ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ

KATEDRA ELEKTROMECHANIKY A VÝKONOVÉ ELEKTRONIKY

BAKALÁŘSKÁ PRÁCE

Ovládání krokového motoru v modelu závodů automobilů pomocí mikrokontroleru Arduino

Václav Frélich **2016**

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: **2015/2016**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Václav FRÉLICH**
Osobní číslo:	**E12B0018P**
Studijní program:	**B2612 Elektrotechnika a informatika**
Studijní obor:	**Elektrotechnika a energetika**
Název tématu:	**Ovládání krokového motoru v modelu závodů automobilů pomocí mikrokontroleru Arduino**
Zadávající katedra:	**Katedra elektromechaniky a výkonové elektroniky**

Zásady pro vypracování:

1. Popište mikrokontroler Arduino a jeho vývojové prostředí.
2. Popište možnosti řízení krokových motorů.
3. Navrhněte a vytvořte ovládání krokových motorů v modelu.
4. Navrhněte možná další vylepšení modelu.

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah kvalifikační práce: **30 - 40 stran**

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

Seznam odborné literatury:

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

Vedoucí bakalářské práce: **Ing. Pavel Světlík**
Katedra elektromechaniky a výkonové elektroniky

Datum zadání bakalářské práce: **15. října 2015**

Termín odevzdání bakalářské práce: **2. června 2016**

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

L.S.

Prof. Ing. Václav Kůs, CSc.
vedoucí katedry

V Plzni dne 15. října 2015

Abstrakt

Předkládaná bakalářská práce je zaměřena na vylepšení stávajícího modelu závodů automobilů. Práce se zabývá mikrokontrolerem Arduino UNO a základními způsoby řízení krokových motorů. Součástí práce je navržený a vyrobený plošný spoj, který pomocí vytvořeného programu pro Arduino UNO obsluhuje krokový motor. Závěrem jsou navrhnuty další možnosti vylepšení závodů automobilů.

Klíčová slova

Arduino UNO, Arduino IDE, krokový motor, digitální piny, displej, model závodu automobilů, statorové cívky, plošný spoj

Abstract

This bachelor thesis is focused on improving the existing model of racing cars. The thesis deals with microcontroler Arduino UNO and the basic methods of controlling stepper motors. Part of the work is designed and manufactured printed circuit board. This printed circuit board and created program for Arduino UNO controlling a stepper motor. Finally are some other opportunities to improve model of racing cars.

Key words

Arduino UNO, Arduino IDE, stepper motor, digital pins, display, model of racing cars, stator coils, printed circuit

Prohlášení

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

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

.....

podpis

V Plzni dne 30.5.2016 Václav Frélich

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce panu Ing. Pavlu Světlíkovi za připomínky a cenné profesionální rady při řešení této práce.

Obsah

OBSAH **8**

ÚVOD **10**

SEZNAM SYMBOLŮ A ZKRATEK **11**

ARDUINO **12**

1.1 VZNIK A VÝVOJ ARDUINA 12
1.2 ZÁKLADNÍ POPIS ARDUINA 12
1.2.1 Hlavní výhody Arduina: *13*
1.3 ARDUINO UNO 14
1.3.2 Programování Arduina UNO *19*
1.4 OSTATNÍ DRUHY OFICIÁLNÍCH DESEK ARDUINO 21
1.4.1 Arduino Mini *21*
1.4.2 Arduino Nano *21*
1.4.3 Arduino Micro *21*
1.4.4 LilyPad Arduino *22*
1.4.5 Arduino Fio *22*
1.4.6 Arduino Leonardo *22*
1.4.7 Arduino Yún *22*
1.4.8 Arduino Mega2560 ..... *22*
1.4.9 Arduino Due *23*
1.4.10 Arduino Esplora *23*
1.4.11 Arduino Robot *23*
1.4.12 Arduino Intel Galileo *23*
1.4.13 Arduino Tre *23*
1.5 SHIELDY PRO ARDUINO 24

2 KROKOVÉ MOTORY **25**

2.1 ROZDĚLENÍ KROKOVÝCH MOTORŮ DLE KONSTRUKCE 25
2.2 ŘÍZENÍ KROKOVÝCH MOTORŮ 26
2.2.1 Unipolární řízení *26*
2.2.2 Bipolární řízení *28*
2.2.3 Mikrokrokování *30*

3 PRAKTICKÁ ČÁST **31**

3.1 NÁVRH PLOŠNÉHO SPOJE 33
3.2 VÝROBA PLOŠNÉHO SPOJE 35
3.3 PLOŠNÝ SPOJ (STRANA SOUČÁSTEK) 36
3.3.1 Použité součástky *36*
3.3.2 Stabilizátor 5V *37*
3.3.3 Spínání cívek krokového motoru *37*
3.3.4 Svorkovnice pro tlačítka a koncové spínače *38*
3.3.5 Svorkovnice pro připojení krokového motoru *38*
3.3.6 Piny pro Arduino *38*
3.4 PLOŠNÝ SPOJ (STRANA MĚDI) 39
3.5 LCD DISPLEJE 40
3.6 KONCOVÉ SPÍNAČE 41

4 MOŽNÉ VYLEPŠENÍ ZÁVODU AUTOMOBILŮ **42**

ZÁVĚR **43**

SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ **44**

PŘÍLOHY 1

Úvod

V současné době je na trhu mnoho vývojových platforem s vlastním vývojovým softwarem. Mezi nejznámější platformy patří Arduino. Arduino je určeno především pro uživatele, kteří chtějí realizovat vlastní projekt a nemají velké zkušenosti s programováním. Arduino může získávat data z připojených senzorů a na základě těchto dat vyhodnocovat okolní prostředí. Dále může řídit různé připojené periferie, jako jsou například krokové motory.

Moderní elektronika, automatizační technika a elektronika stále více využívají pohony, kde se může řídit přesná poloha natočení mechanických systémů. Požadavky jsou především malá setrvačnost a velký kroutící moment. Těmto požadavkům při vhodně zvoleném řízení vyhovují krokové motory.

Cílem této bakalářské práce je popsat mikrokontroler Arduino a základní řízení krokových motorů. Vylepšit stávající model závodů automobilů, který je napájen ze stabilizovaného zdroje napětí a přes stisknutá tlačítka přímo spíná statorové cívky. Tento způsob zapojení nedrží sepnuté fáze. Pro snazší ovládání závodů automobilů je cílem vytvořit prostřednictvím mikrokontrolerů Arduino UNO vhodné řízení krokových motorů, které umožní pohyb modelů autíček.

Seznam symbolů a zkratek

PC ..... Osobní počítač

USB ..... Univerzální seriová sběrnice

C++..... Programovací jazyk C++

GND ..... Uzemnění

PWM..... Pulsně šířková modulace

ICSP..... Protokol pro seriové programování

SRAM..... Statická paměť

EEPROM..... Elektricky mazatelná paměť

I/O..... Vstup/Výstup

Rx ..... Přijímání dat

Tx..... Vysílání dat

IDE ..... Integrované vývojové prostředí

UARTs..... Univerzální asynchronní přijímač / vysílač

HDMI ..... Multimediální rozhraní s vysokým rozlišením

CNC..... Číslicové řízení počítačem

SPI ..... Seriové periferní rozhraní

TWI..... Dvouvodičové seriové rozhraní

UV ..... Ultrafialové záření

LCD ..... Displej z tekutých krystalů

LED ..... Polovodičová světelná dioda

Arduino

Arduino je open-source projekt založený na vlastním hardwaru a softwaru. To znamená, že editovat programy a schémata všech oficiálních modelů může každý. Arduino se využívá ve velkém množství aplikací a je ve světě nejrozšířenější platformou. Modely Arduino umožňují připojení a čtení periferií, mezi které mohou patřit tlačítka, senzory a další. Mohou být také použity na ovládání velkého množství zařízení, ať už se jedná o spínání osvětlení, ovládání robotů nebo také pro ovládání motorů.

