na: [17]

- **Pulsní PWM algoritmus v otevřené smyčce** - vytváří potřebnou úroveň napětí, které je úměrná rychlosti otáčení motoru. Tento způsob lze dobře použít při známé a konstantní mechanické zátěži. V opačném případě se riskuje zablokování motoru. [17]
- **Pulsní PWM algoritmus v uzavřené smyčce a jeho variace** - buď jen jednoduchý PWM algoritmus, který podle zpětné vazby řídí / nastavuje periodu, což je vhodné jen pro pomalu se měnící zatížení hřídele, nebo složitější, ale přesné řízení generování pulsů. Zde se pak řídí střída i frekvence podle požadavku na rychlost otáčení a výsledků zpětné vazby. V případě uzavřené smyčky tak lze přesně řídit rozběh i brzdění stejnosměrného elektromotoru. [17]

### 2.2.1.2 Servomotory

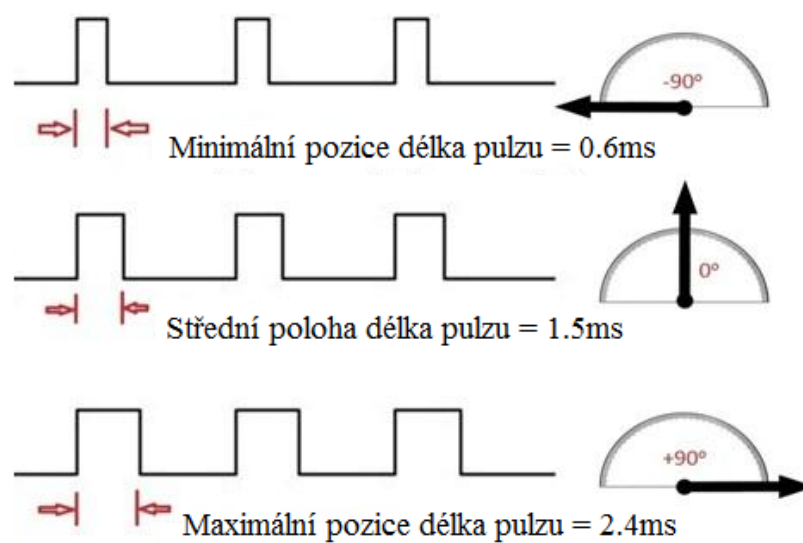
Servomotory jsou klasické elektromotory, které mají převodovku a obsahují řídicí elektroniku se senzorem, pro jednoduché a rychlé nastavování polohy v rozsahu  $180^\circ$  nebo rychlosti otáčení. Tyto motory jsou vhodné pro modely a podobné aplikace zvláště díky jejich nižší spotřebě oproti krokovým motorům, rychlé reakci na změnu a relativně přesným nastavením úhlu a rychlosti. Díky převodům uvnitř motoru, jsou také silnější než klasické elektromotory. [18]



Obr. Servomotor [18]

### 2.2.1.3 Ovládání

Řízení servomotoru se realizuje posláním impulsů přímo do jeho řídicí elektroniky, která vyhodnotí natočení podle šířky impulsu. Impulsy se posílají o frekvenci 30 až 90Hz. Mikro-počítač tedy generuje impulsy PWM. Šířka impulsu může být 0,6 až 2,4ms. [18]



Obr. 13 Ovládání servomotoru [19]

Elektronika uvnitř servomotoru využívá zpětné vazby z elektromotoru, která je tvořena potenciometrem. Ten je připojen k monostabilnímu klopnému obvodu (MKO), ke kterému jsou přivedeny vstupní impulsy. Podle toho se generuje délka impulsu, která se porovnává se vstupními a motorek se otáčí do té doby, než jsou oba impulsy stejné. [18]

## 2.3 Senzory

Prvky použité pro detekci a sběr informací o vnitřních a vnějších stavech se nazývají snímače (**senzory**). V robotice jsou nejdůležitějšími snímanými veličinami poloha, rychlost, zrychlení a síla kloubu. Snímače integrované v robotu posílají informace o každém kloubu a článku do řídicí jednotky a tato řídicí jednotka určuje konfiguraci robota. [6]

### 2.3.1 Ultrazvukový sensor vzdálenosti

Ultrazvukem je nazýváno akustické vlnění, jehož frekvence leží nad hranicí slyšitelnosti. Mezní hodnota frekvence pro toto akustické vlnění je tedy udána hodnotou 20kHz. Ovlivnitelné vlastnosti tohoto vlnění spadají do závislosti na intenzitě (velikost intenzity ultrazvuku určuje ultrazvukovou energii), amplitudě výchylky a frekvenci kmitů, dále je ultrazvuk ovlivněn i vlastnostmi prostředí, ve kterém se šíří. [20]

Ultrazvukové měření je prováděno bezdotykovým měřením. Princip měření vzdálenosti k překážce je založen na principu měření doby mezi vysláním ultrazvukového signálu z vysílače a jeho detekcí na straně přijímače. Doba mezi vysláním a přijetím ultrazvukového signálu je úměrná vzdálenosti vysílače od detekovaného předmětu. K měření vzdálenosti se využívá především dvou principů. První je oddělený zdroj signálu a senzoru, druhý využívá spojeného zdroje signálu se senzorem. [20]

### 2.3.2 IR senzory

IR senzor má schopnost detekovat infračervené světlo odražené od pevných objektů. Dokáže také detekovat infračervené signály vysílané z IR ovladače. IR senzor pracuje ve třech různých režimech: senzor vzdálenosti, senzor barvy a dálkového ovládání.

Senzor se skládá ze dvou částí diody, která vyzařuje infračervené záření, které se odráží od povrchu zpět a ozařuje fototranzistor, který se otevírá v závislosti na síle záření. Hodnotu na fototranzistoru lze měřit jako digitální, kdy senzor funguje jako jednoduchá optická brána nebo lze měřit hodnotu analogově a rozeznávat například vzdálenost od povrchu nebo barvu povrchu. Tento způsob měření je například vhodný pro senzor rozpoznávající čáru.

### 2.3.3 Potenciometry

Potenciometr, je třípólový odpor s posuvným nebo rotačním kontaktem, který tvoří nastavitelný dělič napětí. Potenciometry se běžně používají pro ovládání elektrických zařízení, jako

jsou ovládání hlasitosti zvukových zařízení. Zřídka se používají k přímému ovládání významného výkonu (více než jeden watt), protože výkon rozptýlený v potenciometru by byl srovnatelný s výkonem v řízené zátěži. [23]

### 2.3.4 Enkodéry

Rotační senzory jsou zařízení, která se obecně používají v aplikacích, kde se měří délka, pozice, rychlost, úhel natočení nebo všechny veličiny, které na ně lze převést. Provádějí přeměnu mechanického pohybu na elektrické signály. Běžné rotační enkodéry se dělí na dvě skupiny:

- inkrementální - generují sled pulsů
- absolutní - produkují sled unikátních kódů

Konstrukčně jednodušší inkrementální provedení při rotačním pohybu generuje pulsy, jejichž počet odpovídá určité vzdálenosti, posuvu nebo natočení a závislost počtu pulsů na časový úsek odpovídá rychlosti. Výstupní signál je tedy relativní a informuje jen o pohybu, ne však o přesné pozici. To případně na základě "startovací" pozice musí vyhodnocovat až externí měřicí jednotka (například mikropočítač). Konstrukčně složitější absolutní provedení již generují pro každou pozici natočení konkrétní číselnou hodnotu, tedy v každém momentě je přesně definováno v jakém úhlu natočení se enkodér nachází. [21]

#### 2.3.4.1 Konstrukce

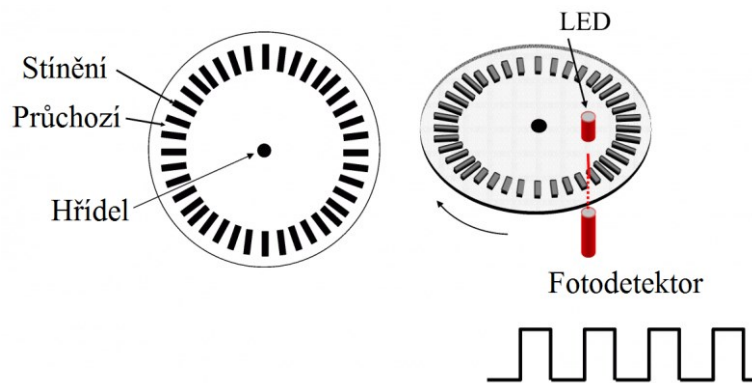
Velmi stručně se dá uvést, že rotační enkodér je elektromechanický převodník, který převádí rotační pohyb na sekvence elektrických digitálních impulsů. V případě rotačního optického enkodéru je celý systém enkodéru složen z následujících hlavních částí:

- Disk s posloupností opticky průhledných a neprůhledných otvorů nebo odrazných a neodrazných plošek
- Optickým vysílačem - nejčastěji diodou LED vysílající světlo ve viditelném nebo infračerveném spektru
- Optickým přijímačem - fototranzistor nebo fotodioda
- Případně může být součástí enkodéru i optika pro zaostření optického paprsku [21]

#### 2.3.4.2 Princip funkce

Disk je spojen s otočnou spojovací hřídelí enkodéru, která je pak v praxi připevněna k hřídeli motoru. Hřídel se tedy otáčí současně s diskem. Světlo vysílané světelným zdrojem (diodou

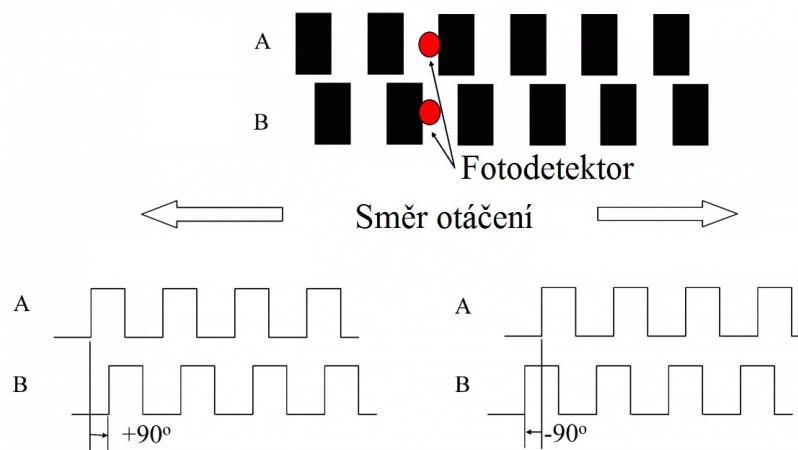
LED), který je umístěn před diskem, prochází diskem pouze průhlednými okénky, zatímco zbývající neprůhledná část disku pohlcuje světlo. Světelné impulsy vytvářené otáčením disku aktivují optický snímač, který je převádí na elektrické impulsy nebo obdélníkový signál. Aby bylo u snímače zajištěno generování obdélníkových impulsů bez rušení, musí být elektrický signál zesílen. [21]



Obr. 14 Funkce enkodéru [21]

### 2.3.4.3 Vyhodnocení signálu

Inkrementální enkodér obvykle generuje dva obdélníkové průběhy, vzájemně posunuté o 90 elektrických stupňů, které se nazývají kanál A a kanál B. Zpracováním samotného kanálu A lze získat informaci o rychlosti otáčení, pomocí druhého kanálu je možné podle sekvence stavů generovaných oběma kanály rozeznat i směr otáčení. [21]



Obr. 15 Vyhodnocení signálu enkodéru [21]

## **II. PRAKTICKÁ ČÁST**

### 3 HARDWAROVÁ REALIZACE SYSTÉMU

Pro realizaci hardwarového vybavení systému bylo nutné navrhnout schémata a desky plošných spojů centrální jednotky a všech modulů. Pro návrh byl použit program Altium Designer. Pomocí tohoto programu lze navrhnout kompletní schémata a desky plošných spojů.

#### 3.1 Centrální jednotka

Centrální jednotka je srdcem celého systému. Zajišťuje propojení všech komponent. Při návrhu bylo dbáno na to, aby jednotka byla co nejvíce univerzální a umožnila připojení velkého množství různých modulů, při zachování dostatečné odolnosti celého systému.

##### 3.1.1 Blokové zapojení

Zapojení jednotky je rozděleno do 10 bloků. Středem celého zapojení je 32 bitový mikropočítač, který zajišťuje řízení všech bloků.

V Horní části jsou dva bloky sloužící k signalizaci a základnímu ovládání. Led panel se skládá z červených a zelených diod, které lze použít pro základní signalizaci běhu systému. Tlačítkový panel umožňuje uživatelské ovládání běhu programu.

USB blok slouží převážně ke komunikaci s vývojovým prostředím Espruino, ale lze tento blok také využít jako USB OTG.

Senzorový blok reprezentuje čtyři konektory RJ12, které slouží k připojení senzorových modulů.

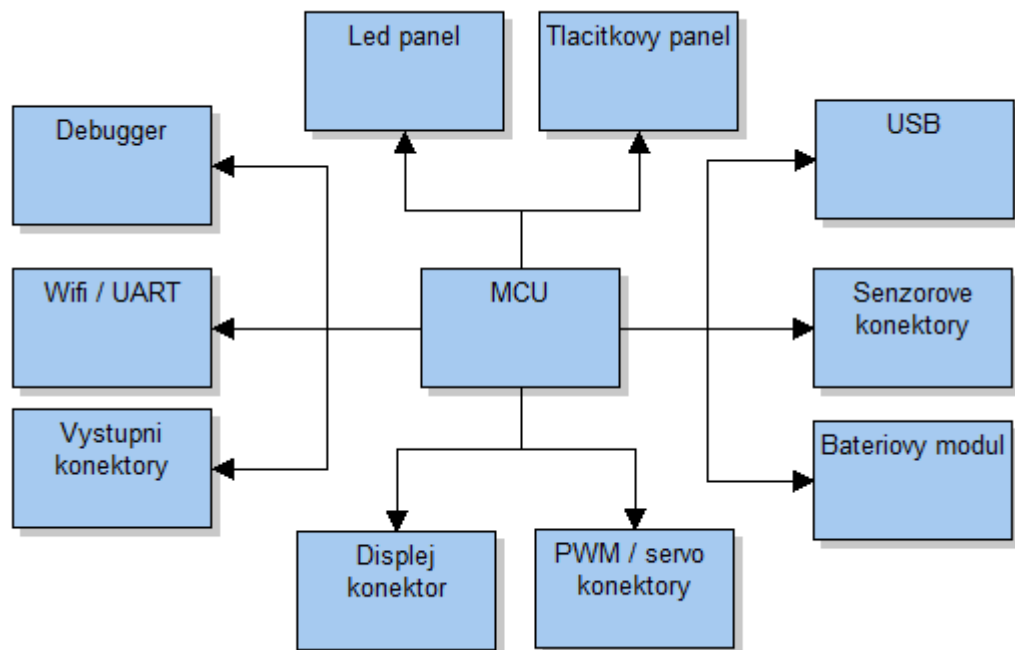
PWM blok je určen pro napájení a řízení servomotorů. Tento blok je také možné použít jako standardní GPIO piny.

Blok výstupních konektorů je určen pro ovládání výstupních modulů, jako je modul stejnosměrného motoru.

Následující blok slouží pro připojení WIFI modulu. Využití tohoto modulu není omezené a lze využít UART nebo SPI komunikace pro připojení dalších modulů.

Předposlední blok reprezentuje standardní SWD komunikaci, která slouží k nahrávání firmware do mikropočítače.

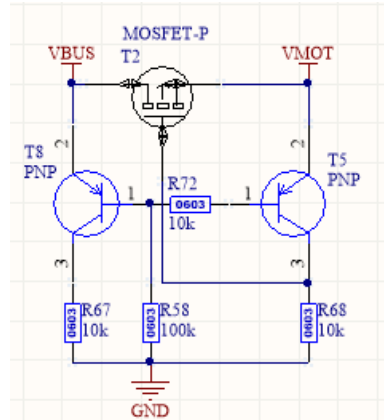
Bateriový modul slouží k napájení centrální jednotky a skrze ni napájí i ostatní moduly připojené do jednotky.



Obr. 16 Blokové zapojení

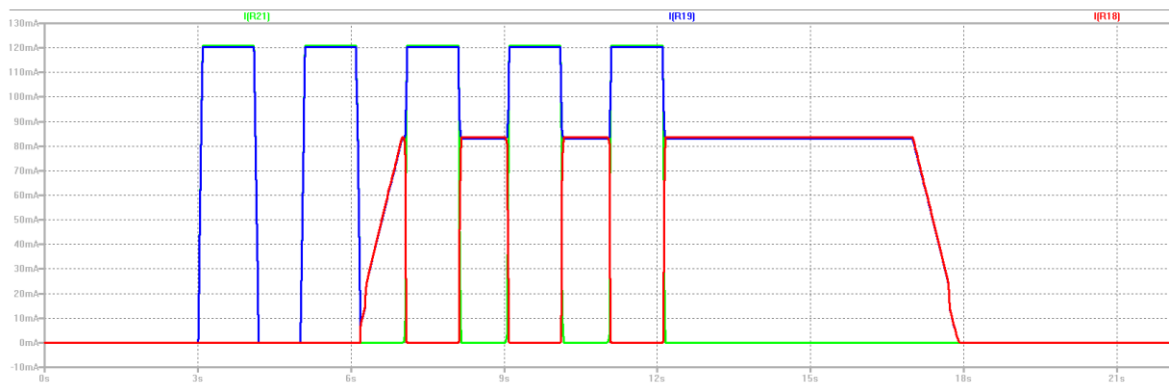
### 3.1.2 Napájení

Napájet centrální jednotku je možné dvěma způsoby. První je pomocí USB konektoru, který slouží pro komunikaci s vývojovým prostředím a druhá z možností je pomocí bateriového modulu. Napájení samotné je rozděleno do tří větví, první 5V větev slouží k napájení elektroniky centrální jednotky, vstupních a výstupních konektorů. Další větev slouží k napájení servomotorových konektorů a poslední větev napájí drivery pro stejnosměrné motory. Účel tohoto rozdělení je zajištění dostatečného napájení při připojení většího počtu modulů. Toto zapojení také umožňuje měnit hodnoty napětí napájení servomotorů a stejnosměrných motorů a tím zvýšit jejich výkon. Všechny tři větve jsou také napájené pomocí USB, aby byla umožněna plná funkčnost desky při napájení pouze z USB. Pro oddělení napájecích větví lze použít diody, ale napěťový úbytek na diodách by způsoboval vysoké výkonové ztráty a tak bylo realizováno zapojení MOSFET tranzistoru jako ideální dioda. MOSFET tranzistory mají velmi nízký odpor a tak způsobují minimální výkonové ztráty v napájení. [4]



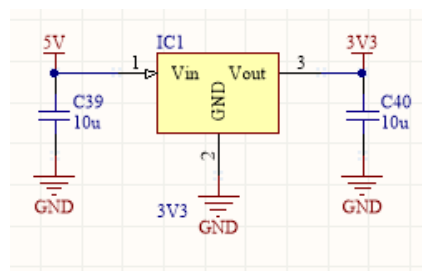
Obr. 17 Zapojení MOSFET tranzistoru

Toto zapojení bylo otestováno pomocí simulačního programu LTSpice. Na obrázku lze vidět simulaci přepínání mezi jednotlivými zdroji napětí. Zátěž zdrojů se rozdělí podle hodnoty napětí napájecího zdroje. Důležitým parametrem je, aby nedocházelo ke zpětnému průchodu proudu mezi zdroji. Modrá křivka reprezentuje proud procházející zátěží, červená křivka znázorňuje USB napájení a zelená křivka bateriový modul.



Obr. 18 Simulace zapojení MOSFETU

Pro napájení mikropočítače je využito napětí 3,3V, které je zajištěno LDO regulátorem napětí, který je napájen 5V větví.



Obr. 19 LDO regulátor centrální jednotky.

Protože mikro počítač využívá AD převodník, je nutné připojit napět'ovou referenci. Bez použití napět'ové reference by docházelo k nepřesnému měření hodnot napětí, protože napět'ový regulátor není schopen poskytnout dostatečně stabilní hodnotu napětí. Zapojení napět'ové reference je stejné, jako předchozí LDO regulátor pouze hodnoty kondenzátorů jsou určeny použitou referencí.

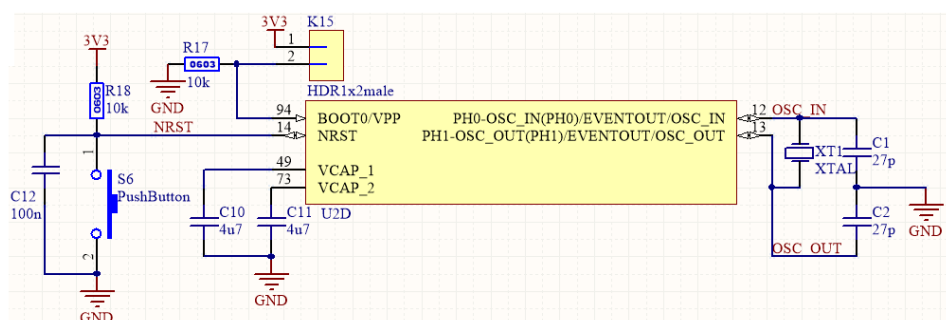
### 3.1.3 Mikro počítač

Pro centrální jednotku byl vybrán 32 bitový mikro počítač STM32F407VET6 od společnosti STMicroelectronics, který má dostatečné hardwarové vybavení pro řízení centrální jednotky. Mikro počítač je založen na architektuře ARM Cortex M4. Tento mikro počítač je vybaven 512 KB FLASH pamětí, 192 KB RAM pamětí a pracuje na frekvenci 168 MHz. Pouzdro mikro počítače bylo zvoleno LQFP-100. [7]

#### Vybavení mikro počítače

- vstupů/výstupů: 82
- Časovačů/čítačů: 10
- Rozhraní: CAN, I2C, I2S, SPI, UART, USB
- AD převodník: 16 kanálů - 12 bit

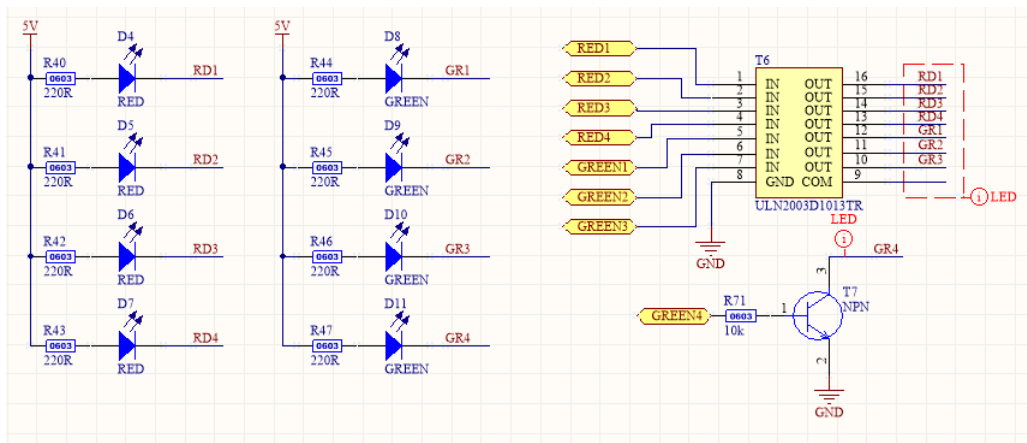
Mikro počítač je vybaven interním krystalem, ale Espruino vyžaduje použití externího 8 MHz krystalu. Také bylo navrženo tlačítko pro restartování mikro počítače. Další důležité zapojení bylo pro nastavení boot režimů, které se provádí pomocí jumperu. Zapojení je zobrazeno na následujícím schématu. Ke každé napájecí větvi mikro počítače jsou osazeny filtrační kondenzátory, které eliminují vysokofrekvenční šum.



Obr. 20 Schéma mikro počítače

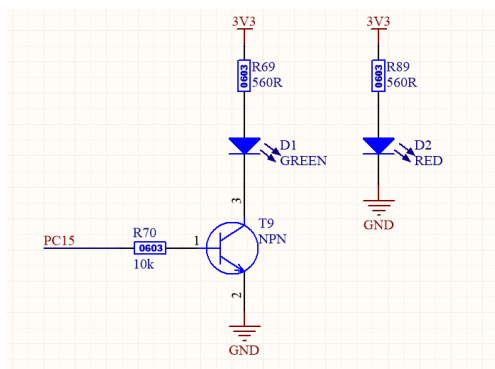
### 3.1.4 Led panel

Aby mohl být signalizován běh programu, byl vytvořen led panel, který se skládá ze 4 zelených diod a 4 červených diod. Jelikož není možné napájet tak velkou zátěž přímo z mikro počítače, bylo nutné použít tranzistorové pole. Led diody mají přesně daný úbytek napětí, a proto bylo nutné dopočítat hodnotu sériových rezistorů, aby byl dosažen požadovaný proud procházející led diodami.



Obr. 21 Schéma zapojení led panelu

System obsahuje ještě další dvě signalizační led diody. První signalizuje napájení a druhá spuštění firmware Espruino. Toto zapojení umožní jednoduché rozpoznání funkčnosti systému a identifikaci problému s napájením nebo firmwarem.

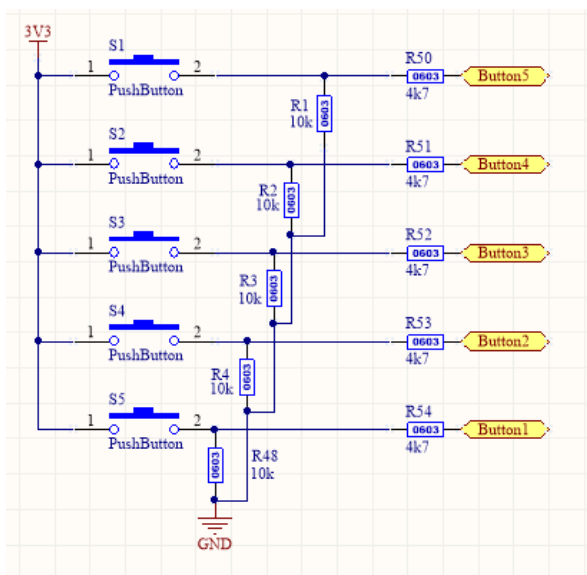


Obr. 22 Schéma zapojení signalizace

### 3.1.5 Tlačítkový panel

Pro možnost ovládání systému bez nutnosti zapojení externích modulů byl vytvořen tlačítkový panel skládající se z pěti tlačítek. Espruino neumožňuje nastavení interních pull down rezistorů mikro počítače, proto musí být rezistory zapojeny externě, jinak by mohlo docházet

k situaci, kdy tlačítka zůstanou v neurčitým stavu. Ke každému tlačítku je připojen do série rezistor omezující proud, procházející tlačítkem.



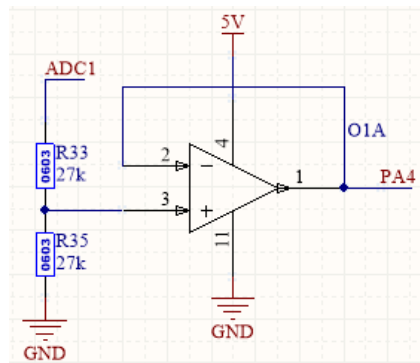
Obr. 23 Schéma zapojení tlačítkového panelu

### 3.1.6 USB

Připojení USB je velmi důležité zajišťuje komunikaci mezi vývojovým prostředím Espruino a mikropočítačem. Do zapojení byl přidán integrovaný obvod pro kompletní elektrostatickou ochranu. Velmi nízká parazitní kapacita a minimální proudové ztráty umožňují použití plné rychlosti USB 2.0. Aby nedocházelo k problémům v komunikaci je nutné signální vodiče na desce plošných spojů vést jako diferenciální pár. Jelikož mikropočítač podporuje USB OTG byla deska osazena čipem pro spínání napětí na výstup. Toto zapojení umožňuje napájení externího zařízení připojeného do USB. Maximální výstupní proud je 500 mA a funkce výstupu je signalizována led diodou. Spouštění výstupního napětí je nutné ovládat softwarově. Pokud dojde k poruše napájení například přetížení je porucha signalizována zpět do mikropočítače.



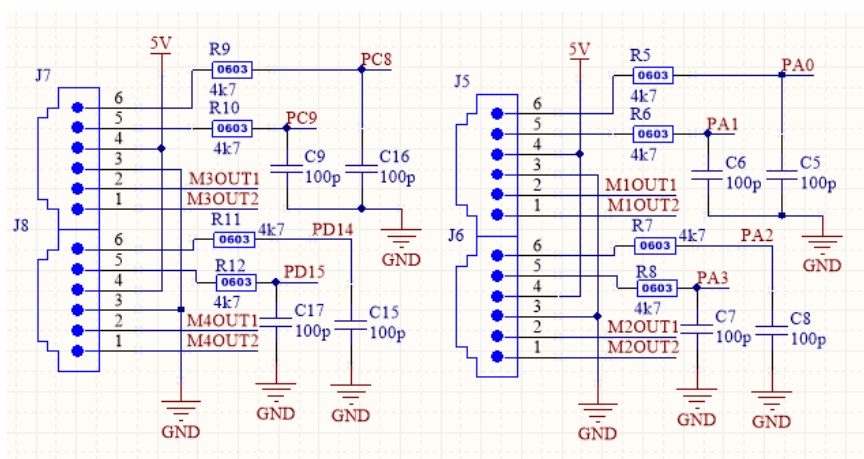
sníží měřenou hodnotu. Následně je zapojen operační zesilovač v zapojení napětového sledovače. Tímto zapojením je docíleno minimálního ovlivnění měřené hodnoty a ochrana mikropočítače před případným vysokým napětím, jelikož se na výstup operačního zesilovače nedostane vyšší napětí než je napájení operačního zesilovače.



Obr. 26 Schéma napětový dělič a sledovač

### 3.1.8 Výstupní konektory

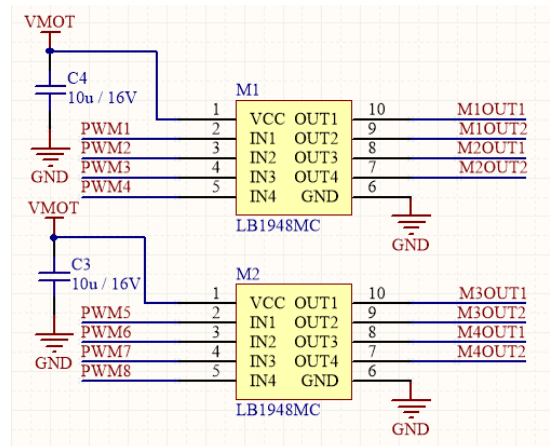
Výstupní konektory jsou určeny převážně pro řízení akčních členů, jako jsou stejnosměrné motory. Aby bylo možné získat zpětnou vazbu o pohybu akčních členů, jsou ke konektorům připojeny dva digitální vstupy / výstupy, které lze použít pro čtení hodnot enkodérů. Pro napájení připojených modulů je připojeno napětí 5 V a GND. Zbylé dva vodiče jsou určeny pro řízení stejnosměrných motorů. Ovládání výstupního výkonu je zajištěno pomocí motorových driverů.



Obr. 27 Schéma zapojení výstupních konektorů

Pro řízení motorů byl zvolen driver LB1948. Jedná se o dvoukanálový obousměrný motorový driver, který pracuje s napětím až 18 V a výstupním proudem 800 mA na kanál. Využití tohoto driveru je několik řízení dvou stejnosměrných motorů, jednoho motoru zapojeného

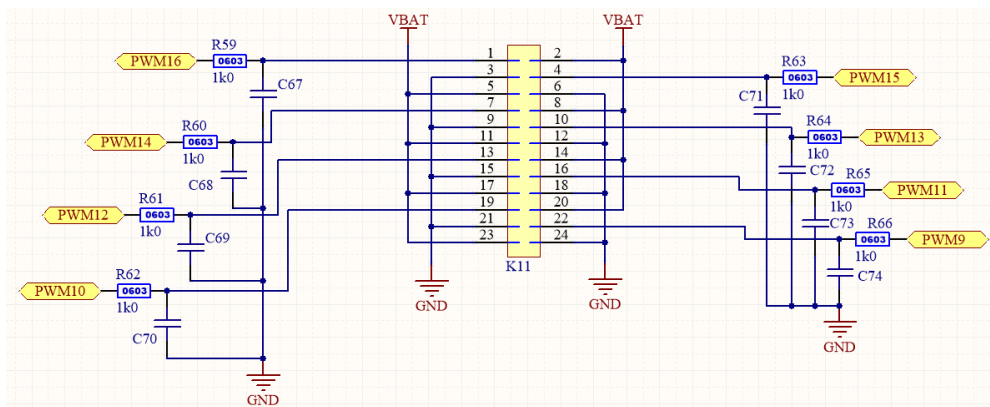
paralelně a jednoho dvoufázového bipolárního krokového motoru. Samotné zapojení je velmi jednoduché, spočívá v připojení kondenzátorů k napájení a správném připojení vstupních a výstupních vodičů. Veškeré potřebné součásti jsou integrovány v driveru. Součástí driveru je také ochrana proti přehřátí a rušič jiskření. Aby bylo možné použít výkonnější motory, jsou drivery napájeny samostatnou napájecí větví, na kterou lze připojit napětí až 16 V.



Obr. 28 Schéma driveru

### 3.1.9 Servomotorové konektory

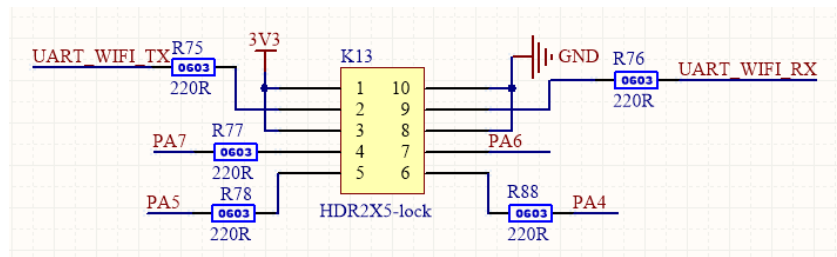
Pro řízení servomotorů je potřebný PWM signál, který je přiveden z mikropočítače. Aby nedocházelo k rušení mezi jednotlivými servomotory, je použit RC filtr, pro dosažení správného filtrování je nutné osadit filtr co nejbližše konektoru. Napájení je řešeno samostatnou větví připojenou přímo k bateriím, aby nedocházelo k výkonovým ztrátám, při regulaci na 5 V. Modelářské servomotory dokáží pracovat s napětím až 7 V, nehrozí tedy k poškození motorů při použití plně nabitých baterií.