Obr. 1.1 *Logo Arduino [1]*

1.1 Vznik a vývoj Arduina

Arduino vzniklo v roce 2005 v italském městě Ivrea. Cílem bylo vytvořit jednoduchý, levný a snadno dostupný vývojový set, který měl sloužit především pro studenty. Arduino se brzy stalo mezi studenty velmi populární, a tak se rozšířilo do celého světa včetně návodů a schémat. Nyní je na trhu dostupná spousta různých desek Arduina včetně jeho neoficiálních typů. [1]

1.2 Základní popis Arduina

Většina typů Arduina je založena na využívání procesorů od firmy Atmel. Procesor je spolu s dalšími elektronickými součástkami umístěn na desce plošného spoje, kde typicky pro Arduino převládá modrá barva. Většina Arduino desek obsahuje také převodník, který slouží pro komunikaci PC a procesoru. Existují však typy desek, které mají převodník zabudovaný v procesoru z důvodu úspory místa na desce. Některé typy desek nemají žádný převodník. V tomto případě je nutné využít pro programování externí převodník. [1]

1.2.1 Hlavní výhody Arduina:

- ***Nízká cena***

Desky Arduino jsou oproti jiným platformám založených na mikrokontrolerech poměrně levné. V dnešní době jsou navíc dostupné různé náhrady Arduina, které vycházejí levněji, než originální deska Arduino. Model Arduino UNO se pohybuje v ceně kolem 400 Kč. Lze si také sestavit vlastní Arduino, na internetu je mnoho návodů na sestavení. [2]

- ***Podpora více operačních systémů***

Software Arduino (IDE) je dostupný pro Windows, Linux, Macintosh OSX. Většina ostatních prostředí pro mikrokontrolery je dostupná jen pro Windows. [2]

- ***Přehledné a jednoduché programovací prostředí***

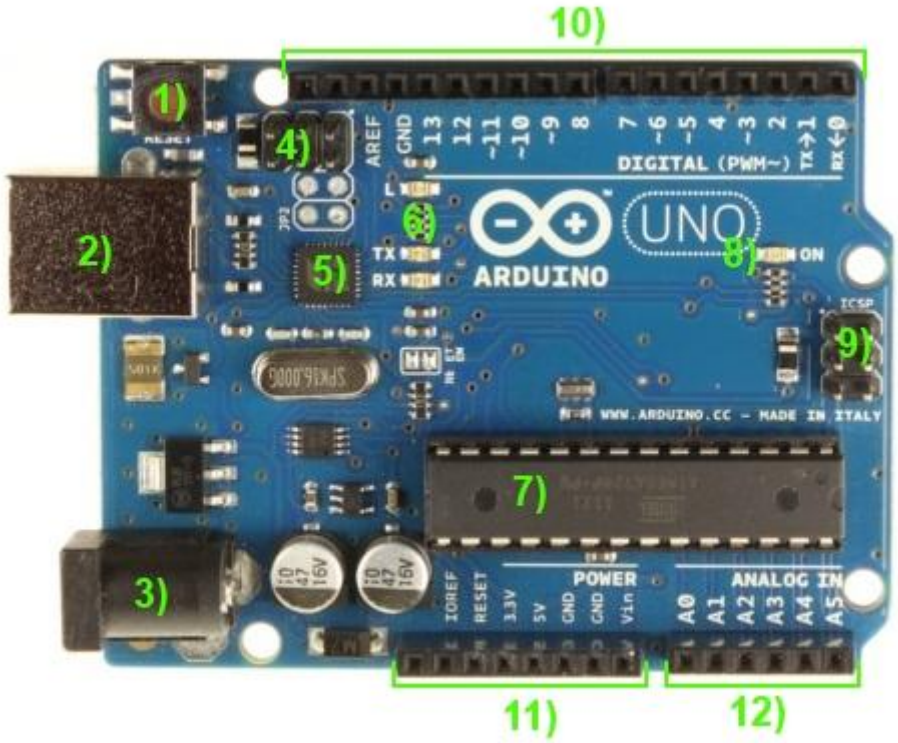
Arduino software je vhodný jak pro zkušenější uživatele, tak i pro začátečníky, kteří mají malé zkušenosti s programováním. Pro takové uživatele nabízí velké množství ukázkových kódů, které jsou implementovány přímo ve vývojovém prostředí a prezentují možnosti využití Arduina. Proto je také Arduino často využíváno jako učební pomůcka. [2]

- ***Open source***

Arduino nabízí velké možnosti využití pro uživatele, kteří méně rozumí technickým detailům. Uživatelé tak mohou použít již vytvořená zapojení desek Arduino včetně zdrojových kódů. Jazyk může být také rozšířen knihovnami vytvořenými v C++. Schémata desek Arduino jsou volně dostupné a proto si mohou návrháři vytvořit i vlastní verzi modulu. [2]

1.3 Arduino UNO

Arduino UNO patří v současné době mezi nejvyužívanější model Arduina, protože disponuje dostatečným výkonem pro většinu aplikací a zároveň má kompaktní rozměry. Nyní již existuje několik jeho verzí. Poslední z verzí je označena jako Arduino UNO Rev3. UNO pracuje na mikroprocesoru ATmega 328, který má 32 kB paměť, kde 0,5 kB zabírá bootloader, který zabezpečuje naprogramování Arduina po seriové lince a následné spuštění programu. Paměť SRAM má velikost 2 kB a EEPROM je o velikosti 1 kB. Paměť EEPROM je možné číst nebo do ní zapisovat pomocí knihovny EEPROM. Oscilátor pracuje na frekvenci 16MHz. [5]



Obr. 1.2 *Popis Arduino desky [5]*

1) Tlačítko Reset

Viditelný červený mikrospínač, který se používá pro spuštění programu od začátku. Většinou je popsán na desce nápisem RESET. [5]

2) Konektor USB typu B

U novějších modelů se můžeme setkat i s typem micro USB, jelikož je podstatně menší a nezabírá mnoho místa na desce jako klasické USB. Některé desky tyto konektory vůbec nemají, proto buďto potřebují k naprogramování externí programátor anebo je zde jiný způsob připojení například přes Bluetooth nebo Ethernet. [5]

3) Napájecí konektor

Napájení Arduina je možné provést několika způsoby. Jedním způsobem je externí napájecí zdroj, který se zapojí do napájecího konektoru desky. Doporučuje se rozsah napájecího napětí mezi 7 až 12V. Při vyšším napětí by se mohl regulátor napětí, který toto napětí externího zdroje upravuje na 5V, přehřát a poškodit desku.

Další možností je napájení z baterie. Tato možnost se nejčastěji využívá pro projekty, kde není možné dodat napájení ze sítě, nebo kde je nutná mobilita ovládaného zařízení. Externí baterie se zapojuje do pinu Vin a do pinu GND. Napájení je možné využít i přes rozhraní USB. V tomto případě napájecí napětí neprochází přes stabilizátor, ale deska je napájena přímo napětím 5V. [3]

4) ICSP rozhraní

Slouží pro externí programování USB-serial převodníku. Arduina, která nemají převodník nebo mají převodník přímo v mikrokontroleru, rozhraní ICSP neobsahují. Pro běžného uživatele je toto rozhraní nevyužitelné. [5]

5) USB-serial převodník

USB-seriál převodník je prostředník pro komunikaci mezi PC a mikrokontrolerem. [5]

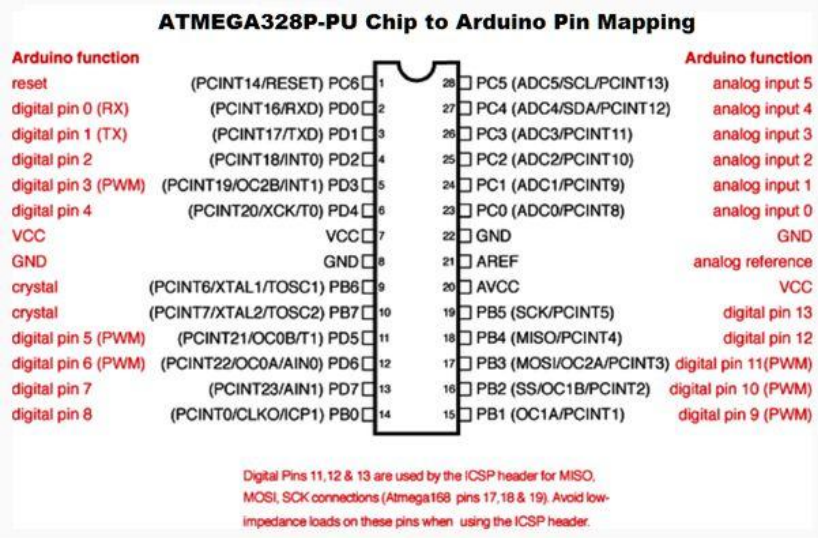
6) LED diody L, Rx, Tx

Dioda, která je popsána na desce písmenem L je často využívána, protože je připojena na digitální výstup č. 13. Dioda tedy svítí, pokud máme na tomto digitálním výstupu

hodnotu logické 1. Výhoda je, že spínání výstupu vidíme a nemusíme tak připojovat například externí diodu. Některá Arduina tuto diodu nemají. Diody, které jsou označeny písmeny Rx a Tx slouží pro seriovou komunikaci, kde dioda Rx bliká, pokud se přijímají data a dioda Tx bliká tehdy, jsou-li data posílána. [5]

7) Mikrokontroler

Mikrokontroler je integrovaný obvod, který má v sobě mnoho elektronických součástek, které tvoří elektrický obvod. Obsahuje paměť pro uložení programu a operační paměť. Najdeme ho na všech deskách, kde je většinou zapájen přímo do plošného spoje Arduina. Arduino UNO obsahuje mikrokontroler ATMEGA328P, který je zasunut do patice. [5]



Obr. 1.3 *Rozložení vývodů ATMEGA328P [9]*

8) Dioda ON

Dioda ON indikuje, že je deska Arduina napájená. [5]

9) ICSP rozhraní

ICSP rozhraní slouží pro programování mikrokontroleru. Využívají ho i některé shieldy. [5]

10) Digitální piny

Tyto piny slouží pro připojení součástek nebo obvodů, které chceme spínat.

11) Piny

Zde se nacházejí převážně napájecí piny Arduina. Pin označený názvem IOREF poskytuje referenční napětí, s kterým mikrokontroler pracuje. Správně nakonfigurovaný shield si přečte napětí na pinu a vybere příslušný zdroj napětí na 5V nebo na 3,3V. Pin RESET se typicky používá v případě, kdy je nutné k shieldu přidat resetovací tlačítko. Piny 5V a 3,3V a GND slouží pro napájení. Pin Vin slouží pro externí napájecí zdroj, kterým se může napájet deska Arduino. Externím zdrojem může být například baterie. [3]

12) Analogové vstupy / výstupy

Na tyto piny se připojují například senzory, u kterých měříme analogovou hodnotu. Mohou se využít i jako digitální vstupy a výstupy. [5]

1.3.1.1 Vstupně výstupní piny Arduina UNO

UNO obsahuje 14 I/O pinů, které mohou být využity jako vstupní nebo výstupní. To jestli bude pin využit jako vstupní nebo výstupní si uživatel zvolí na začátku programové části, kde si také nastaví, zda chce využívat vnitřní PULLUP rezistory. Tyto piny pracují s napětím 5V. Každý pin může přijímat nebo dodávat proud o hodnotě 20mA. Maximální hodnota, která nesmí být překročena je 40mA, jelikož při této hodnotě by došlo k poškození desky Arduino. Některé z I/O pinů mají specifické funkce. [3]

Piny 0, 1 slouží pro sériovou komunikaci. Pin 0 (Rx) slouží pro příjem dat a pin 1 (Tx) pro vysílání dat.

Piny 2, 3 mají využití pro externí přerušení. K jejich použití se využívá funkce attachInterrupt(). Přerušení může být vyvoláno například stoupající nebo klesající hranou nebo změnou hodnoty logické hladiny signálu.

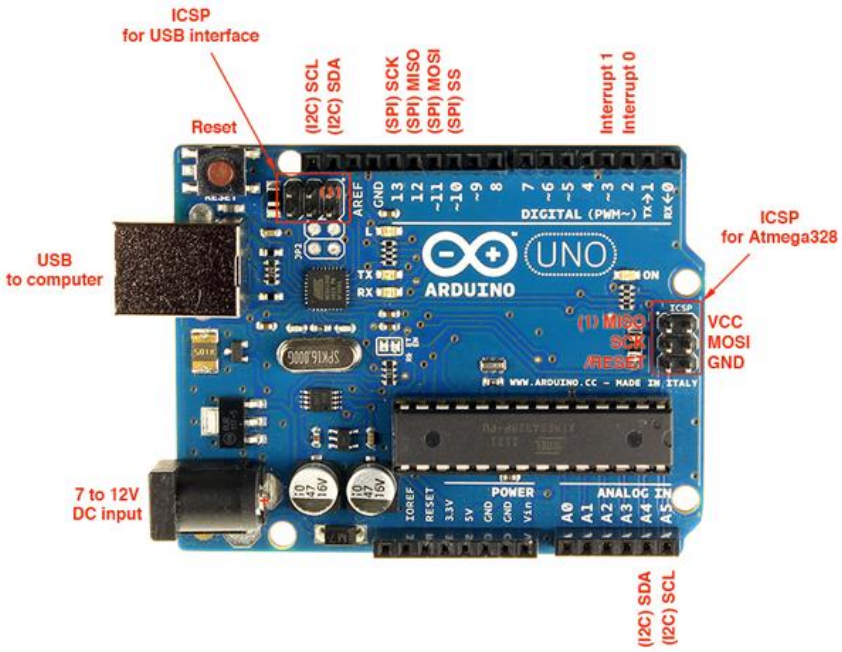
Piny 3, 5, 6, 9, 10, 11 mohou být využity pro pulsně šířkovou modulaci (PWM).

Piny 10, 11, 12, 13 s užitím knihovny SPI umožňují SPI komunikaci.

Pin 13 pro tento pin je na desce umístěna LED dioda. Nejčastěji se využívá jako ukázkový program pro Arduino, který obsluhuje diodu. Výhodou je, že se nemusí připojovat externí dioda a uživatel vidí spínání diody přímo na desce.

Piny SDA, SCL, A4, A5 podporují TWI komunikaci pomocí knihovny Wire.