Obr. 29 Schéma servomotorového konektoru

### 3.1.10 Wifi konektor

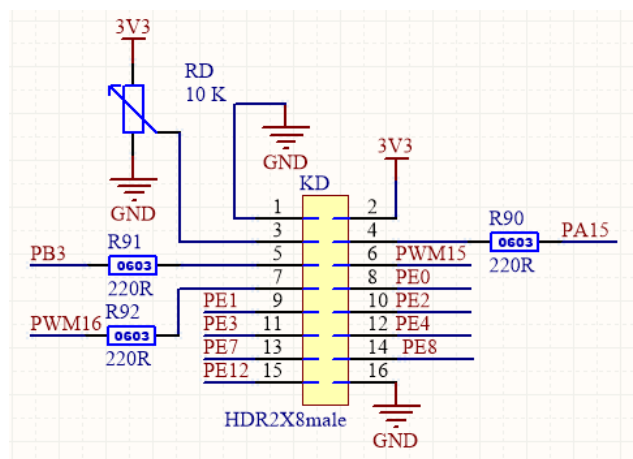
Wifi modul využívá pro komunikaci s mikropočítačem UART. Jelikož bylo velké množství pinů konektoru pro modul nevyužito, byly přivedeny na konektor nevyužité piny mikropočítače, které lze využít pro připojení budoucích modulů. Komunikační vodiče jsou vybaveny terminačními rezistory, aby nedocházelo k odrazům signálu.



Obr. 30 Schéma wifi konektoru

### 3.1.11 Displej konektor

Deska je také vybavena konektorem Micro-MaTch, který slouží k připojení displeje. Zapojení je navrženo pro dva druhy displejů. První z nich je jednoduchý TFT LCD displej, který je řízen paralelním řadičem. Pro regulaci jasu displeje je ke konektoru osazen trimr. Druhou z možností je řízení pomocí komunikace SPI. V současné době ještě nebyly zvoleny konkrétní displeje, pro které budou navrženy redukční desky pro zabudování do systému.

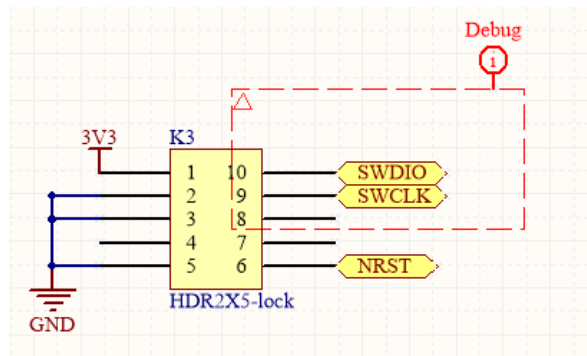


Obr. 31 Schéma zapojení konektoru pro displej

### 3.1.12 Debugger

V první verzi systému byl navržen debugger, který využíval převodníku FTDI FT232H, který umožňuje převádět komunikace jako je JTAG, SPI, I2C a UART na USB. Samotný

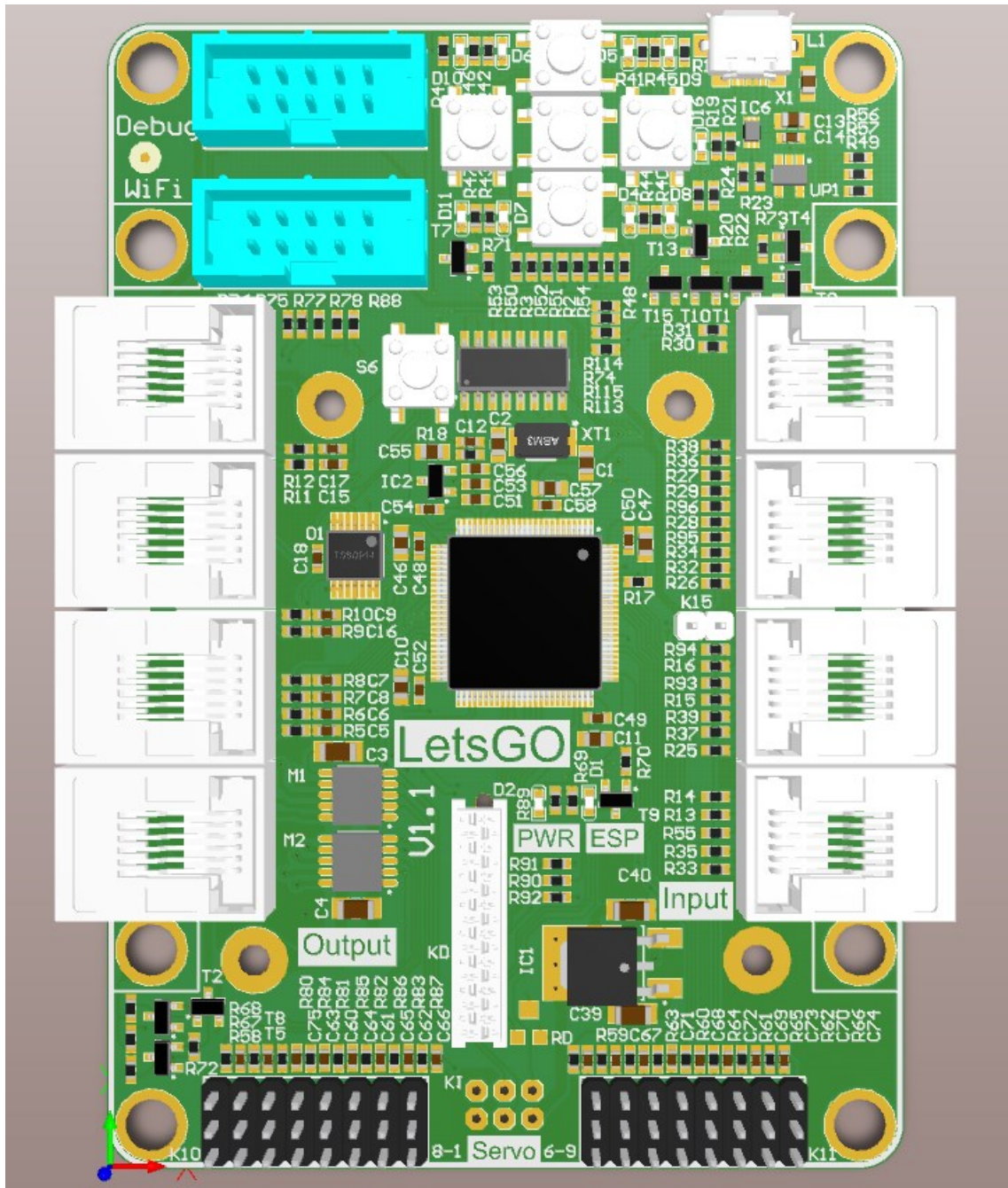
debugger se v počítači hlásí jako COM port. Toto zapojení velmi zjednodušilo připojení jednotky k počítači a nahrávání vlastního firmware. Debugger byl sestaven a otestován, ale vzhledem k vysoké ceně všech komponent debuggeru bylo rozhodnuto, že nebude využit přímo na centrální desce a bude použit standardní konektor pro SWD komunikaci. Debugger s čipem FTDI lze vyrobit jako samostatný modul, pro případy, kdy bude často nahráván vlastní firmware. Schéma zapojení debuggeru je uvedeno v příloze.



Obr. 32 Schéma zapojení konektoru debuggeru

### 3.1.13 Deska plošných spojů

Hlavní deska byla navržena jako dvouvrstvý plošný spoj se součástkami na jedné straně. Využitím pouze dvou vrstev bylo možné dosáhnout úspor při výrobě, ale nevýhodou bylo zkomplikování návrhu vodivých cest, který byl časově mnohem náročnější než s použitím čtyřvrstvé desky. Další komplikací byl prostor pro komponenty. Bylo nutné dosáhnout minimálních výsledných rozměrů desky a tak sebemenší změny v umístění součástek a jejich zapojení znamenaly úpravu velké části vodivých cest desky. Na desku byly umístěny základní popisy, jelikož je na desce velké množství komponent nebylo možné umístit podrobnější informace. V rozích desky jsou umístěny montážní díry, které umožňují jednoduché zabudování desky do robota sestaveného ze stavebnice Lego Technic. [2]



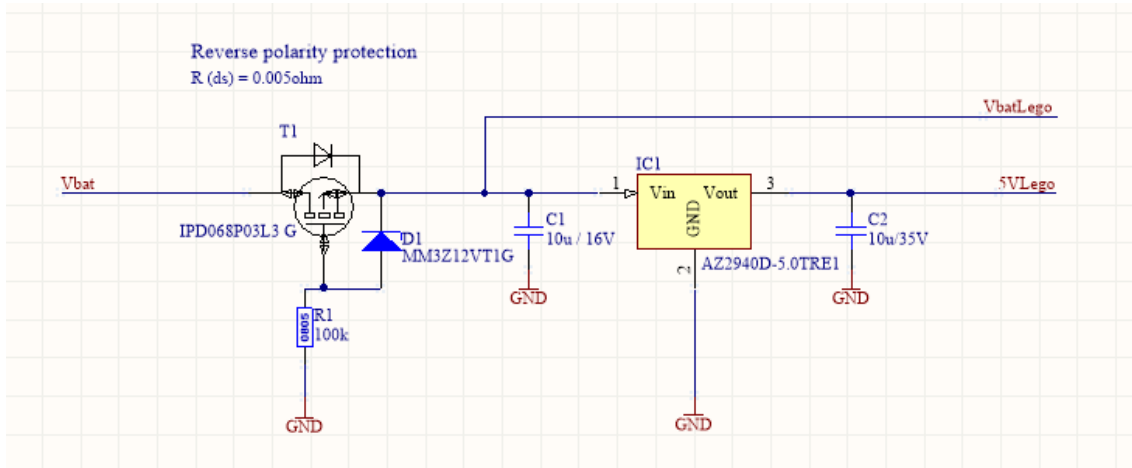
Obr. 33 DPS centrální jednotky

### 3.2 Bateriový modul

Aby nebylo nutné neustálé napájení pomocí USB konektoru, byl vytvořen bateriový modul, který umožňuje připojení AA bateriových článků. Připojení bateriového modulu umožní samostatný pohyb robota bez nutnosti využití jakéhokoli kabelového připojení. Modul je k centrální desce připojen pomocí distančních sloupeků, které také slouží k přenosu elektrické energie.

### 3.2.1 Schéma

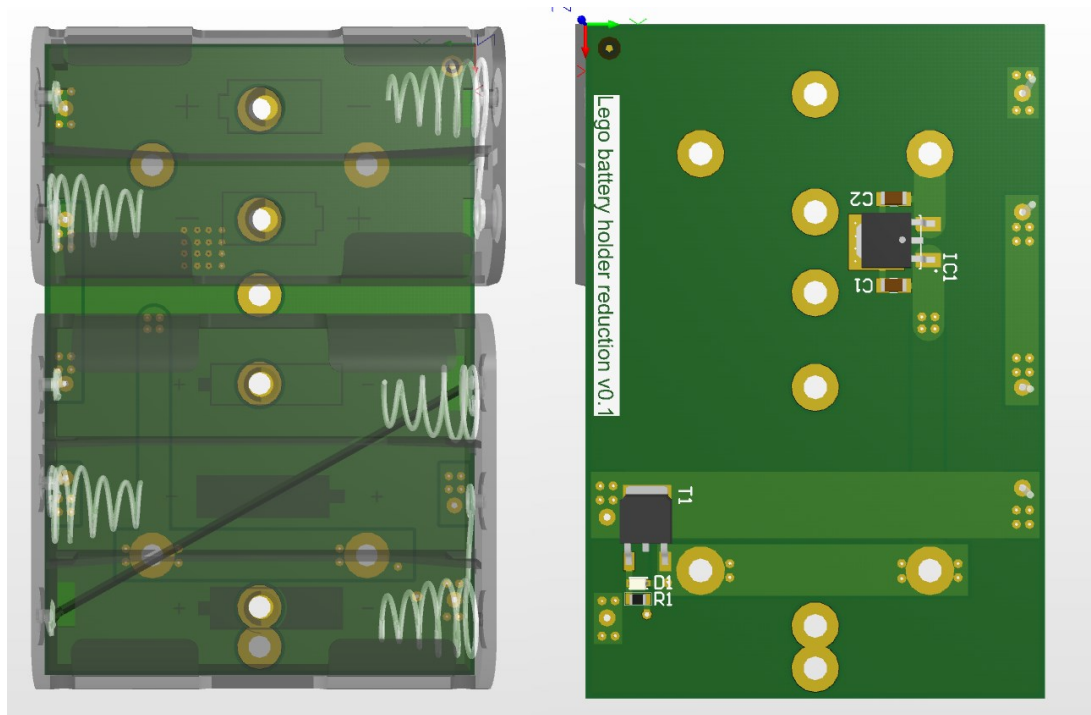
Modul je vybaven ochranným MOSFET tranzistorem, který chrání desku před obrácením polarity bateriových článků. Následně je napětí baterií připojeno do napájecích větví stejnosměrných motorů a servomotorů. Pro napájení centrální jednotky je osazen regulátor na 5V, aby nedošlo k poškození součástek při použití plně nabitých baterií.



Obr. 34 Schéma bateriového modulu

### 3.2.2 Deska plošných spojů

Deska byla navržena jako dvouvrstvý plošný spoj, kdy na jedné straně jsou osazeny komponenty a na druhé držáky baterií. Deska je navržena tak, aby bylo možné použít držáky pro 4 až 5 baterií.

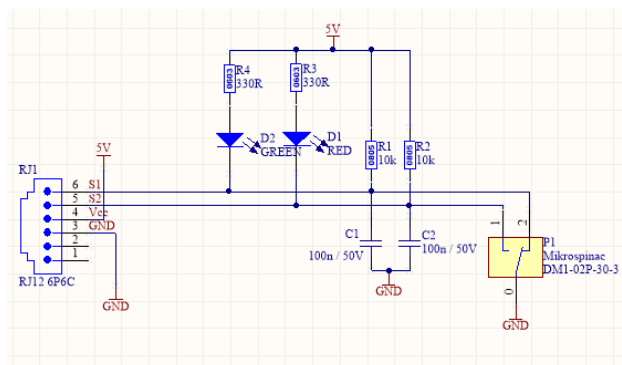


Obr. 35 DPS bateriového modulu

### 3.3 Mikrospínač

Jedná se o jeden ze základních sensorových modulů, který využívá mikrospínače. Signál z modulu je v mikropočítači vyhodnocováno jako zapnuto / vypnuto. Aby měl mikrospínač vizuální zpětnou vazbu, jsou na desku přidány dvě led diody, které se přepínají v závislosti na sepnutí mikrospínače. Připojení k mikropočítači je realizováno pomocí dvou vodičů, kdy jeden je při stisku sepnut a druhý rozepnut.

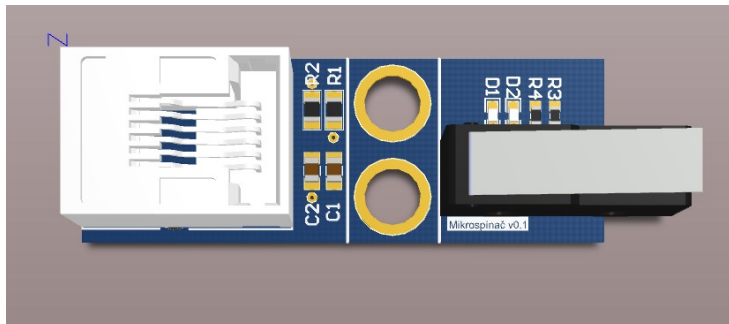
#### 3.3.1 Schéma



Obr. 36 Schéma mikrospínače

### 3.3.2 Deska plošných spojů

Deska je navržena jako dvouvrstvý plošný spoj. Součástky jsou osazeny z jedné strany. Na desku byly umístěny montážní díry pro stavebnici Lego Technic.



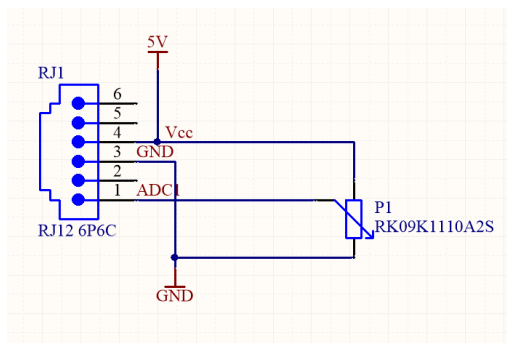
Obr. 37 DPS mikrospínače

## 3.4 Potenciometr

Aby bylo možné využít AD převodníků, byl vytvořen jednoduchý modul s potenciometrem, ten funguje jako dělič napájecího napětí. Pomocí potenciometru lze jednoduše nastavovat například rychlost motorů, což je velmi užitečné například při ladění programu robota sledujícího čáru. Možným vylepšením modulu by mohla být signalizační dioda, která by měnila intenzitu záření v závislosti na otočení potenciometru.

### 3.4.1 Schéma

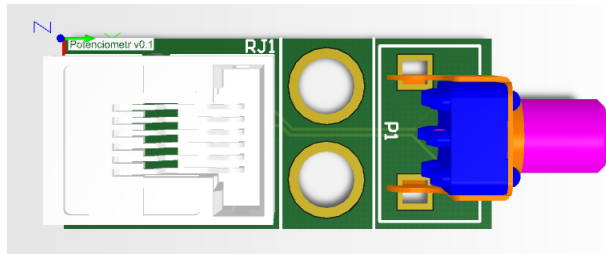
Zapojení modulu je velmi jednoduché, napájení je připojeno k potenciometru. Vzhledem k zapojení konektorů na hlavní desce nejsou potřeba žádné další součástky. Výstupní napětí je připojeno na analogový vodič sensorového konektoru.