UNO obsahuje na své desce 6 analogových pinů. Tyto piny umožňují měřit elektrické napětí přivedené na tyto vstupy a se změřenou hodnotou dále pracovat. Každý vstup poskytuje rozlišení 10 bitů, což je 2^{10} = 1024 různých hodnot. Výchozí nastavení je od 0 do 5 V. Horní hranice napětí lze změnit pomocí funkce analogReference(). Tyto analogové piny lze použít i jako digitální vstupy a výstupy. [3]

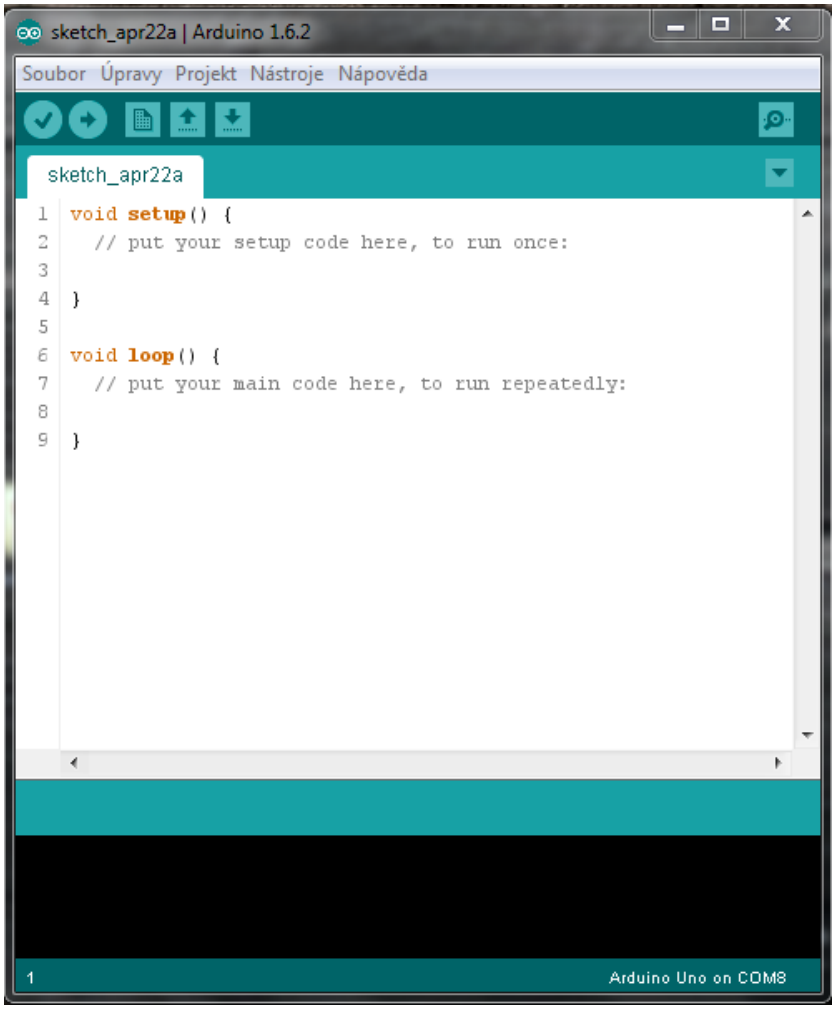


Obr. 1.4 *Rozložení pinů Arduina UNO [3]*

1.3.2 Programování Arduina UNO

Programovací prostředí se nazývá Arduino IDE a je napsáno v jazyku Java. Jeho výhodou je, že se může spustit pod větším počtem operačních systémů, kromě Windows také na Mac OSX nebo GNU/Linux. Samotné vývojové prostředí je velmi dobře navrženo pro uživatele, jelikož je jednoduché a přehledné.

Pro programování Arduina se může využít více programovacích jazyků, mezi které patří C, C++, ale je možné použít také knihovnu Wiring, což je nadstavba jazyka C++. Program ve Wiring je o něco obsáhlejší.



Obr. 1.5 *Programovací prostředí Arduino IDE*

Programovací prostředí Arduino IDE vypadá v podstatě jako rozšířený textový editor. V horní liště menu jsou klasické položky. Mezi tyto položky patří například položka soubor, úpravy, projekt, nástroje a nápověda.

- ***Soubor***

Zde mimo jiné najdeme velké množství ukázkových kódů, které prezentují základní využití desky Arduino. Tyto kódy jsou přehledně okomentovány a jsou uzpůsobeny pro základní pochopení, jak Arduino pracuje.

- ***Projekt***

Pod touto položkou v menu se ukrývá základní seznam knihoven, které jsou k dispozici již po nainstalování IDE nebo je zde možnost přidat si knihovnu vlastní. Tyto knihovny, které se dají volně stáhnout, velmi zjednodušují práci s ovládanými periferiemi, jako jsou například displeje. Uživatel nemusí složitě posílat data na displej, ale může využít některé funkce, které knihovna umožňuje.

- ***Nástroje***

Tato položka se využívá předtím, než začneme Arduino programovat. Je nutné zde nastavit správný název desky Arduino a také vybrat správný port ke kterému je Arduino připojeno. V menu nástroje je také možnost zobrazit seriový monitor. V případě komunikace přes seriovou linku vidíme, která data se právě přenášejí.

Pod hlavním menu můžeme najít další funkční tlačítka, které mají následující funkce:

Tlačítko ověřit - zkontroluje správnost napsaného kódu. V případě, že objeví některé chyby, objeví se nápis „chyba kompilace“. Vypíše se seznam chyb, které je nutné opravit. V případě, že je vše v pořádku a v kódu není žádná chyba, je možné nahrát program do mikrokontroleru.

Tlačítko nahrát - nejdříve se zkontroluje správnost napsaného kódu a pokud zde není chyba, program se nahraje do mikrokontroleru.

Tlačítko nový - založí nový sketch. Sketch je pojmenování pro vytvářený projekt.

Tlačítko otevřít - slouží pro otevření vytvořeného projektu.

Tlačítko uložit - slouží pro uložení projektu.

Tlačítko seriový monitor - po stisknutí tohoto tlačítka se mohou sledovat přenášená data po seriové lince.

1.4 Ostatní druhy oficiálních desek Arduino

V dnešní době je na trhu poměrně široká nabídka desek Arduino, každý model má specifické využití v určitých aplikacích.

1.4.1 Arduino Mini

Tato verze se řadí k nejmenším z oficiálních verzí Arduina. Z důvodu malých rozměrů na desce chybí USB port, a proto je nutné pro programování využít externí převodník. Využívá se zde procesor ATmega328 o taktu 16MHz. Výkon je srovnatelný s ostatními většími deskami Arduina. Pro svoji velikost má využití například v dálkových ovladačích a chytrých vypínačích. Obsahuje 8 analogových vstupních pinů a 14 vstupně výstupních pinů, kde 6 jich může být použito jako PWM výstup. [1]

1.4.2 Arduino Nano

Arduino Nano je velmi podobné verzi Arduino Mini. Zásadním rozdílem je přítomnost převodníku a portu USB. Je uzpůsobeno pro zapojení do nepájivého pole. [1]

1.4.3 Arduino Micro

Deska obsahuje 20 vstupně výstupních pinů, kde 7 jich může být použito jako PWM výstupy a 12 jako analogové vstupy. Používá se převážně pro vytvoření vlastního herního ovladače nebo klávesnice. Deska je osazena procesorem ATmega32u4 o frekvenci 16MHz,

který má v sobě zabudovaný převodník. Pro PC se může tvářit jako klávesnice nebo myš a může posílat signály stisknutí klávesy nebo pohybu myši. Tuto operaci umožňují i ostatní desky Arduino, ale je nutné u nich přeprogramovat převodník. [1]

1.4.4 LilyPad Arduino

LilyPad Arduino založené na ATmega328 o frekvenci 8MHz má minimální množství součástek na desce, a proto se díky kompaktním rozměrům využívá především v chytrém oblečení. Vodivé spoje v oblečení jsou tvořeny vodivými nitěmi. K dostání je i ve více variantách, které mohou obsahovat USB. [1]

1.4.5 Arduino Fio

Obsahuje procesor ATmega328P o frekvenci 8MHz. Deska je používána především pro připojení bezdrátových modulů, mezi které patří například Wi-Fi nebo Bluetooth. Tyto bezdrátové moduly většinou pracují s napětím 3,3V. Na této desce je využito napětí 3,3V oproti klasickým 5V, které využívá většina ostatních desek Arduina. [1]

1.4.6 Arduino Leonardo

Patří k nejnovější desce v rodině Arduino. Jeho srdcem je procesor ATmega32u4 pracující na frekvenci 16MHz. Má 20 vstupně výstupních pinů. [8]

1.4.7 Arduino Yún

Využívá čipu ATmega32u4, kromě tohoto čipu je zde také čip Atheros AR9331, který umožňuje běh Linuxu Linino. Komunikaci mezi těmito čipy zajišťuje softwarový bridge. Jedná se tedy o velmi výkonnou desku. Na desce jsou dva USB porty. Jeden slouží pro potřebu Linuxu a jedná se o port USB 2.0. a druhý port microUSB je využíván pro programování ATmega32u4. Navíc zde najdeme ještě ethernet port pro připojení k síti. [1]

1.4.8 Arduino Mega2560

Tento typ Arduina se využívá především v náročnějších aplikacích, kde je třeba více vstupně výstupních pinů. Deska je založena na výkonném ATmega 2560 a krystalovém oscilátoru 16MHz. Obsahuje 54 digitálních vstupně výstupních pinů, u kterých jich lze 14 použít jako PWM. Dále obsahuje 16 analogových vstupů a 4 UARTs seriové porty. [1]

1.4.9 Arduino Due

Arduino Due je první Arduino, které je založeno na výkonném 32-bitovém mikroprocesoru Cortex-M3, který je mnohem výkonnější, než čip na Arduino Mega. Frekvence oscilátoru je 84MHz. Ostatní desky Arduino mají převážně 8-bitové čipy s maximální frekvencí 16MHz. Deska je vybavena dvěma USB konektory. Jeden slouží k připojení klávesnice nebo myši a druhý pro programování čipu. Obsahuje 54 digitálních vstupně výstupních pinů. Operační napětí, které může být na I/O pinech je 3,3V. Shieldy, které pracují s 5V jsou zcela nevhodné pro použití u tohoto typu Arduina, jelikož by mohlo dojít k jeho poškození. Je nutné použít shieldy pracující na napětí 3,3V. [1]

1.4.10 Arduino Esplora

Tento typ Arduina je osazen procesorem ATmega32u4 a je určen pro vytvoření vlastního herního setu. Přímo na desce obsahuje joystick, posuvný potenciometr a tlačítka. Je zde také zabudován piezzo bzučák, tříosý akcelerometr, teploměr a piny pro připojení LCD displeje. [1]

1.4.11 Arduino Robot

Jedná se o desku, která je vyvinuta pro vytvoření vlastního robota. Obsahuje procesor ATmega32u4. Zajímavostí Arduina Robot je, že na desce je obsažen kompas. [1]

1.4.12 Arduino Intel Galileo

První deska, která vznikla spoluprácí společnosti Intel je osazena 32-bitovým čipem Intel Quark SoC X1000 s frekvencí 400Hz. Obsahuje 6 analogových vstupů a 14 vstupně výstupních digitálních pinů. Je zde Ethernet port a dva USB porty, a navíc slot pro microSD kartu. Je zde také mini-PCI Express slot, který je využíván pro připojení přídavných karet. Arduino Intel Galileo podporuje shieldy pracující na napětí 5V, ale pomocí přepnutí jumperu mohou být použity shieldy pracující s napětím 3,3V. [1]

1.4.13 Arduino Tre

Arduino Tre, které ještě není uvedeno na trhu, bude nejnovějším a zároveň nejvýkonnějším typem Arduina. Bude obsahovat procesor s taktem 1 GHz, a proto bude k využití především pro náročné výpočetní aplikace. Bude obsahovat HDMI port, 5 USB

portů, kde jeden z nich bude využíván pro programování Arduina a ostatní budou využity pro připojení zařízení k Linuxu. Na desce budou také dva audio konektory. Díky těmto parametrům bude schopný tento model konkurovat menším počítačům, mezi které patří například Raspberry Pi. [1]

1.5 Shieldy pro Arduino

Shieldy můžeme označovat za jakousi nadstavbu Arduina. Jedná se o desky, které je možné snadno připojit k Arduinu. V dnešní době je na trhu velké množství těchto shieldů, mezi které patří například Ethernet shield, shieldy pro ovládání motorů, Wi-Fi shield a další. [4]

2 Krokové motory

Principem krokových motorů jsou různě modifikované synchronní stroje, které mají schopnost převést změnu buzení na přesný úhel natočení, tzv. krok. [10].

Krokové motory mají dvě základní části, ze kterých se skládají - stator a rotor. Stator je tvořen z cívek, které jsou nejčastěji spínány pomocí řídící elektroniky. Proud, který prochází cívkou statoru, vytváří magnetické pole, které přitahuje pól magnetu rotoru. Rotor je tvořen permanentními magnety a je uložen ve valivých kuličkových ložiskách. Vhodné spínání cívek vytvoří rotující magnetické pole, které otáčí rotorem. [11]

2.1 Rozdělení krokových motorů dle konstrukce

Krokové motory můžeme rozdělit podle konstrukčního provedení na tři základní typy:

- ***Krokové motory s pasivním rotorem***

Krokové motory s pasivním rotorem jsou označovány také jako reluktanční nebo reakční. Stator i rotor jsou nejčastěji tvořeny z ocelových plechů s vysokou permeabilitou. Na statoru a rotoru jsou umístěny vyjádřené póly. Při připojení napětí na stator se vytvoří magnetické pole na jednom páru pólů. Následně dojde k přitažení rotorových pólů k pólům statoru.

- ***Krokové motory s aktivním rotorem***

Krokové motory s aktivním rotorem mají rotor tvořen permanentními magnety. Stator je tvořen vyjádřenými póly. Pohyb motoru je založen na principu působení magnetických sil. Krokové motory s aktivním rotorem mají větší kroutící moment než motory s pasivním rotorem.

- ***Krokové motory hybridní***

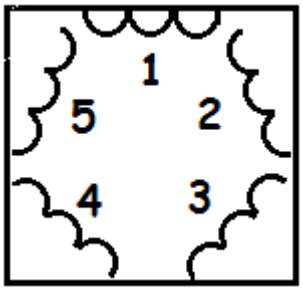
Hybridní krokové motory jsou kombinací aktivního a pasivního krokového motoru. Rotor je tvořen permanentním magnetem. Na pólech permanentního magnetu jsou nasazeny nástavce, které mají na sobě drážky. [13] [14]

2.2 Řízení krokových motorů

Variantu řízení krokového motoru volíme podle požadovaného kroutícího momentu, přípustného odběru a přesnosti nastavení polohy. Otáčení motoru je omezeno na několik stovek kroků za sekundu z důvodu přechodových magnetických jevů. Kdyby se tato frekvence spínání cívek zvýšila nebo by byl motor příliš zatížen, docházelo by ke ztrátě kroků. Řízení krokových motorů je možné rozdělit do dvou základních skupin - řízení unipolární a řízení bipolární. [11]