Obr. 38 Schéma modulu potenciometru

### 3.4.2 Deska plošných spojů

Deska je navržena jako dvouvrstvý plošný spoj. Součástky jsou osazeny z jedné strany a na desku byly umístěny montážní díry pro stavebnici Lego Technic.



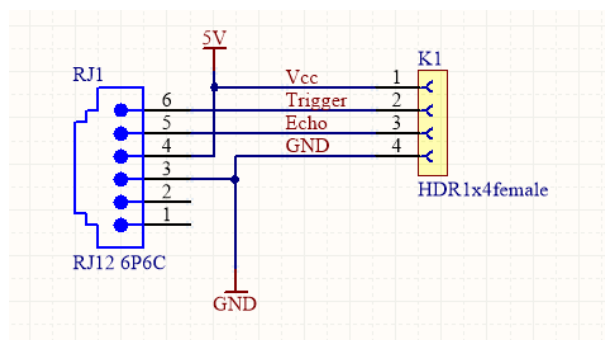
Obr. 39 DPS potenciometru

## 3.5 Ultrazvukový senzor

Ultrazvukový senzor je užitečný modul pro měření vzdálenosti robota od překážky. Tyto senzory dokáží měřit vzdálenosti od 2 cm až do vzdálenosti 4 m. Nejdříve bylo zamýšleno vytvořit kompletní modul senzoru, ale vzhledem cenám komponent s porovnáním ceny funkčních ultrazvukových modulů bylo rozhodnuto, že bude vyroben pouze redukční modul, do kterého lze připojit již prodávané moduly.

### 3.5.1 Schéma

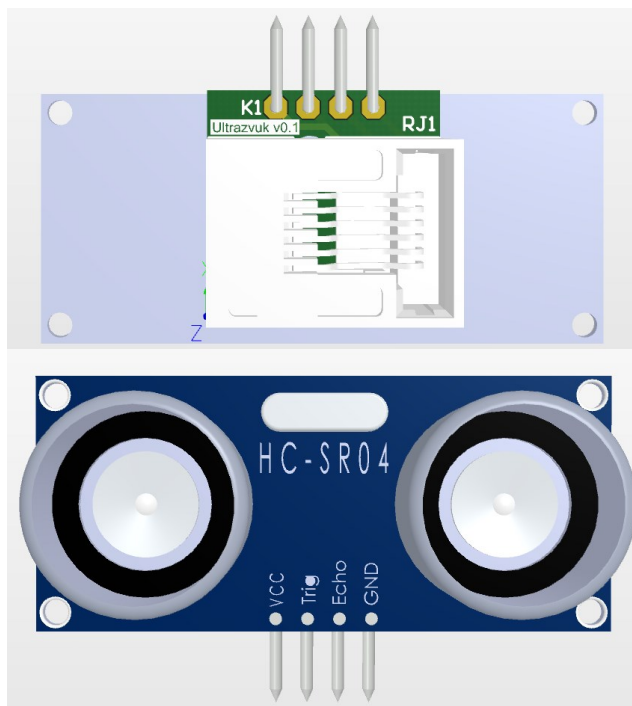
Redukce obsahuje konektor RJ12, který je propojen ke kolíkové zásuvce, ke které bude připojen ultrazvukový modul.



Obr. 40 Schéma redukce pro ultrazvukový modul

### 3.5.2 Deska plošných spojů

Malá redukční destička s konektorem pro připojení centrální jednotky je umístěna na zadní straně ultrazvukového modulu. Pro samotný modul jsou na 3D tiskárně vytvořeny redukce pro zabudování do stavebnice Lego Technic.



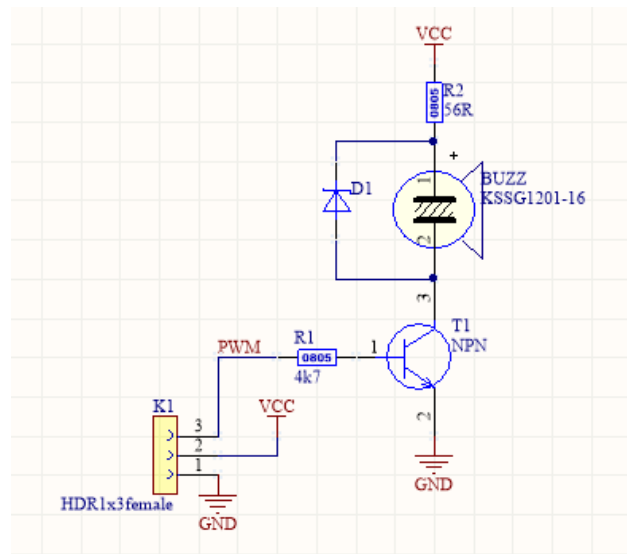
Obr. 41 DPS ultrazvukového senzoru

## 3.6 Bzučák

Jedná se o jednoduchý elektromagnetický audio měnič, který lze připojit do servomotorového konektoru. Zvukový výstup měniče je řízen pomocí PWM signálu. Díky ovládání PWM signálem lze pomocí měniče zahrát melodii nebo jej použít jako sirénu.

### 3.6.1 Schéma

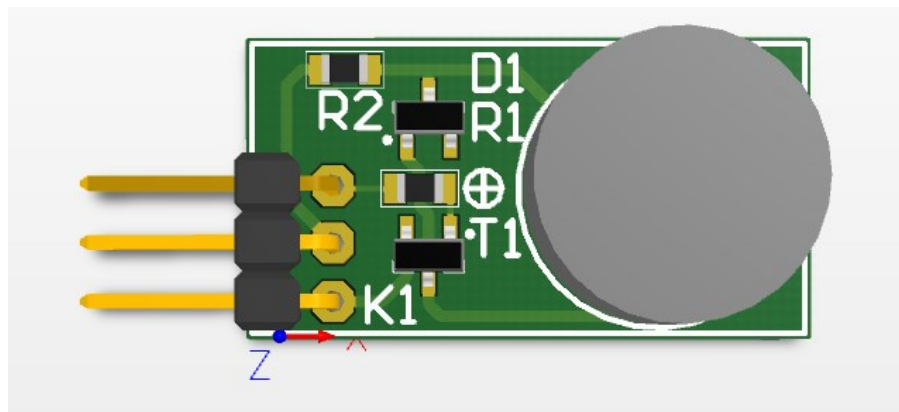
Měnič je řízen pomocí NPN tranzistoru, který je ovládán PWM signálem z konektoru. Do série k měniči je zapojen rezistor, který omezuje proud procházející měničem. Aby nedošlo k poškození při spínání je paralelně k měniči zapojena schottkyho dioda, která uzavírá obvod ve chvíli, kdy je zavírán tranzistor.



Obr. 42 Schéma zapojení bzučáku

### 3.6.2 Deska plošných spojů

Deska je navržena jako dvouvrstvý plošný spoj. Desku by bylo možné vyrábět jako jednovrstvý plošný spoj, ale modul bude vyráběn na panelu s ostatními moduly, které jsou navrženy na dvouvrstvé desce. Součástky jsou osazeny z jedné strany.



Obr. 43 DPS Bzučáku

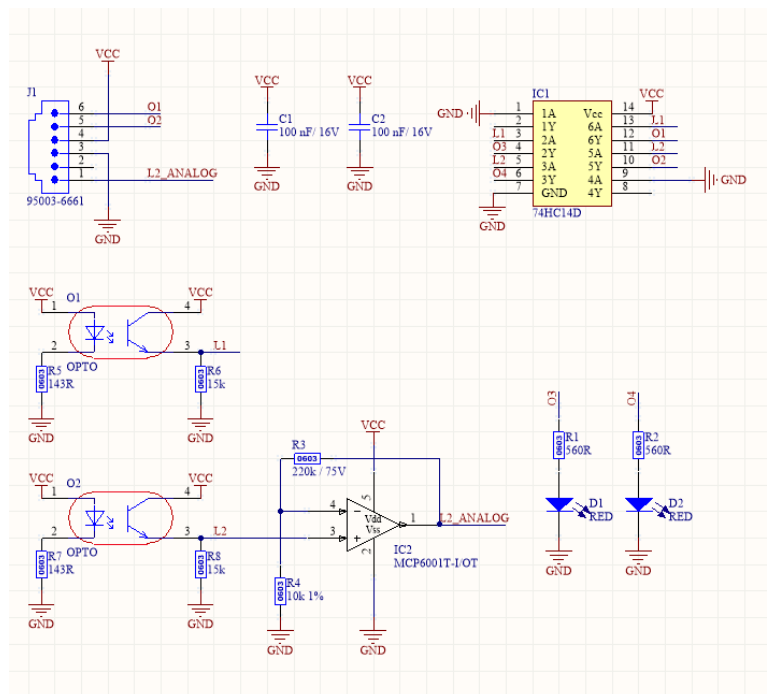
## 3.7 IR Senzor

IR senzor je modul určený pro robota sledujícího čáru. První senzory čáry byly vytvořeny s použitím levných senzorů, které rozeznávaly čáru jako binární hodnotu, tento způsob měření se ukázal jako nedostatečný, protože nebylo možné určit vzdálenost senzoru od čáry. To způsobilo kmitání robota v okolí čáry a snížilo rychlost pohybu. Proto byl vytvořen modul,

který snímá hodnotu i analogovou. Pomocí nově měřené hodnoty lze určovat vzdálenost senzoru od čáry a tím přesněji regulovat pohyb robota.

### 3.7.1 Schéma

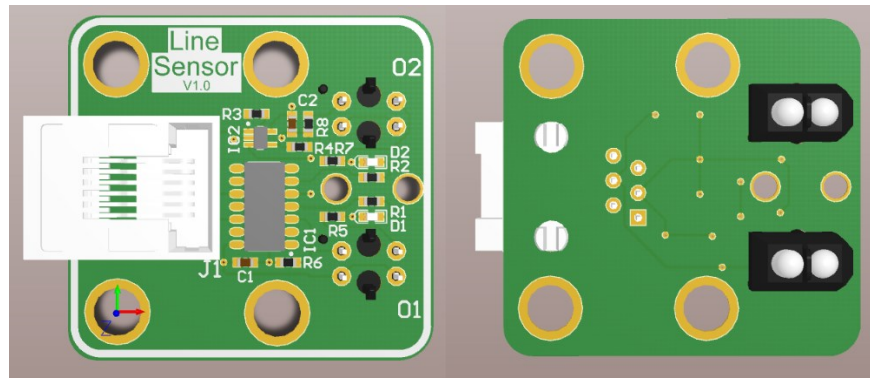
Modul je vybaven dvěma IR vysíláči, signál odražený od povrchu podkladu je odražen do fototranzistorů. Výstup fototranzistorů je přiveden na schmittův klopný obvod, který vytváří přesný obdélníkový signál pro měření binární hodnoty senzoru. Pomocí schmittova klopného obvodu jsou také spínány signalizační led diody. Pro jeden z fototranzistorů je připojen operační zesilovač, který zesiluje hodnotu napětí pro výstup do analogového vodiče konektoru.



Obr. 44 Schéma zapojení IR senzoru

### 3.7.2 Deska plošných spojů

Jedná se o dvouvrstvou desku plošného spoje, v tomto případě jsou součástky osazovány na obě strany modulu. Na horní straně je umístěn konektor a veškeré součástky potřebné pro správnou funkci senzorů, na spodní straně jsou umístěny IR vysíláče s fototranzistory. Aby nedocházelo k rušení mezi jednotlivými vysíláči, jsou na desku umístěny montážní díry pro umístění clony. Na desce jsou také připraveny montážní díry pro zástavbu do stavebnice Lego Technic.



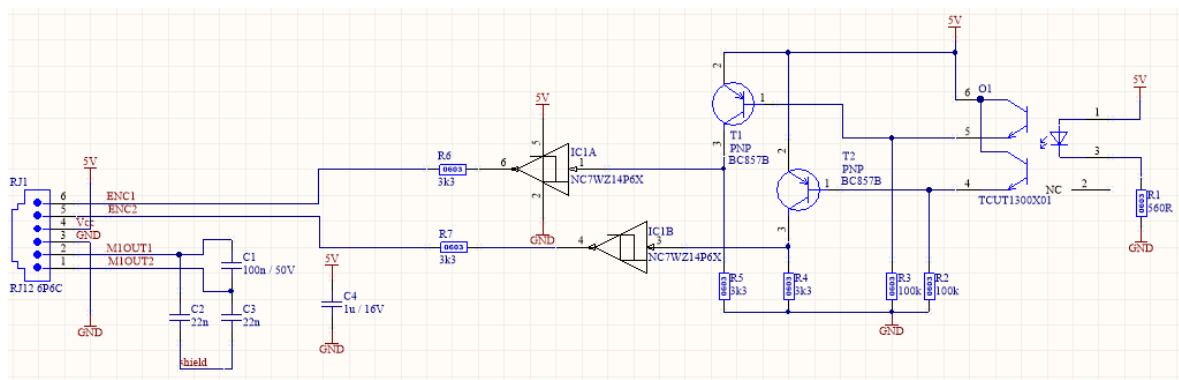
Obr. 45 DPS IR senzoru

### 3.8 DC motor

Pro stejnosměrný motor byl vytvořen modul, který umožňuje jeho bezpečné zapojení pomocí RJ12 konektoru.

#### 3.8.1 Schéma

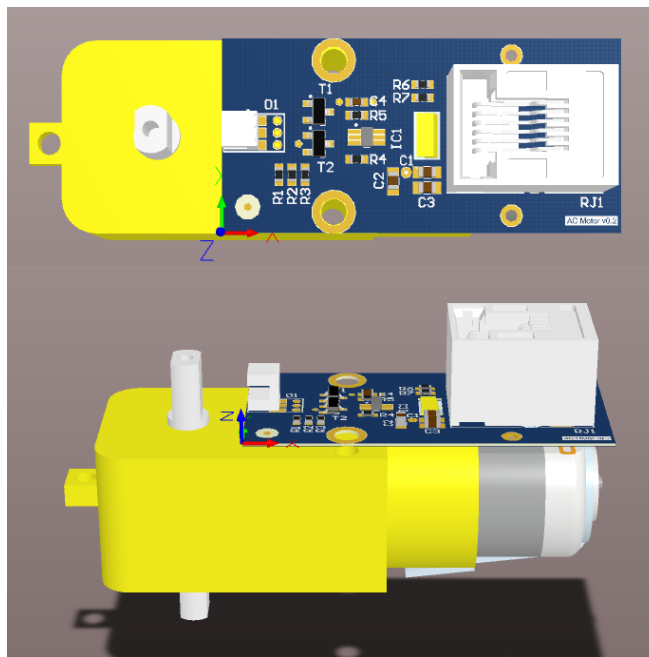
Na modulu se nachází kompletní elektronika pro kvadratický enkodér, který je realizován pomocí led diody a fototranzistorů. Úprava výstupního signálu je zajištěna pomocí PNP tranzistorů, které zesilují signál z fototranzistorů a schmittův klopný obvod, který vyhladí výstup z tranzistoru na přesný obdélníkový signál. Na schématu je vytvořen propoj pro napájení motoru a přidány kondenzátory pro potlačení vznikajícího šumu.



Obr. 46 Schéma zapojení DC motoru s enkodérem

#### 3.8.2 Deska plošných spojů

Modul se pomocí šroubů upevní na stejnosměrný motor. Napájení motoru je nutné připájet k DPS. Na hřídel motoru je nutné umístit disk s dírkami, které využívá enkodér pro rozpoznání pohybu motoru. Na druhou část hřídele je umístěno kolo. Pro zástavbu do stavebnice lego Technic je nutné vytisknout redukční díl pomocí 3D tiskárny.



Obr. 47 DPS DC motoru.

### 3.9 Bateriový modul s nabíjením

Bateriový modul byl navržen pro nabíjení čtyř NiMH článků. Řízení nabíjení obstarává mikropočítač STM32, který spíná nabíjecí napětí na články. Napětí každého článku je neustále monitorováno. Pokud jsou zapojeny baterie s rozdílnými hodnotami, je spuštěno balancování článků, které spočívá v paralelním spínání diod, které omezí nabíjení článků s vysokou hodnotou napětí. Celá nabíječka je napájena pomocí USB konektoru, aby bylo možné nabíjet čtyři sériově zapojené články, byl sestaven vysoce účinný měnič, který napětí dostatečně zvýší.

Tento návrh zůstal pouze jako schématický návrh a dosud nebyl realizován. Finanční náročnost řešení bateriového modulu s vlastním nabíjením je příliš vysoká, a proto je v současnosti používán pouze modul bez nabíječky.

Schématický návrh modulu s nabíjením je uveden v příloze.

## 4 PROGRAMOVÁNÍ CENTRÁLNÍ JEDNOTKY

Pro programování centrální jednotky bylo zvoleno vývojové prostředí Espruino, které je volně dostupné. Aby bylo možné programovat je potřeba provést dva základní postupy. Instalace vývojového prostředí a nahrání firmwaru do centrální jednotky.

### 4.1 Nahrání firmware

Jako první je potřeba ze stránek projektu Espruino stáhnout nejnovější dostupnou verzi firmware pro mikropočítač STM32F407, který je dostupný na stránce <http://www.espruino.com/Download>.

Následně je potřeba připojit k centrální jednotce programovací zařízení pro „flashování“, například jakýkoli STM32 Discovery kit, který je vybaven převodníkem SWD na USB, což umožní použít ST-LINK Utility. V programu je potřeba připojit centrální jednotku, vybrat stažený binární soubor firmwaru a provést nahrání. Nyní je možné připojit centrální desku k IDE Espruino pomocí USB.

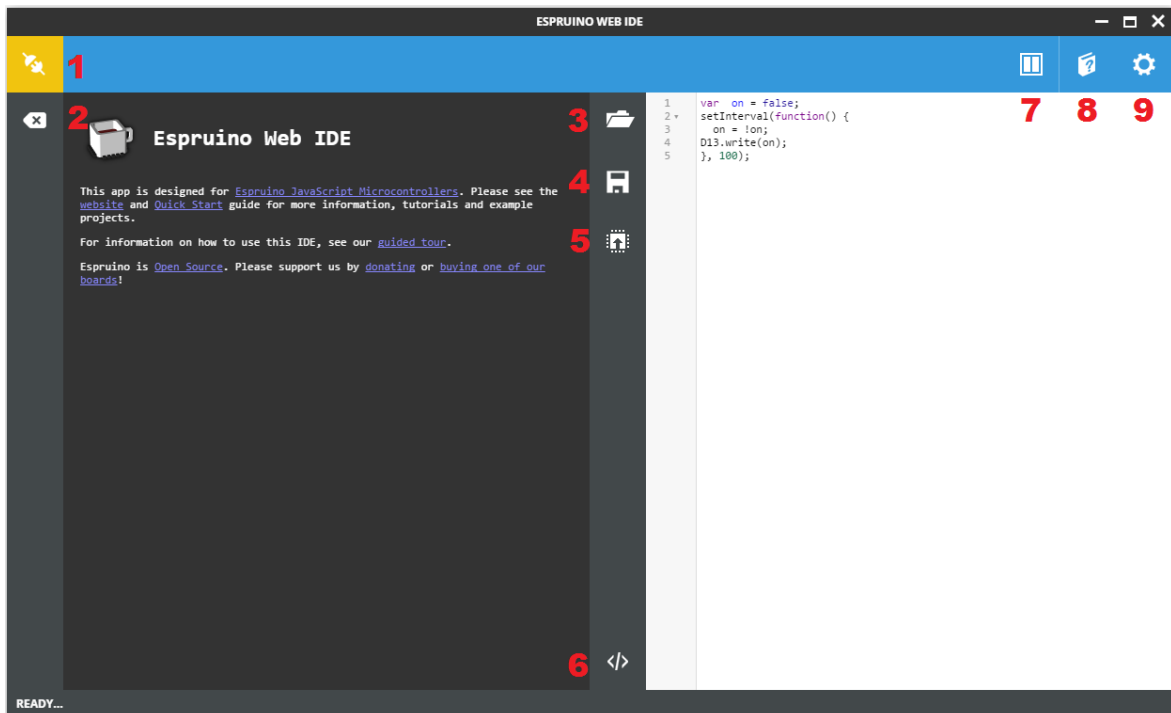
### 4.2 Instalace a popis IDE

Instalace IDE je velmi jednoduchá. Pro instalaci je nutné mít nainstalovaný webový prohlížeč Google Chrome a následně z obchodu Chrome nainstalovat aplikaci Espruino Web IDE. Po instalaci se webové IDE přidá mezi vaše aplikace a je připraveno k použití.

Po spuštění IDE se zobrazí základní obrazovka, na které jsou všechny potřebné ovládací prvky. Obrazovka vývojového prostředí je rozdělena do dvou částí první v levé straně slouží jako konzole a druhá pro psaní kódu programu. Konzoli lze také použít pro přímé zadávání kódu.

#### Popis ovládacích prvků

1. Připojení nebo odpojení centrální jednotky.
2. Vymazání konzole.
3. Otevřít uložený kód.
4. Uložit napsaný kód.
5. Nahrání kódu do centrální jednotky.
6. Přepnutí rozdělení ide horizontální / vertikální.
7. Nápověda
8. Nastavení



Obr. 48 Popis IDE

## 4.3 Programování

Tato kapitola se zabývá vysvětlením základů programování ve vývojovém prostředí a ovládnutí modulů. Ukázky jsou vytvořeny pro oba programovací jazyky podporované prostředím.

### 4.3.1 Výpis do konzole a deklaráce proměnných

Prvním příkladem je jeden z jednodušších příkazů a přitom velice užitečný je výpis do konzole. Do konzole lze vypisovat text tak i přímo hodnoty proměnných.

Výpis do konzole je realizován příkazem `console.log()`. Tento příkaz má jednu nevýhodu, pokud je v programu zavolán příkaz pro výpis do konzole a není připojen USB kabel, program se pozastaví a čeká, dokud není připojen USB kabel a IDE. Další možností výpisu je příkaz `print()`;

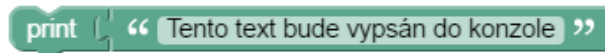
Deklarace proměnné se provádí pomocí příkazu `var`, není nutné zadávat datový typ pouze nadefinovat hodnotu proměnné.

Na příkladu je proveden výpis do konzole oběma možnými způsoby a následně také výpis hodnoty proměnné.

## JavaScript

```
console.log("Tento text bude vypsán do konzole");  
print("Tento text bude vypsán do konzole");  
  
var promena = 123;  
console.log(promena);
```

## Blockly



Obr. 49 Příkaz pro výpis do konzole



Obr. 50 Deklarace proměnné a výpis do konzole

### 4.3.2 Aritmetické operace

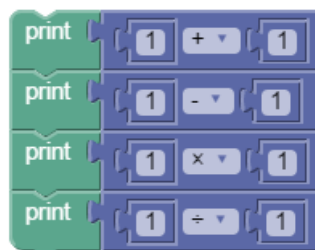
V tomto příkladu jsou zobrazeny základní aritmetické operace, které jsou následně vypsány do konzole. Při použití grafického programovacího jazyka a vnoření více výpočetních bloků, jsou nejdříve vypočítány nejhlouběji vnořené bloky.

V příkladu jsou provedeny jednoduché aritmetické operace a výsledky vypsány do konzole.

## JavaScript

```
console.log(1 + 1);  
console.log(1 - 1);  
console.log(1 * 1);  
console.log(1 / 1);
```

## Blockly



Obr. 51 Aritmetické operace

### 4.3.3 Čtení digitálního vstupu

Čtení digitálních vstupů je realizováno příkazem `digitalRead(pin)`. Pokud nebyl dříve režim pinu nadefinován jako "input", provede se reset a nastaví se do režimu "input". Pokud je jako argument předáváno pole pinů, je návratem číslo, hodnoty jednotlivých pinů je nutné získat převodem do binární hodnoty.

#### JavaScript

```
console.log(digitalRead(BTN1));
```

#### Blockly



Obr. 52 Čtení digitálních vstupů

### 4.3.4 Zápis digitálního výstupu

Zápis digitálních výstupů se provádí pomocí příkazu `digitalWrite(pin, hodnota)`. Pokud není nadefinován režim pinu, provede se automaticky reset a nastavení do výstupního režimu.

#### JavaScript

```
digitalWrite(RED1, true);
```

Druhá možnost

```
RED1.write(true);
```

#### Blockly



Obr. 53 Zápis digitálního výstupu

### 4.3.5 Čtení analogové hodnoty

Analogově digitální převodník se volá příkazem `analogRead(pin)`. Mikropočítač využívá 12 bitový převodník, který rozlišuje 4096 hodnot. Hodnota naměřená převodníkem je dělena rozsahem, aby výsledná hodnota byla v rozsahu 0 až 1.

## JavaScript

```
console.log(analogRead(A6));
```

## Blockly



Obr. 54 Čtení analogové hodnoty

### 4.3.6 Zápis analogové hodnoty – pulzně šířková modulace

Analogový výstup podporuje velmi malé množství desek a proto pokud není dostupný digitálně analogový převodník, je výstup nahrazen PWM signálem, pokud je použita modulace je možné zadat jako jeden z parametrů i frekvenci. Analogový výstup se nastavuje pomocí příkazu `analogWrite(pin, hodnota, {frekvence})`. Hodnotu je nutné zadávat stejně jako při analogovém čtení, tudíž v rozsahu 0 až 1. Hodnota je automaticky vynásobena rozsahem převodníku. Při použití PWM signálu hodnota 1 reprezentuje 100 % střídu signálu.

## JavaScript

```
analogWrite(PWM1, 0.075, { freq: 50 });
```

## Blockly



Obr. 55 Zápis analogové hodnoty - PWM

### 4.3.7 Ovládání tlačítek

Pro obsluhu tlačítek je ideální využít funkce `setWatch(funkce, pin, možnosti)`. Použitím této funkce se nastaví sledování daného pinu a při stisku tlačítka je vyvoláno přerušení, ve kterém je zaznamenán čas vyvolání přerušení a funkce předána do fronty hlavní smyčky programu.

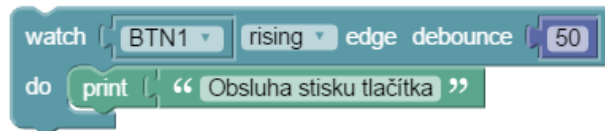
Nevýhodou je, že čipy STM32 nedokáží sledovat dva a více pinů se stejným pořadovým číslem na různých portech, nelze tedy sledovat zároveň např. PA0 a PB0. Maximální počet sledovaných pinů proto odpovídá počtu portů procesoru, tj. v našem případě 16.

V příkladu je nastaveno sledování tlačítka a při jeho stisku je proveden výpis do konzole. Do příkazu jsou přidány dodatečné parametry, jako je ochrana proti zámkitu a výběr, na kterou hranu bude sledovač reagovat.

## JavaScript

```
setWatch(function () {  
    console.log("Obsluha stisku tlačítka");  
}, BTN1, { repeat: true, debounce: 50, edge: "rising" });
```

## Blockly



Obr. 56 Vyvolání přerušení stiskem tlačítka

### 4.3.8 Periodicky spouštěný kód

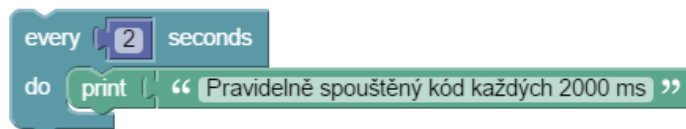
K periodickému spouštění funkce slouží příkaz `setInterval(funkce, čas)`. Tato funkce přebírá dva parametry. První je funkce, která je volána po uplynutí času, který je zadán jako druhý parametr v milisekundách. Při zadání času delšího než 5 s může dojít ke zpoždění až 1 s, to je zapříčiněno přecházením do úsporného režimu v čase nečinnosti.

V příkladu je volána funkce výpisu do konzole každé 2 s.

## JavaScript

```
setInterval(function () { console.log("Pravidelně spouštěný kód každých 2000 ms"); },  
2000);
```

## Blockly



Obr. 57 Pravidelné spouštění kódu

### 4.3.9 Mikrospínač

Tento příklad demonstruje využití mikrospínače, kdy schopnost přepínání spínače je signalizována led diodami. V tomto příkladu jsou využity oba signály dostupné z modulu, ale program lze řešit s pomocí pouze jednoho signálu při sledování náběžné i sestupné hrany.

## JavaScript

```

setWatch(
  function ()
  {
    RED1.write(true); GREEN1.write(false);
  }
  ,C12,{ repeat: true, debounce: 50, edge: "rising" });
setWatch(
  function ()
  {
    RED1.write(false); GREEN1.write(true);
  }
  , D2, { repeat: true, debounce: 50, edge: "rising" });

```

## Blockly



Obr. 58 Použití mikrospínače

### 4.3.10 Senzor čáry a servomotory

Tento příklad popisuje použití senzoru čáry pro řízení dvou servomotorů. Pro rozpoznání z které strany robot překračuje hranici, jsou vytvořeny dvě pomocné proměnné. Senzor čáry posílá dva digitální signály, které vyvolávají přerušení. V obsluze přerušení je ověřeno, ze které strany byla překročena čára a informaci vypíše do konzole. Na základě informace o přechodu čáry je nastavena rychlost servomotoru.

```

var right = true;
var left = true;

analogWrite(PWM15, 0.1, { freq: 50 });
analogWrite(PWM16, 0.1, { freq: 50 });

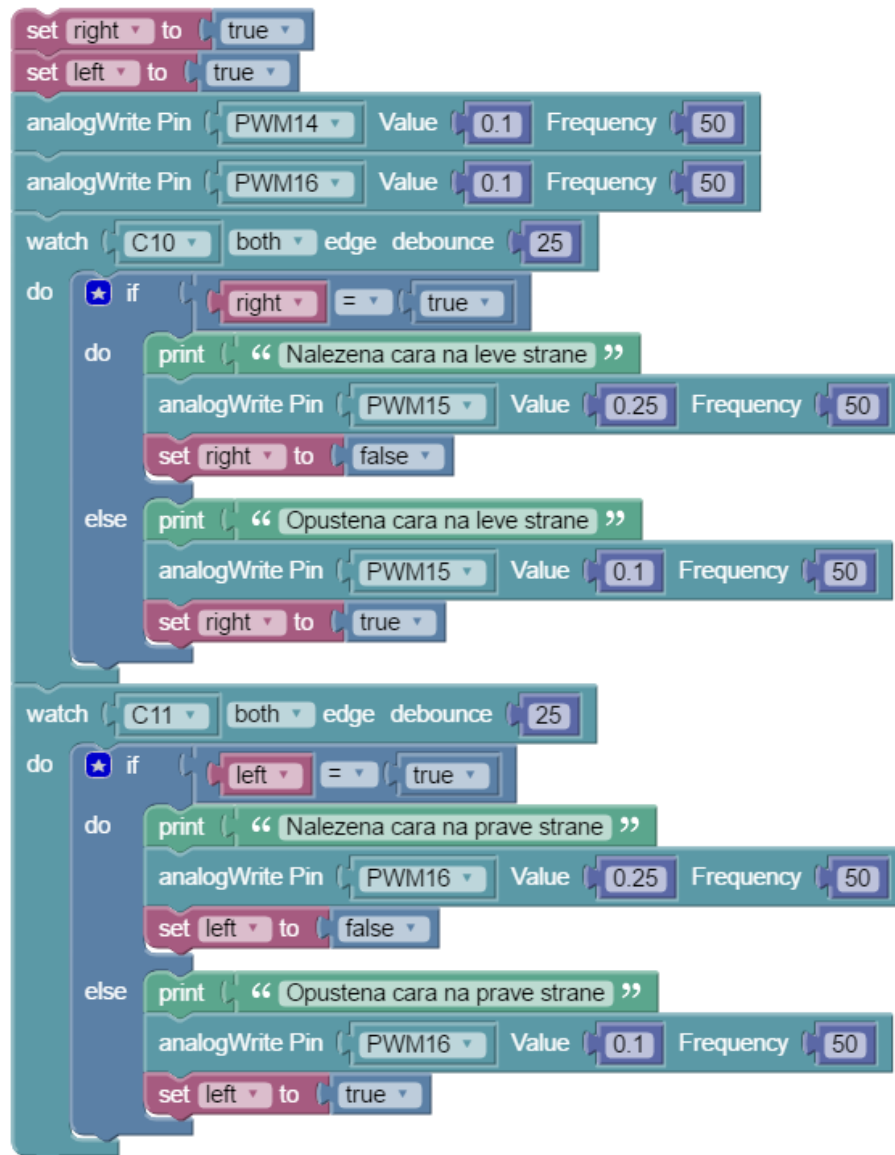
setWatch(function () {

```

```
if (right === true) {
  print("Nalezena cara na prave strane ");
  analogWrite(PWM15, 0.25, { freq: 50 });
  right = false;
}
else {
  print("Opustena cara na prave strane ");
  analogWrite(PWM15, 0.1, { freq: 50 });
  right = true;
}
},C10, { repeat: true, debounce: 50, edge: "both" });

setWatch(function () {
  if (left === true) {
    print("Nalezena cara na leve strane ");
    analogWrite(PWM16, 0.25, { freq: 50 });
    left = false;
  }
  else {
    print("Opustena cara na leve strane ");
    analogWrite(PWM16, 0.1, { freq: 50 });
    left = true;
  }
},C11, { repeat: true, debounce: 50, edge: "both" });
```

## Blockly



Obr. 59 Sledovač čáry

### 4.3.11 Ultrazvukový senzor

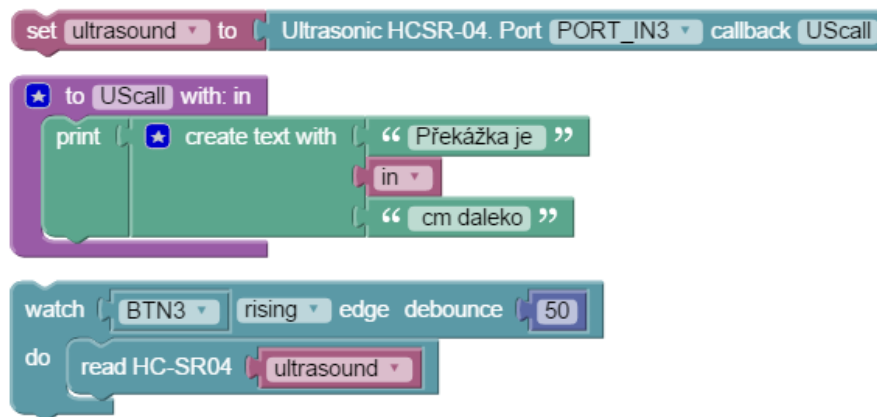
Pro využití ultrazvukového modulu je Espruino již vybaveno, proto je vytvoření programu velmi jednoduché, stačí nadefinovat nový ultrazvukový senzor, kterému jsou nastaveny piny spouště, přijímače a funkce pro zpracování získané hodnoty senzoru.