2.2.1 Unipolární řízení

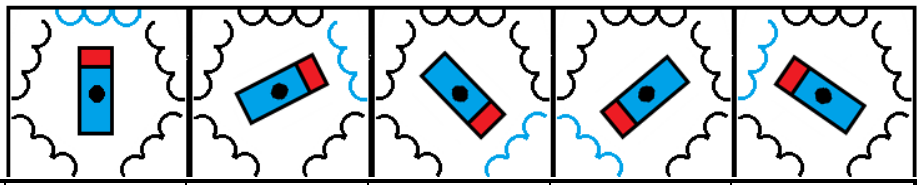
U unipolárního řízení může proud procházet pouze jedním směrem. S tímto buzením má motor nejmenší odběr, ale zároveň tedy i nejmenší kroutící moment. Výhodou tohoto řízení je, že elektronika spínání je velmi jednoduchá. Ke spínání cívek se může použít na každou cívku jeden tranzistor. Tranzistory mohou být spínány například přes Arduino. Pro menší motory lze využít integrovaný obvod ULN2803. [11]



Obr. 2.1 *Rozložení cívek na statoru*

2.2.1.1 Jednofázové unipolární řízení

U jednofázového unipolárního řízení prochází proud v jednom okamžiku pouze jednou cívkou. Na obr. 2.2 je znázorněno natáčení rotoru podle spínaných cívek na statoru, který je tvořen pěti cívkami. Sepnutá cívka, která přitahuje pól magnetu rotoru, je v tabulce znázorněna číslicí 1 a vyznačena modrou barvou. Postupným spínáním těchto cívek je vytvářen rotační pohyb. Tento typ řízení je použit v praktické části této bakalářské práce. [11]

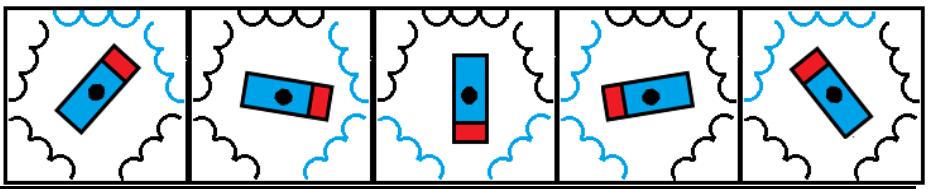


Cívka 1	1	0	0	0	0
Cívka 2	0	1	0	0	0
Cívka 3	0	0	1	0	0
Cívka 4	0	0	0	1	0
Cívka 5	0	0	0	0	1

Obr. 2.2 *Aktivní cívky při jednofázovém unipolárním řízení*

2.2.1.2 Dvoufázové unipolární řízení

Dvoufázové unipolární řízení využívá průchodu proudu vždy dvěma cívkami statoru v jednom okamžiku. Cívky s protékajícím proudem se vždy nachází vedle sebe. Rotor se natočí do polohy mezi aktivními cívkami. Výhodou tohoto řízení je, že motor má dvojnásobný kroutící moment oproti jednofázovému řízení. Zároveň má ale dvojnásobně vyšší spotřebu elektrické energie. [11]

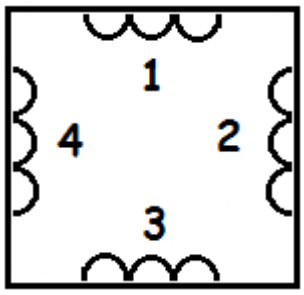


Cívka 1	1	0	0	0	1
Cívka 2	1	1	0	0	0
Cívka 3	0	1	1	0	0
Cívka 4	0	0	1	1	0
Cívka 5	0	0	0	1	1

Obr. 2.3 *Aktivní cívky při dvoufázovém unipolárním řízení*

2.2.2 Bipolární řízení

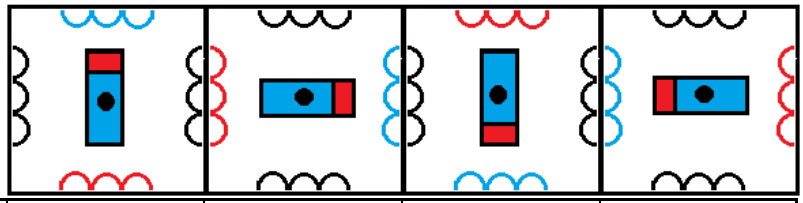
Proud prochází vždy dvěma protilehlými cívkami. Tyto cívky jsou zapojeny tak, aby měly vůči sobě opačně orientované magnetické pole. Výhodou bipolárního řízení je, že oproti unipolárnímu řízení má větší kroutící moment. Nevýhodou je však vyšší spotřeba energie. K řízení jsou potřeba dva H-můstky (pro každou větev jeden). Integrovaný obvod, který je vhodný pro toto řízení je H-můstek L293D. [11]



Obr. 2.4 *Rozložení cívek na statoru*

2.2.2.1 Jednofázové bipolární řízení

U jednofázového bipolárního řízení v jednom okamžiku prochází proud dvěma protilehlými cívkami, které tvoří navzájem opačně orientované magnetické pole. Tímto řízením se docílí vyššího kroutícího momentu, ale za cenu vyšší spotřeby elektrické energie. [11]

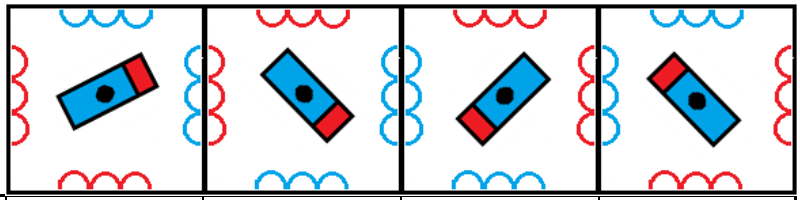


Cívka 1	1	0	1	0
Cívka 2	0	1	0	1
Cívka 3	1	0	1	0
Cívka 4	0	1	0	1

Obr. 2.5 *Aktivní cívky při jednofázovém bipolárním řízení*

2.2.2.2 Dvoufázové bipolární řízení

Při tomto řízení prochází proud všemi cívkami statoru. Rotor je natočen vždy mezi dvěma sousedními cívkami. Dvě cívky vždy přitahují kladný pól, respektive záporný pól permanentního magnetu rotoru. Oproti jednofázovému unipolárnímu řízení má toto řízení vyšší moment, ale zároveň i vyšší spotřebu elektrické energie. [11]

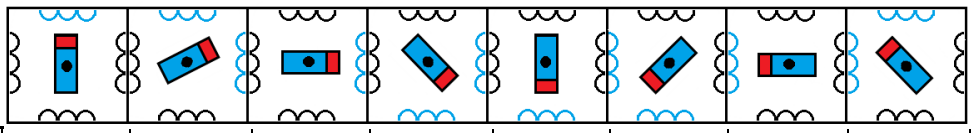


Cívka 1	1	1	1	1
Cívka 2	1	1	1	1
Cívka 3	1	1	1	1
Cívka 4	1	1	1	1

Obr. 2.6 *Aktivní cívky při dvoufázovém bipolárním řízení*

2.2.2.3 Bipolární řízení s polovičním krokem

Při řízení s polovičním krokem je výhoda, že krokový motor má větší počet poloh natočení, což umožňuje vyšší přesnost natočení rotoru. Proud prochází v jednom okamžiku jednou nebo dvěma cívkami, podle aktuálního natočení. Nevýhodou tohoto řízení je kolísání kroutícího momentu. Způsob řízení je uveden na obr. 2.7. [11]



Cívka 1	1	1	0	0	0	0	0	1
Cívka 2	0	1	1	1	0	0	0	0
Cívka 3	0	0	0	1	1	1	0	0
Cívka 4	0	0	0	0	0	1	1	1