Spoušť senzoru je nastavena na tlačítko BTN3, při jeho stisku proběhne měření vzdálenosti a následně je vypsána hodnota vzdálenosti do konzole.

## JavaScript

```
var sensor = require("HC-SR04").connect(B11, B10, function (dist) {
  console.log("Překážka je " + dist + " cm daleko");
});
setWatch(function () {
  sensor.trigger();
}, BTN3, { repeat: true, debounce: 50, edge: "rising" });
```

## Blockly



Obr. 60 Ovládání ultrazvukového senzoru

### 4.3.12 Stejnoseměrný motor s enkodérem

Aby bylo možné řídit otáčky stejnosměrného motoru, je nutné na vstup motorového driveru vysílat PWM signál. Pro obousměrné otáčení jsou potřeba dva PWM signály, kdy každý ze signálů reprezentuje opačný směr otáčení. Ovládání rychlosti otáčení je realizováno pomocí potenciometru, jehož hodnota je naměřena a poté přepočítána, aby určovala střihu jednotlivých PWM signálů. Frekvence spínání signálu je nastavena dostatečně vysoko, aby nedocházelo k pískání motoru, které vzniká při použití příliš nízké frekvence.

Informaci o pohybu motoru lze získat pomocí enkodéru. Espruino je vybaveno knihovnou pro kvadratické enkodéry, proto stačí pouze nadefinovat nový enkodér, kterému je nutné nastavit vstupní piny a funkci, ve které je hodnota enkodéru přičítána do proměnné.

Hodnota enkodéru je uložena v proměnné a její hodnota je pro uživatele skrytá, proto je nastaveno sledování tlačítka BTN3 a při jeho stisku je hodnota enkodéru vypsána do konzole.

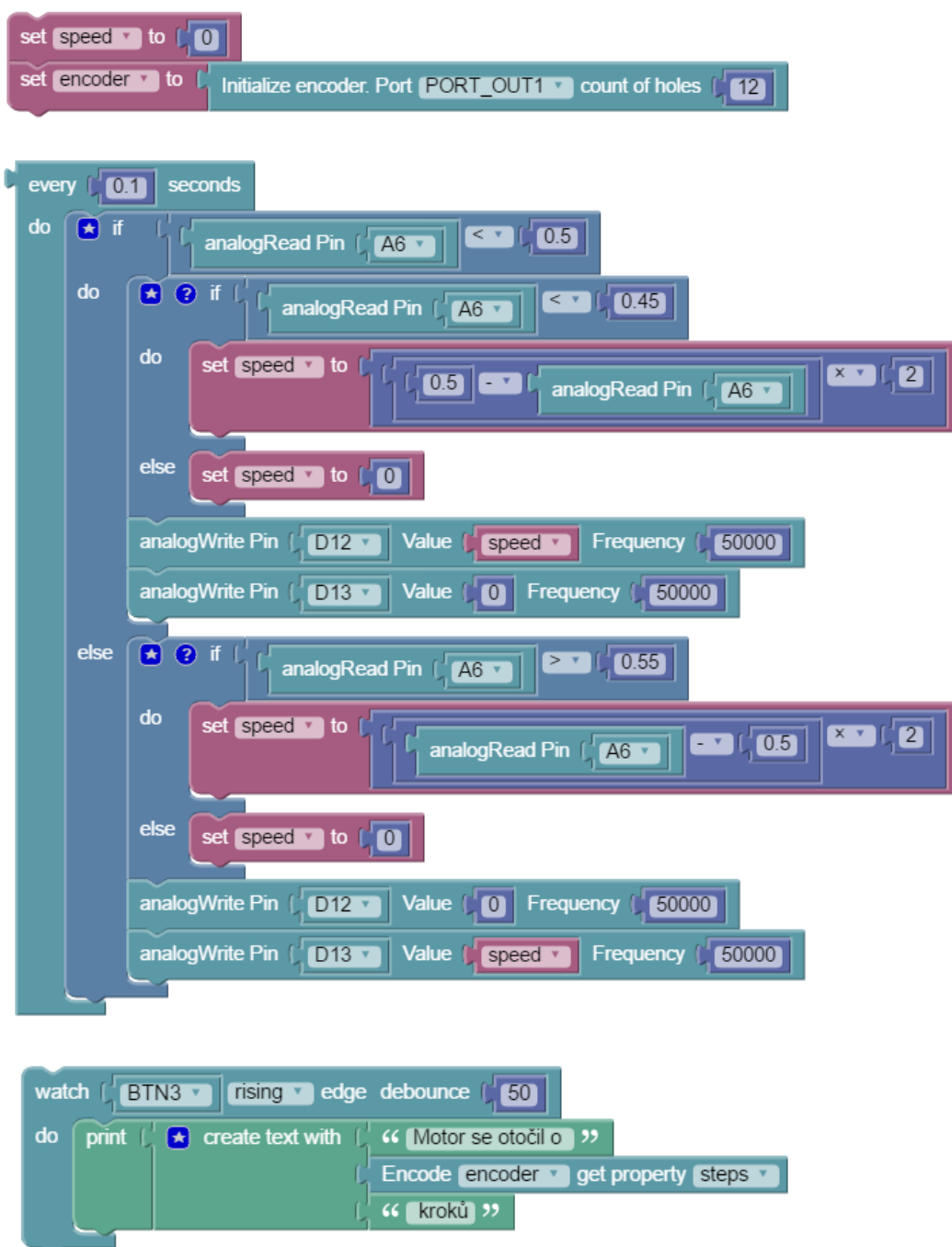
## JavaScript

```
var speed = 0;
var step = 0;
require("Encoder").connect(D15, D14, function (direction) {
    step += direction;
});

setInterval(function ()
    {
        if (analogRead(A6) < 0.5) {
            if (analogRead(A6) < 0.45) {
                speed = (0.5 - analogRead(A6)) * 2;
            }
        }
        else
        {
            speed = 0;
        }
        analogWrite(D12, speed, { freq: 50 });
        analogWrite(D13, 0, { freq: 10000 });
    }
    else
    {
        if (analogRead(A6) > 0.55)
        {
            speed = (analogRead(A6) - 0.5) * 2;
        }
        else
        {
            speed = 0;
        }
        analogWrite(D12, 0, { freq: 10000 });
        analogWrite(D13, speed, { freq: 50 });
    }
}, 100);

setWatch(function () {
    console.log("Motor se otočil o " + step + " pulzů");
}, BTN3, { repeat: true, debounce: 50, edge: "rising" });
```

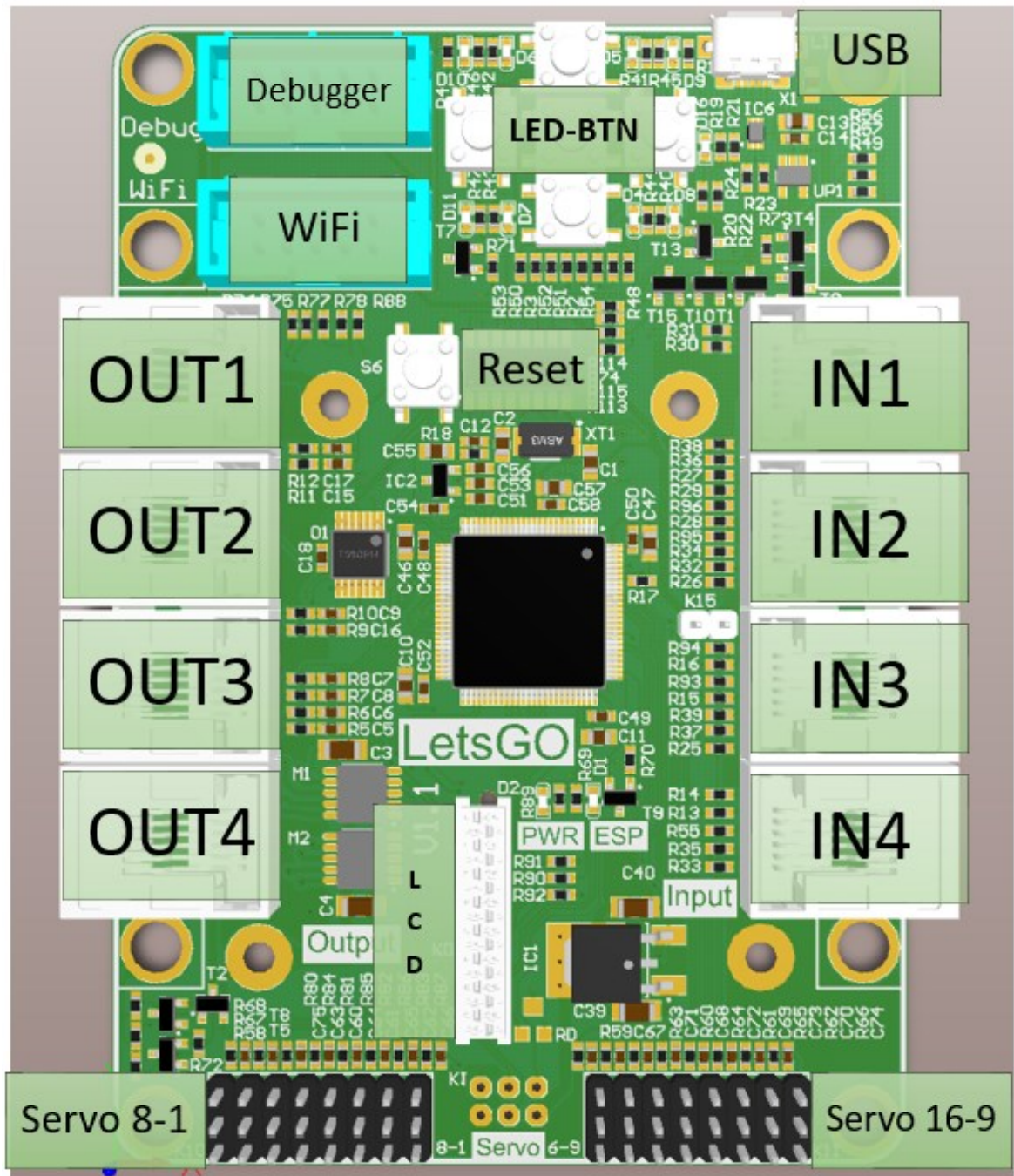
## Blockly



Obr. 61 Ovládání stejnosměrného motoru s enkodérem

## 5 DOKUMENTACE

V dokumentaci jsou popsány a zobrazeny veškeré dostupné prvky centrální jednotky. Jednotlivé konektory jsou popsány v tabulkách s informacemi o jejich zapojení a funkcionalitě. Stejným způsobem jsou také popsány ovládací a signalizační prvky.



Obr. 62 Popis konektorů DPS

## SEZNAM PRVKŮ

USB	Propojení Espruino IDE s PC, podporuje funkci USB OTG.
Debugger	Umožňuje nahrání vlastního firmware a ladění programu.
WiFi	Konektor sloužící k připojení WiFi modulu, podporuje UART a SPI.
RESET	Restart mikropočítače.
IN1-4	Senzorové konektory.
OUT1-4	Výstupní – motorové konektory.
LCD	Konektor pro připojení LCD displeje.
SERVO8-1	Konektor pro řízení servomotorů.
SERVO16-9	Konektor pro řízení servomotorů.
PWR	Signalizační led dioda napájení.
ESP	Signalizační led dioda Epruina.
LED-BTN	Ovládací a signalizační panel.

### 5.1 Senzorové konektory

Pro zapojení senzorových konektorů jsou použity konektory RJ12 s šesti vodiči. Zapojení a funkce jednotlivých konektorů jsou popsány v následující tabulce.

Tab. 1 Zapojení senzorových konektorů

Senzorové konektory											
IN 1			IN 2			IN 3			IN 4		
č.	MCU PIN	Funkce	č.	MCU PIN	Funkce	č.	MCU PIN	Funkce	č.	MCU PIN	Funkce
1	PA7	AD	1	PA6	AD	1	PA5	AD	1	PA4	AD
2	PD6	GPIO	2	PD5	GPIO	2	PD4	GPIO	2	PD3	GPIO
3	-	GND	3	-	GND	3	-	GND	3	-	GND
4	-	5V	4	-	5V	4	-	5V	4	-	5V
5	PC12	GPIO, UART	5	PB6	GPIO, PWM, UART, I2C, CAN	5	PB10	GPIO, PWM, UART, I2C	5	PC10	GPIO, UART
6	PD2	GPIO, UART	6	PB7	GPIO, PWM, UART, I2C, CAN	6	PB11	GPIO, PWM, UART, I2C	6	PC11	GPIO, UART

### 5.2 Výstupní – motorové konektory

Zapojení motorových konektorů využívá konektorů RJ12. Zapojení a funkce jednotlivých pinů konektorů jsou uvedeny v následující tabulce.

Tab. 2 Zapojení výstupních konektorů

Výstupní konektory											
OUT 1			OUT 2			OUT 3			OUT 4		
č.	MCU PIN	Funkce	č.	MCU PIN	Funkce	č.	MCU PIN	Funkce	č.	MCU PIN	Funkce
1	PD12	DOUT,PWM	1	PB15	DOUT,PWM	1	PE14	DOUT,PWM	1	PE11	DOUT,PWM
2	PD13	DOUT,PWM	2	PB14	DOUT,PWM	2	PE13	DOUT,PWM	2	PE9	DOUT,PWM
3	-	GND	3	-	GND	3	-	GND	3	-	GND
4	-	5V	4	-	5V	4	-	5V	4	-	5V
5	PD15	GPIO, PWM	5	PC9	GPIO, PWM	5	PA3	GPIO, PWM, UART	5	PA1	GPIO, PWM, UART
6	PD14	GPIO, PWM	6	PC8	GPIO, PWM	6	PA2	GPIO, PWM, UART	6	PA0	GPIO, PWM, UART

Vodiče 1 a 2 motorových konektorů jsou řízeny pomocí motorových driverů, které spínají výstupní napětí do konektoru. V následující tabulce je popsáno propojení driverů s konektory a mikropočítačem.

Tab. 3 Tabulka zapojení motorových driverů

Driver M1			Driver M2		
Konektor	Driver Pin	MCU Pin	Konektor	Driver Pin	MCU Pin
OUT4-2	IN1	PE9	OUT2-2	IN1	PB14
OUT4-1	IN2	PE11	OUT2-1	IN2	PB15
OUT3-2	IN3	PE13	OUT1-2	IN3	PD13
OUT3-1	IN4	PE14	OUT1-1	IN4	PD12

Tabulka 4 popisuje vnitřní logiku motorového driveru.

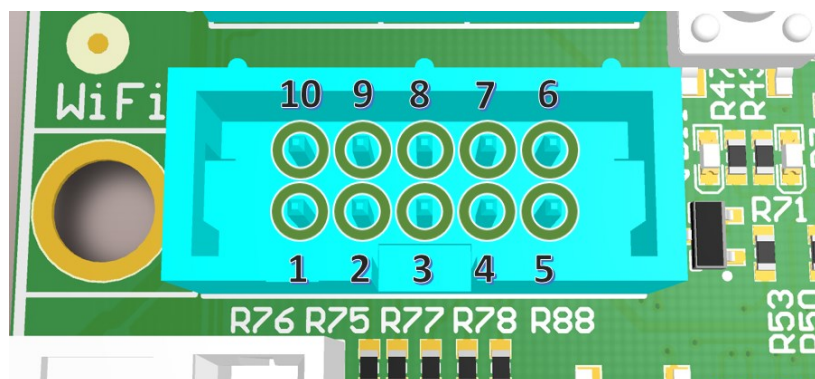
Tab. 4 Tabulka řízení motorového driveru

Input				Output				Remarks	
IN1	IN2	IN3	IN4	OUT1	OUT2	OUT3	OUT4		
L	L	L	L	OFF	OFF	OFF	OFF	Stand-by	
L	L			OFF	OFF			1CH	
H	L			H	L				Forward
L	H			L	H				Reverse
H	H			L	L				Brake
		L	L			OFF	OFF	2CH	
		H	L			H	L		Forward
		L	H			L	H		Reverse
		H	H			L	L		Brake

### 5.3 Wifi konektor

Na obrázku je zobrazen konektor WiFi modulu a číslování jeho vodičů. Tento konektor není striktně určen pro WiFi modul a může být použit pro zapojení dalších modulů, které jsou

kompatibilní se zapojením konektoru. Konektor je vybaven zámkem, aby nedocházelo k obrácení jeho polarity.



Obr. 63 Zapojení WiFi konektoru

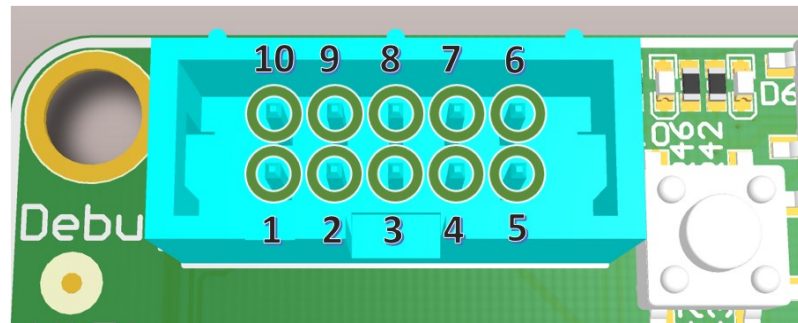
V tabulce je popsáno zapojení jednotlivých vodičů k mikro počítači a jejich funkcionalita.

Tab. 5 Zapojení WiFi konektoru

WIFI		
č.	MCU PIN	Funkce
1	-	3V3
2	PC6	GPIO, PWM, UART
3	-	3V3
4	PA7	GPIO, PWM, SPI1
5	PA5	GPIO, PWM, SPI1
6	PA4	GPIO, SPI1
7	PA6	GPIO, PWM, SPI1
8	-	GND
9	PC7	GPIO, PWM, UART
10	-	GND

## 5.4 Debugger / ST-Link

Jedná se o konektor, který je používán pouze pro nahrávání firmware a ladění vlastních programů. Na obrázku je uvedeno číslování jednotlivých vodičů konektoru. Konektor je vybaven zámkem, aby nedocházelo k obrácení jeho polarity.



Obr. 64 Zapojení debuggeru

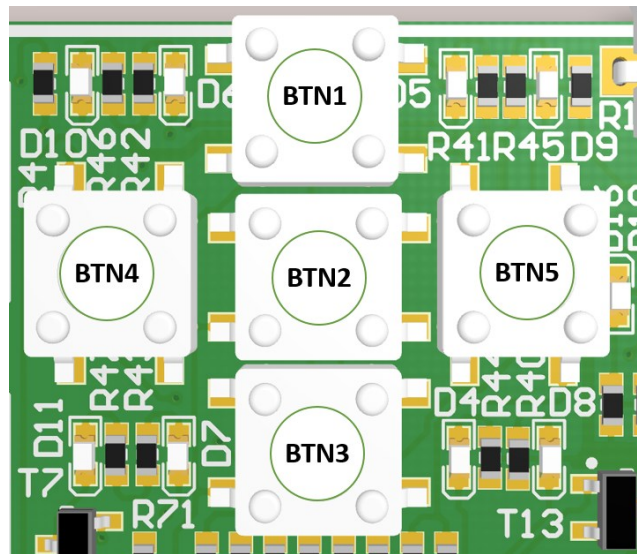
Zapojení konektoru je popsáno v následující tabulce.

Tab. 6 Zapojení debuggeru

DEBUGGER		
č.	MCU PIN	Funkce
1	-	3V3
2	-	GND
3	-	GND
4	-	
5	GND	-
6	NRST	Reset
7	-	-
8	-	-
9	PA14	SWCLK
10	PA13	SWDIO

## 5.5 Ovládací a signalizační panel

Panel se skládá z pěti tlačítek, která jsou připojena k mikropočítači. Značení jednotlivých tlačítek je zobrazeno na následujícím obrázku. V rozích panelu jsou umístěny signalizační led diody. V každém rohu se nachází jedna zelená a jedna červená led dioda.



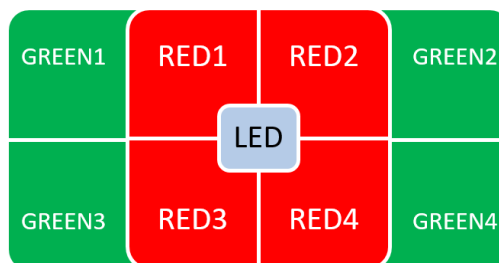
Obr. 65 Zapojení tlačítkového panelu

Tabulka popisuje propojení tlačítek s piny mikropočítače.

Tab. 7 Zapojení tlačítkového panelu

Tlačítkový panel		
č.	Tlačítko	MCU PIN
1	BTN1	PE10
2	BTN2	PC5
3	BTN3	PC4
4	BTN4	PC3
5	BTN5	PC2

Na obrázku je graficky znázorněno rozložení a značení jednotlivých led diod umístěných v panelu.



Obr. 66 Rozložení led panelu na DPS

Tabulka popisuje propojení jednotlivých led diod s mikropočítačem.

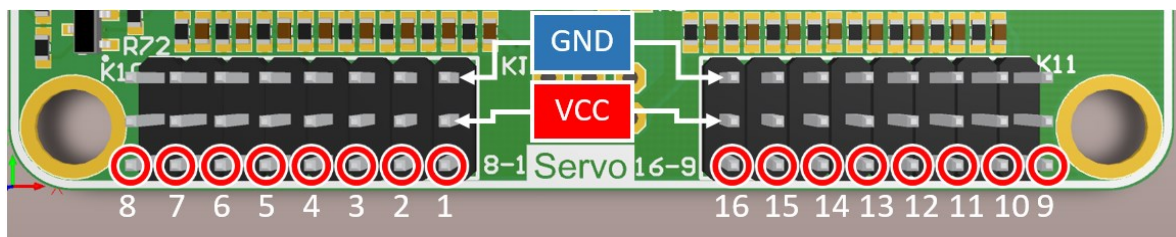
Tab. 8 Zapojení led panelu

Led panel		
č.	Dioda	MCU PIN
Zelené		
1	GREEN1	PA8
2	GREEN2	PC13
3	GREEN3	PB13
4	GREEN4	PB12
Červené		
5	RED1	PD11
6	RED2	PD10
7	RED3	PD9
8	RED4	PD8

## 5.6 Servomotorové konektory

Na obrázku jsou konektory servomotorů. Pro připojení jsou použity pinové lišty 3 x 8.

Horní řada pinové lišty je připojena k zemnicímu vodiči. Prostřední řada slouží k napájení a v poslední řadě jsou signální vodiče připojené do mikropočítače.



Obr. 67 Zapojení servomotorových konektorů

Tabulka popisuje propojení signálních vodičů servomotorových konektorů s mikropočítačem a funkce, které lze využít.

Tab. 9 Zapojení servomotorových konektorů

Servomotorové konektory								
Pin 1-8								
č	1	2	3	4	5	6	7	8
MCU Pin	PE9	PE11	PE13	PE14	PB14	PB15	PD13	PD12
Funkce	GPIO, PWM	GPIO, PWM	GPIO, PWM	GPIO, PWM	GPIO, PWM	GPIO, PWM	GPIO, PWM	GPIO, PWM
Pin 9-16								
č	9	10	11	12	13	14	15	16
MCU Pin	PE5	PE6	PB0	PB1	PB4	PB5	PB8	PB9
Funkce	GPIO, PWM	GPIO, PWM	GPIO, PWM	GPIO, PWM	GPIO, PWM	GPIO, PWM	GPIO, PWM, I2C	GPIO, PWM, I2C

## ZÁVĚR

Cílem práce bylo analyzovat současné robotické systémy dostupné na trhu. Zjistit jejich výhody, nevýhody a na základě získaných informací navrhnout vlastní robotický systém, který je vhodný pro výuku robotiky.

Při návrhu systému byl kladen důraz na možnosti programování. Hlavním požadavkem pro programovací jazyky byla možnost programovat pomocí grafického editoru. Z tohoto důvodu bylo zvoleno vývojové prostředí Espruino, které umožňuje programování v javascriptu a Blockly. Mikro počítač pro centrální jednotku byl zvolen od výrobce STMicroelectronics, konkrétně model STM32F407VET, který s frekvencí 168 MHz poskytuje dostatečný výpočetní výkon pro řešení robotických problémů. Tento mikro počítač má také bohatou výbavu periférií umožňujících řízení široké škály sensorových a akčních modulů. Mimo prostředí Espruino lze také nahrávat vlastní alternativní firmware pomocí komunikace SWD a tím umožnit programování a ladění s využitím dalších programovacích jazyků.

Hardware systému byl navržen pomocí programu Altium Designer, který umožňuje kompletní návrh desek plošných spojů. Včetně centrální jednotky byly také navrženy základní sensorové a akční moduly systému. Desky byly navrženy jako dvouvrstvý plošný spoj se součástkami osazenými na horní straně, což umožnilo jednodušší strojní osazování desek. Pouze u senzoru čáry bylo nutné Vysílače a přijímače umístit na spodní stranu desky.

Navržené desky plošných spojů byly vyrobeny společností Gatema a.s. a následně osazeny součástkami. Po úspěšné kompletaci desek byla centrální jednotka společně s moduly testována. Na základě informací získaných z testování byl návrh centrální jednotky vylepšen. Nyní je navržena třetí verze centrální jednotky. Upravené desky plošných spojů byly panelizovány pro výrobu ve větším počtu kusů.

Další část práce se zabývala vysvětlením práce s vývojovým prostředím Espruino a základům ovládání tohoto prostředí. Byla vysvětlena funkčnost jednotlivých systémových modulů a možnosti jejich ovládání v obou dostupných programovacích jazycích javascript a Blockly. Součástí jsou také vzorové příklady ovládání jednotlivých modulů modulů.

Aby bylo možné systém používat ve výuce, byla vytvořena dokumentace, která popisuje zapojení jednotlivých konektorů a veškeré možnosti jejich využití pro řízení modulů.

System je navržen tak, aby bylo možné jednoduše vyvíjet nové rozšiřující moduly nebo využít moduly například ze stavebnic Lego Mindstorms. Tudiž může být systém využíván pro širokou škálu studentů s různou úrovní zkušeností.

**SEZNAM POUŽITÉ LITERATURY**

- [1] PINKER, Jiří. Mikroprocesory a mikropočítače. Praha: BEN - technická literatura, 2004. ISBN 80-730-0110-1.
- [2] ZÁHLAVA, Vít. Návrh a konstrukce desek plošných spojů: principy a pravidla praktického návrhu. 1. vyd. Praha: BEN - technická literatura, 2010, 123 s. ISBN 978-80-7300-266-4.
- [3] PARK, Eun Jung. Lego Mindstorms EV3: stavíme a programujeme roboty. Brno: Computer Press, 2015. ISBN 978-80-251-4385-8.
- [4] CATSOULIS, John. Designing embedded hardware. 2nd ed. Sebastopol, CA: O'Reilly, c2005. ISBN 0596007558.
- [5] VÁŇA, Vladimír. ARM pro začátečníky. Praha: BEN - technická literatura, 2009. ISBN 978-80-7300-246-6
- [6] ÚREDNÍČEK, Zdenek. Robotika. Zlín, 2012. ISBN 978-80-7454-223-7.
- [7] YIU, Joseph. The definitive guide to ARM® Cortex-M3 and Cortex-M4 processors. Third edition. ISBN 0124080820.
- [8] PEHLIVANIAN, Ara a Don NGUYEN. JavaScript okamžitě. Brno: Computer Press, 2014. ISBN 978-80-251-4163-2.
- [9] About. Scratch [online]. [cit. 2017-04-28]. Dostupné z: <https://scratch.mit.edu/about>.
- [10] Blockly. Google developers [online]. [cit. 2017-03-28]. Dostupné z: <https://developers.google.com/blockly/>
- [11] Lego Mindstorms NXT. In: Wikipedia: the free encyclopedia [online]. San Francisco(CA): Wikimedia Foundation, 2016 [cit. 2017-04-06]. Dostupné z: [https://en.wikipedia.org/wiki/Lego\\_Mindstorms\\_NXT](https://en.wikipedia.org/wiki/Lego_Mindstorms_NXT)
- [12] Lego Mindstorms EV3. In: Wikipedia: the free encyclopedia [online]. San Francisco(CA): Wikimedia Foundation, 2016 [cit. 2017-04-06]. Dostupné z: [https://en.wikipedia.org/wiki/Lego\\_Mindstorms\\_EV3](https://en.wikipedia.org/wiki/Lego_Mindstorms_EV3)
- [13] Lego Mindstorms. In: Wikia: the free encyclopedia [online]. [cit. 2017-04-06]. Dostupné z: <http://bzpowercircle.wikia.com/wiki/LEGO>
- [14] VODA, Zbyšek. Průvoce světem Aarduina. *ARDUINO.CZ* [online]. , 280 [cit. 2017-05-15]. Dostupné z: <https://arduino.cz/>

- [15] Javascript for Microcontrollers. Espruino [online]. [cit. 2017-05-15]. Dostupné z: <http://www.espruino.com>
- [16] Makeblock. Makeblock [online]. [cit. 2017-05-15]. Dostupné z: <http://www.makeblock.com/>
- [17] El. motory a jejich řízení. Automatizace.hw.cz [online]. [cit. 2017-05-15]. Dostupné z: <http://automatizace.hw.cz>
- [18] Řízení modelářského servomotoru. ZAVAVOV [online]. [cit. 2017-05-15]. Dostupné z: <http://zavavov.vzap.eu/cz/>
- [19] SELVARAJ, Ganesh. Servo Motor Control Using 555 Timer IC. In: EngineersGarage [online]. [cit. 2017-05-15]. Dostupné z: <https://www.engineersgarage.com/contribution/expert/servo-motor-control-using-555-timer-ic>
- [20] NECKÁŘ, Pavel. Ultrazvuk a senzory SRFO2. In: *Posterus.sk* [online]. [cit. 2017-05-15]. Dostupné z: <http://www.posterus.sk/?p=10661>
- [21] NECKÁŘ, Antonín Vojáček. Rotační enkodéry pro řízení pohonů a pohybu. In: *automatizace.hw* [online]. 2017 [cit. 2017-05-15]. Dostupné z: <http://automatizace.hw.cz/rotacni-ekodery-pro-rizeni-pohonu-a-pohybu.html>
- [22] LEGO Mindstorms EV3. In: *GamingGadgets* [online]. 2013 [cit. 2017-05-16]. Dostupné z: <http://gaminggadgets.de/lego>
- [23] Potenciometr. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-05-16]. Dostupné z: <https://cs.wikipedia.org/wiki/Potenciometr>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

IR	Infračervené záření
DPS	Deska plošných spojů
DC	Stejnoseměrný proud
GPIO	Všestranný vstup / výstup
IDE	Integrované vývojové prostředí
RAM	Paměť s libovolným (náhodným) výběrem (přístupem), neboli vyrovnávací paměť počítače.
PWM	Pulzně šířková modulace
MCU	Mikro počítač
FLASH	Elektricky programovatelná paměť typu RAM.
LCD	Displej s kapalnými krystaly
LED	Světlo emitující dioda
NiMH	Nikl-metal hydridový akumulátor
Li-pol	Lithium-polymerový akumulátor
ZŠ	Základní škola

**SEZNAM OBRÁZKŮ**

Obr. 1 Prostředí LabVIEW .....	13
Obr. 2 Lego RCX [13] .....	14
Obr. 3 Lego NXT [11] .....	15
Obr. 4 Lego EV3 [22] .....	16
Obr. 5 Arduino IDE .....	17
Obr. 6 Arduino Nano [14].....	18
Obr. 7 Arduino Uno [14] .....	19
Obr. 8 Arduino Robot [14] .....	20
Obr. 9 Espruino web IDE .....	21
Obr. 10 Espruino Board [15] .....	22
Obr. 11 mBlock IDE.....	24
Obr. 12 Makeblock Orion [16] .....	25
Obr. 14 Ovládání servomotoru [19].....	28
Obr. 15 Funkce enkodéru [21].....	31
Obr. 16 Vyhodnocení signálu enkodéru [21].....	31
Obr. 17 Blokové zapojení .....	34
Obr. 18 Zapojení MOSFET tranzistoru .....	35
Obr. 19 Simulace zapojení MOSFETU .....	35
Obr. 20 LDO regulátor centrální jednotky.....	35
Obr. 21 Schéma mikropočítače.....	36
Obr. 22 Schéma zapojení led panelu .....	37
Obr. 23 Schéma zapojení signalizace .....	37
Obr. 24 Schéma zapojení tlačítkového panelu.....	38
Obr. 25 Schéma zapojení USB .....	39
Obr. 26 Schéma senzorových konektroů .....	39
Obr. 27 Schéma napěťový dělič a sledovač.....	40
Obr. 28 Schéma zapojení výstupních konektorů .....	40
Obr. 29 Schéma driveru .....	41
Obr. 30 Schéma servomotorového konektoru .....	41
Obr. 31 Schéma wifi konektoru.....	42
Obr. 32 Schéma zapojení konektoru pro displej.....	42
Obr. 33 Schéma zapojení konektoru debuggeru.....	43

Obr. 34 DPS centrální jednotky .....	44
Obr. 35 Schéma bateriového modulu .....	45
Obr. 36 DPS bateriového modulu .....	46
Obr. 37 Schéma mikrospínače .....	46
Obr. 38 DPS mikrospínače .....	47
Obr. 39 Schéma modulu potenciometru .....	47
Obr. 40 DPS potenciometru .....	48
Obr. 41 Schéma redukce pro ultrazvukový modul .....	48
Obr. 42 DPS ultrazvukového senzoru .....	49
Obr. 43 Schéma zapojení bzučáku .....	50
Obr. 44 DPS Bzučáku .....	50
Obr. 45 Schéma zapojení IR senzoru .....	51
Obr. 46 DPS IR senzoru .....	52
Obr. 47 Schéma zapojení DC motoru s enkodérem .....	52
Obr. 48 DPS DC motoru .....	53
Obr. 49 Popis IDE .....	55
Obr. 50 Příkaz pro výpis do konzole .....	56
Obr. 51 Deklarace proměnné a výpis do konzole .....	56
Obr. 52 Aritmetické operace .....	56
Obr. 53 Čtení digitálních vstupů .....	57
Obr. 54 Zápis digitálního výstupu .....	57
Obr. 55 Čtení analogové hodnoty .....	58
Obr. 56 Zápis analogové hodnoty - PWM .....	58
Obr. 57 Vyvolání přerušování stiskem tlačítka .....	59
Obr. 58 Pravidelné spouštění kódu .....	59
Obr. 59 Použití mikrospínače .....	60
Obr. 60 Sledovač čáry .....	62
Obr. 61 Ovládání ultrazvukového senzoru .....	63
Obr. 62 Ovládání stejnosměrného motoru s enkodérem .....	65
Obr. 63 Popis konektorů DPS .....	66
Obr. 64 Zapojení WiFi konektoru .....	69
Obr. 65 Zapojení debuggeru .....	70
Obr. 66 Zapojení tlačítkového panelu .....	71