Obr. 2.7 *Aktivní cívky při řízení s polovičním krokem*

2.2.3 Mikrokrokování

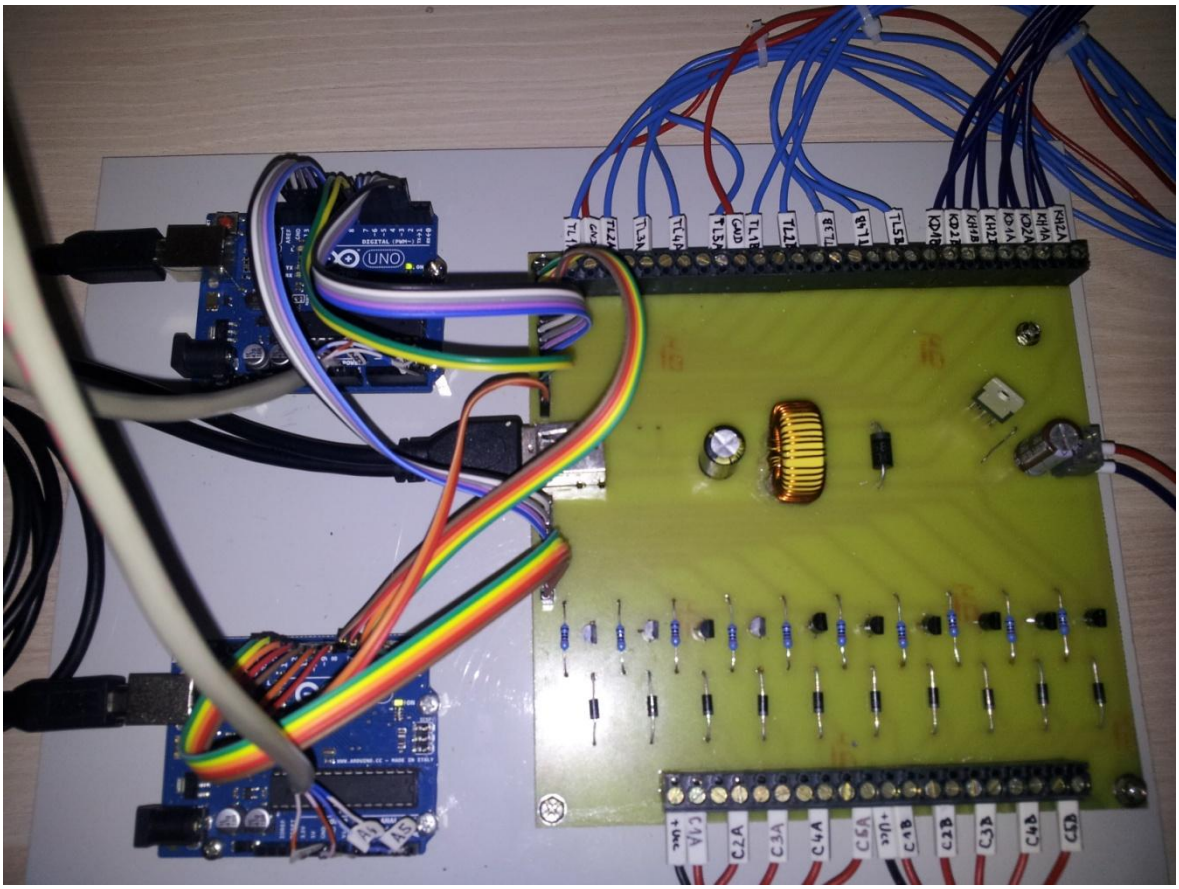
Mikrokrokování spočívá v rozdělení plné délky kroku do menších inkrementů pohybu rotoru. V jedné fázi krokového motoru se postupně snižuje velikost budícího proudu protékaného cívkou a v sousední fázi se velikost budícího proudu zvyšuje. Výsledný vektor magnetické indukce se pohybuje mezi krajními polohami. Mikrokrokování zvyšuje citlivost krokového motoru. Při zvyšování počtu poloh však značně klesá točivý moment. Pokud je frekvence mikrokrokování mála, začnou se objevovat mechanické rezonance motoru. [12][13]

3 Praktická část

Model závodů automobilů je tvořen z několika částí. Hlavní částí jsou dva krokové motory s aktivním rotorem. Krokové motory mají na statoru umístěno 5 cívek. Jsou řízeny pomocí unipolárního jednofázového řízení. Motory jsou ovládány tlačítky. Každé tlačítko spíná jednu statorovou cívku krokového motoru. Postupným spínáním cívek se navíjí imitace na hřídel rotoru a pohybují se modely autíček.

Obr. 3.1 *Pohled na model závodů automobilů*

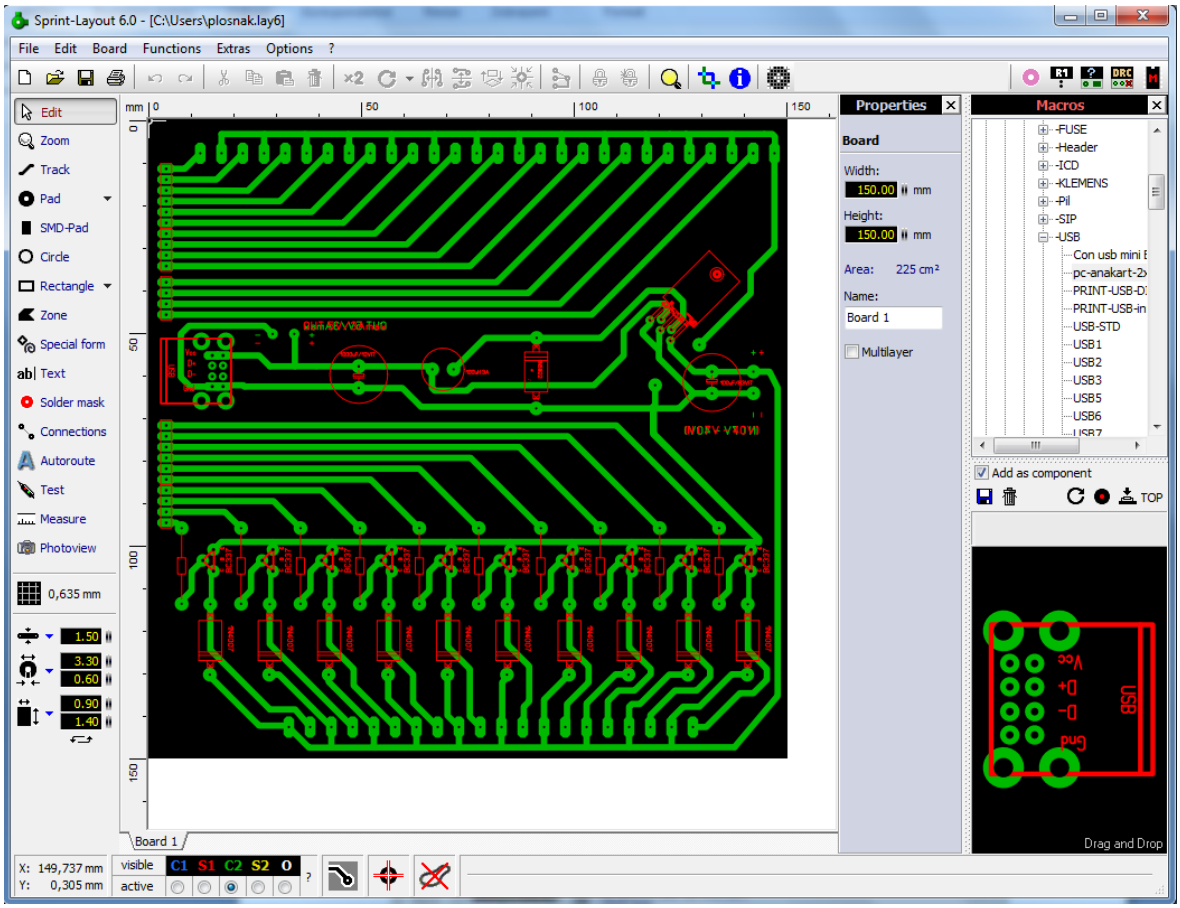
Pro ovládání modelu závodu automobilů byla použita navržená řídící elektronika a dvě desky Arduino UNO. Program pro Arduino UNO je uveden v příloze této práce. Jedna deska Arduino UNO ovládá jízdu červeného autíčka a druhá ovládá jízdu modrého autíčka na nakloněné rovině. Obě Arduina mají v sobě nahrán stejný program. Při zapnutí stabilizovaného napájecího napětí o velikosti 30V se obě autíčka nastaví řídícím programem do výchozí polohy. To znamená, že autíčka automaticky pomocí spínání fází krokového motoru pojedou dolů, dokud se nesepne dolní koncový spínač. Po sepnutí dolního koncového spínače se autíčka zastaví a odblokují se tlačítka. Postupným spínáním tlačítek se autíčka rozjedou směrem nahoru a po uvolnění dolního koncového spínače se začne počítat čas zobrazovaný na LCD displeji. Spínané statorové cívky krokového motoru jsou detekovány rozsvícením LED diody. Jestliže autíčko vyjede do horní polohy a sepne koncový spínač, zastaví se čas na displeji. Programově je nastaveno, že autíčko zůstane pět sekund v horní pozici. Poté automaticky sjíždí dolů, dokud se nesepne dolní koncový spínač. Celý cyklus se může opakovat.