---

Obr. 67 Rozložení led panelu na DPS .....	71
Obr. 68 Zapojení servomotorových konektorů.....	72

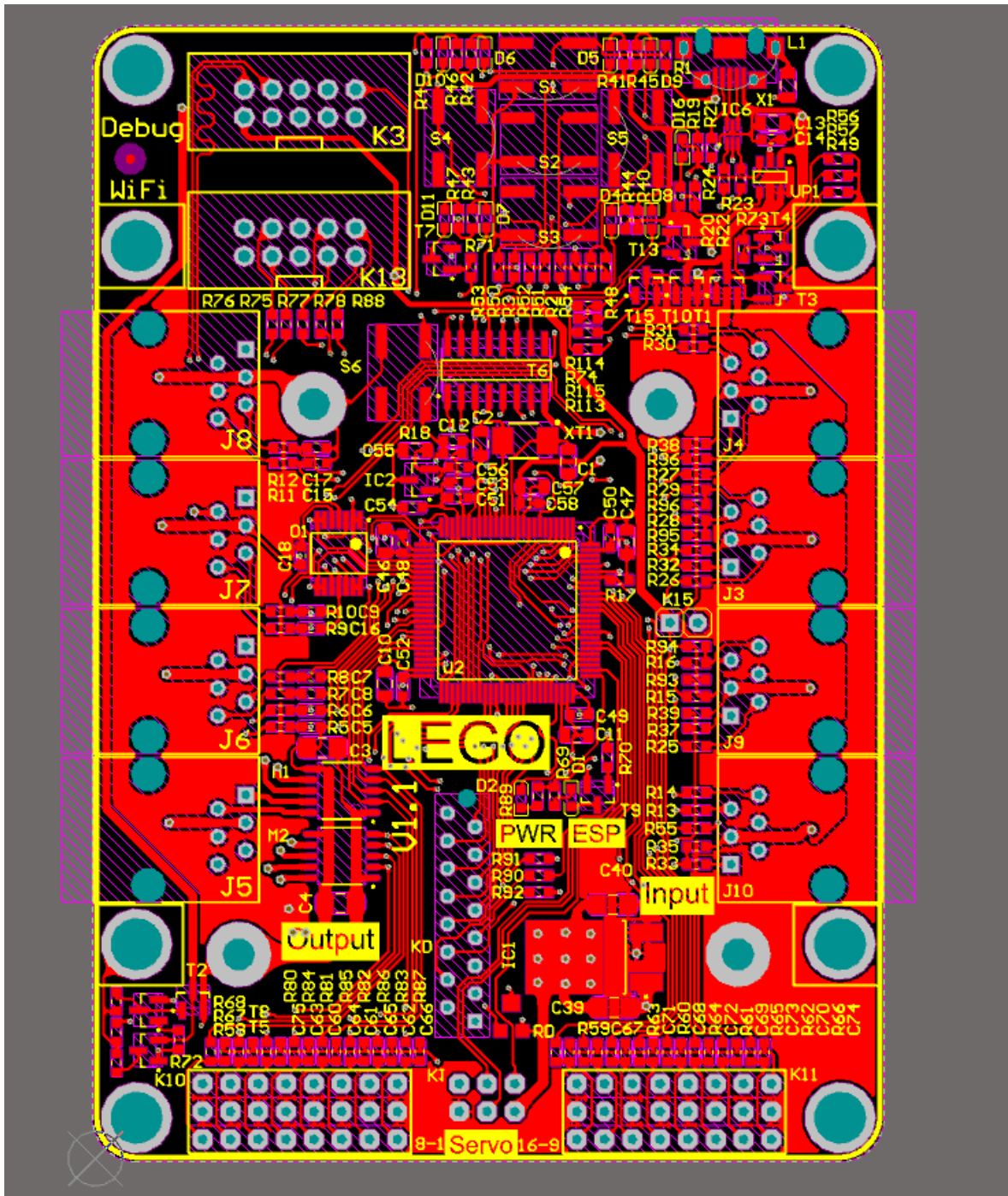
**SEZNAM TABULEK**

Tab. 1 Zapojení sensorových konektorů.....	67
Tab. 2 Zapojení výstupních konektorů .....	68
Tab. 3 Tabulka zapojení motorových driverů.....	68
Tab. 4 Tabulka řízení motorového driveru .....	68
Tab. 5 Zapojení WiFi konektoru.....	69
Tab. 6 Zapojení debuggeru .....	70
Tab. 7 Zapojení tlačítkového panelu.....	71
Tab. 8 Zapojení led panelu .....	72
Tab. 9 Zapojení servomotorových konektorů.....	72

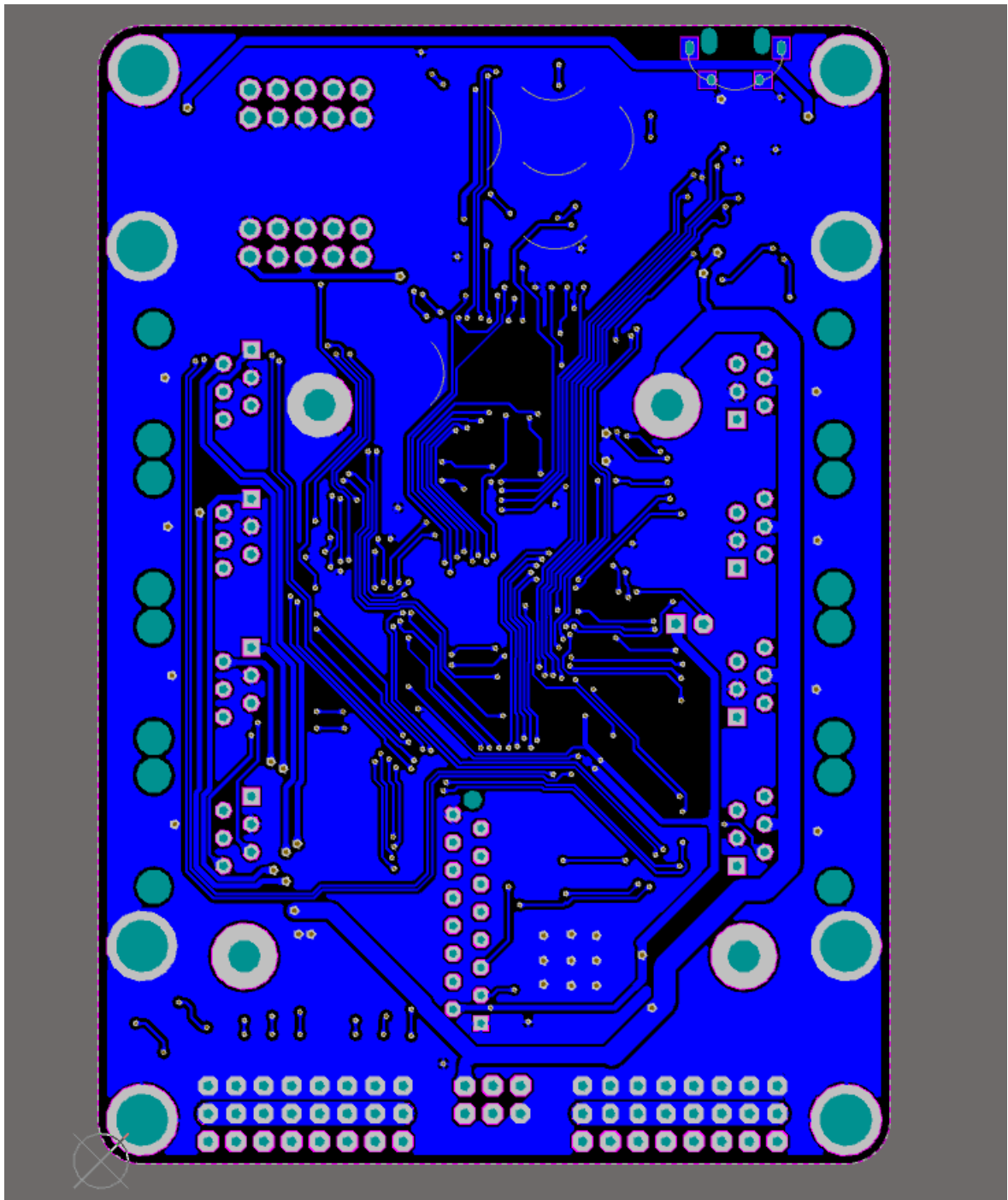
## SEZNAM PŘÍLOH

- PI DPS centrální jednotky horní strana
- PII DPS centrální jednotky spodní strana
- PIII Schéma Degugger
- PIV Schéma nabíječky NiMH
- PV Projekt centrální jednotky
- PVI Projekt bateriového modulu
- PVII Projekt mikrospínače
- PVIII Projekt potenciometru
- PIX Projekt ultrazvukového senzoru
- PX Projekt senzoru čáry
- PXI Projekt DC motoru
- PXII Projekt bzučáku

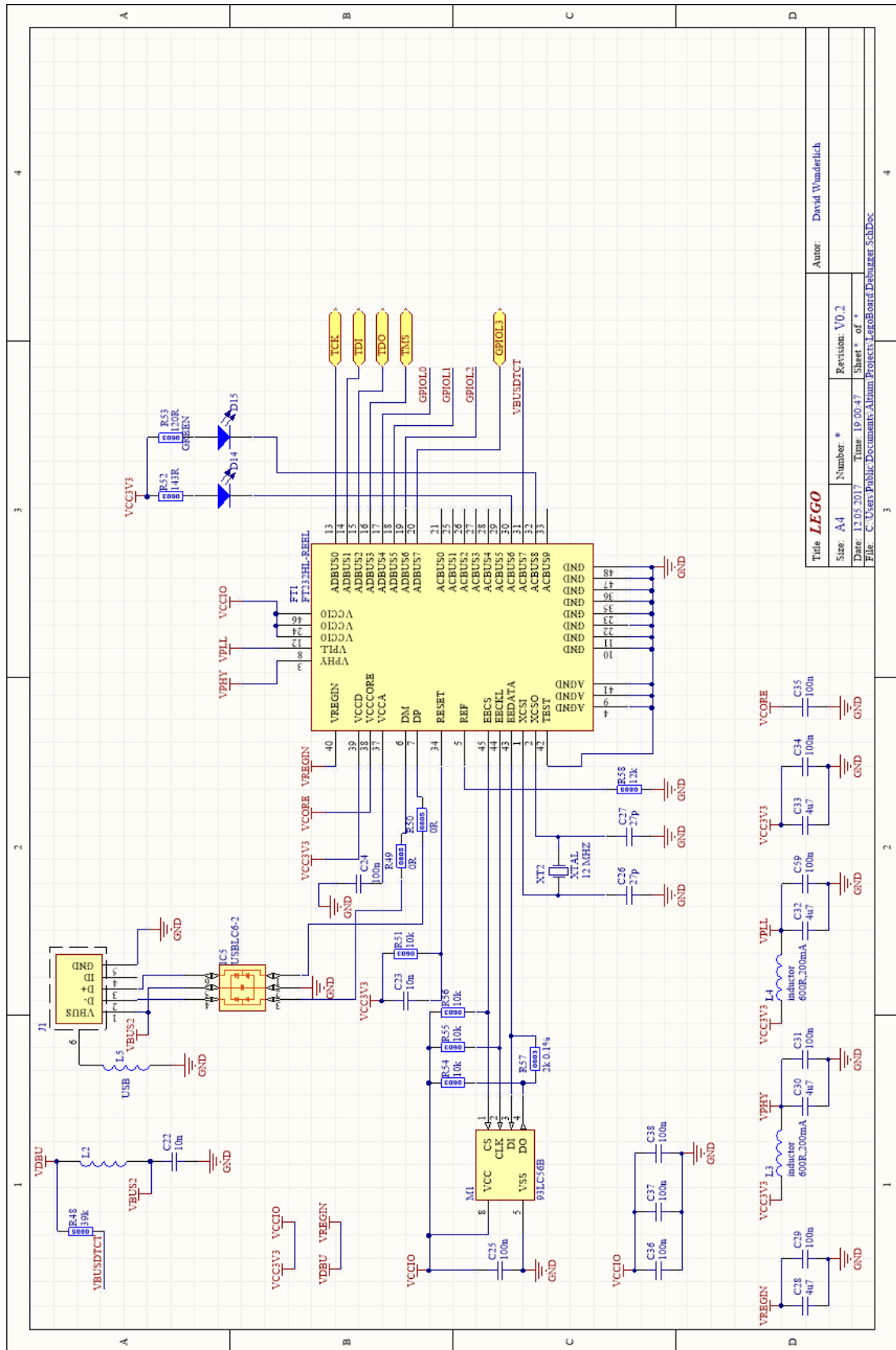
# PŘÍLOHA P I: DPS CENTRÁLNÍ JEDNOTKY HORNÍ STRANA



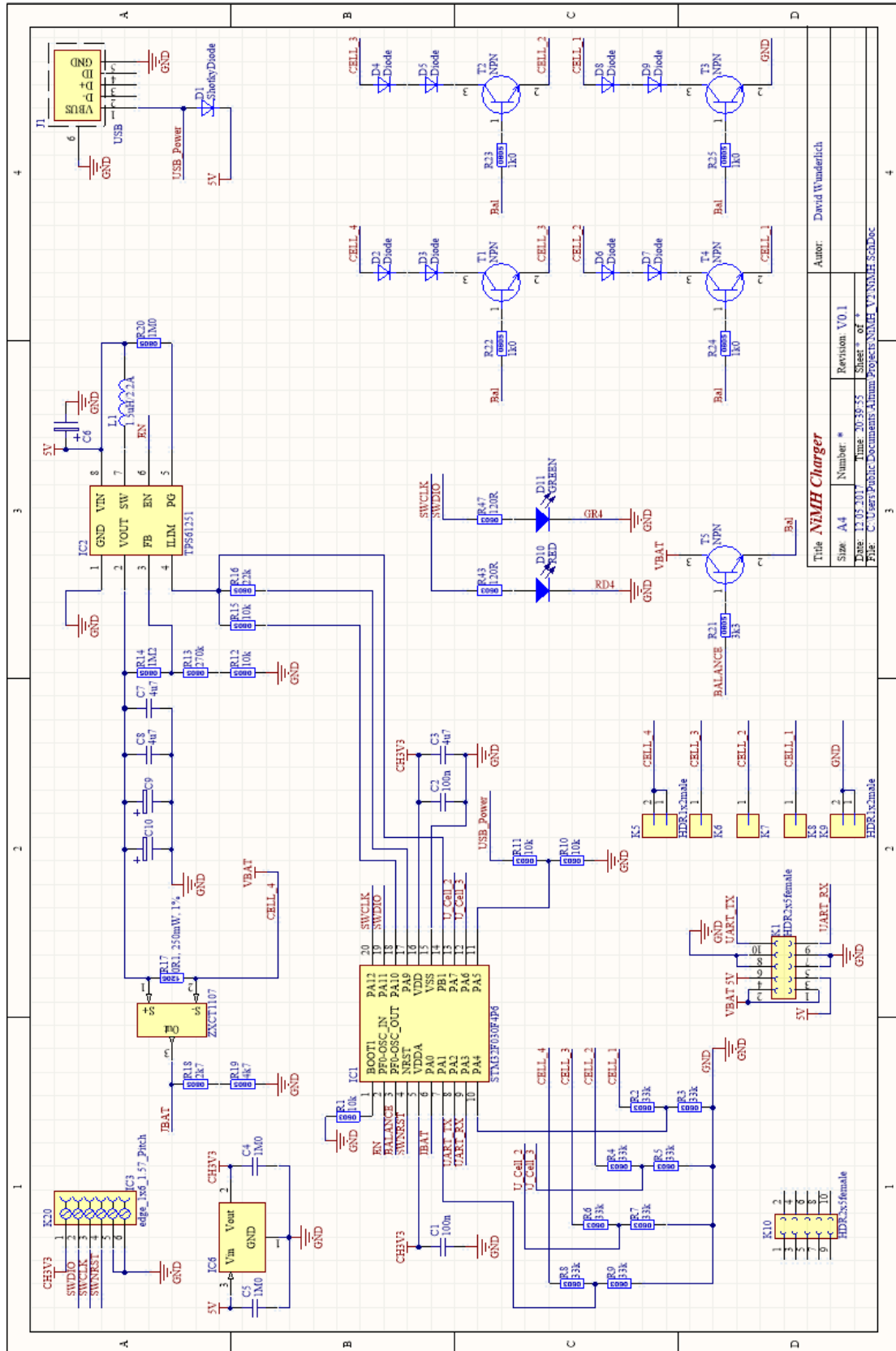
## PŘÍLOHA P II: DPS CENTRÁLNÍ JEDNOTKY SPODNÍ STRANA



# PŘÍLOHA P III: SCHÉMA DEGUGGER



# PŘÍLOHA P IV: SCHÉMA NABÍJEČKY NIMH



Title: <b>NiMH Charger</b>		Autor: David Wunderlich	
Size: A4	Number: *	Revision: V0.1	
Date: 13.05.2017	Time: 20:38:55	Sheet: * of *	
File: C:\Users\Public\Documents\Altimu\Projects\NiMH_V01\NiMH_SchDoc			