Obr. 3.2 *Řídící elektronika závodů automobilů*

3.1 Návrh plošného spoje

Plošný spoj byl navržen pomocí softwaru Sprint-Layout. Tento software je určen pro vytváření jednovrstvých, dvouvrstvých i vícevrstvých desek plošných spojů. Sprint-Layout obsahuje rozsáhlé knihovny, kde jsou běžně dostupné součástky. Je zde i možnost vytvoření vlastních součástek. Lze také přidat další knihovny se součástkami, které jsou dostupné na internetu. Výhodou je přehledné a intuitivní prostředí, které uživateli umožňuje rychle se orientovat a naučit se s tímto softwarem pracovat.



Obr. 3.3 *Prostředí Sprint-Layout*

Nástroje pro jednotlivé pracovní kroky při výrobě plošného spoje, jsou umístěny v levém sloupci a obsahují plošky pro pájení, spoje, nápisy a další. Kliknutím se vybere nástroj, se kterým chceme pracovat. Pod nástroji se mohou nastavovat vlastnosti jako například šířka spoje, vnitřní i vnější průměr pájecího bodu a jemnost mřížky. Pomocí nástroje jemnost mřížky se může nastavit vysoká jemnost a práce je potom opravdu velmi přesná, jelikož lze nastavit přesnost až 1/100 mm. [7]

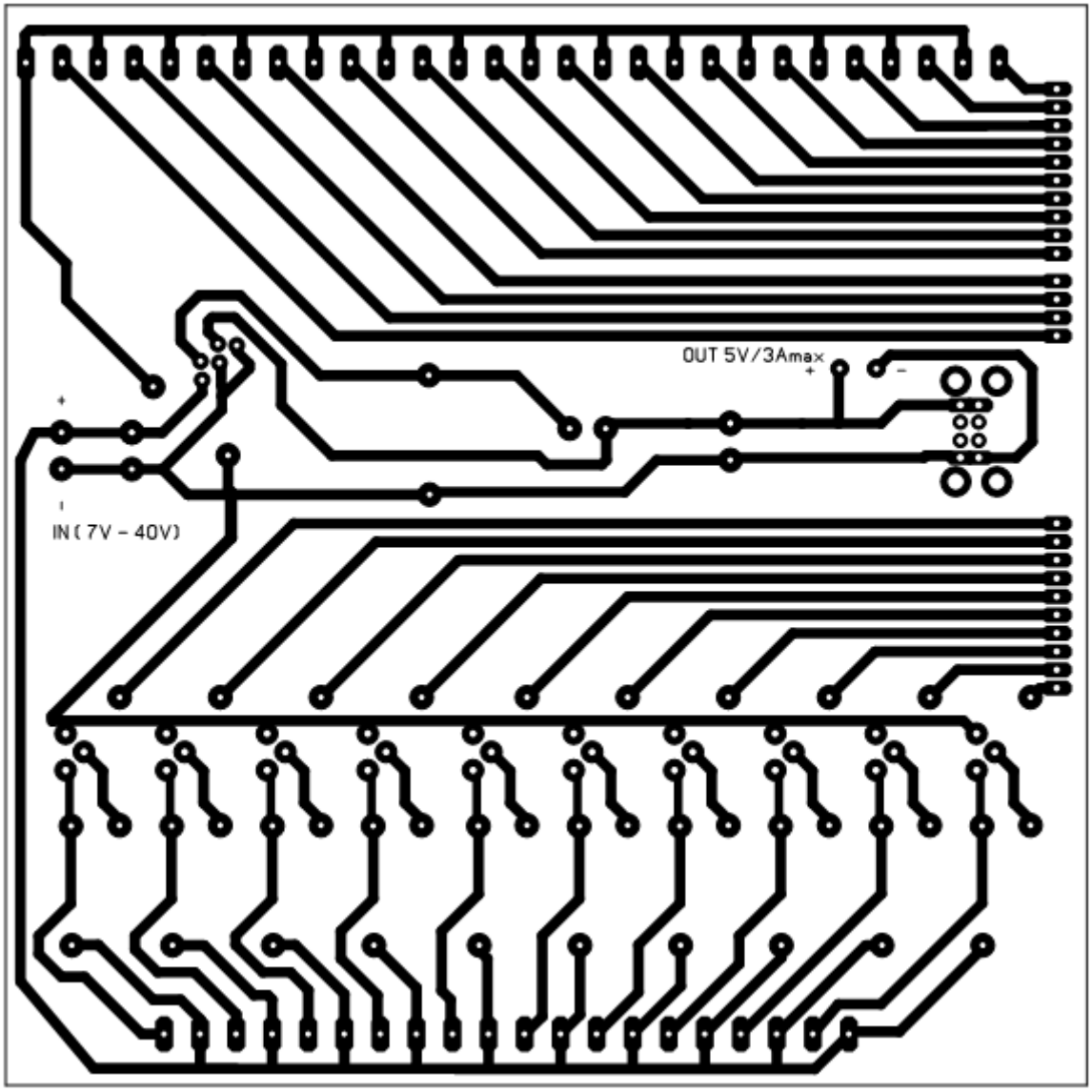
Sprint-Layout umožňuje práci ve více vrstvách plošného spoje a mezi těmito vrstvami se dá snadno přepínat. Každá strana vrstvy má plochu součástek a plochu spojů.

Software obsahuje také funkci autorouter, která ulehčuje práci při návrhu plošného spoje. Uživatel propojí součástky a software automaticky najde nejkratší cestu, kde vytvoří cestičku plošného spoje. Tato funkce však neumí navrhnout celý plošný spoj a je tedy nutná ruční úprava.

Navržený plošný spoj lze exportovat do formátů GIF, BMP, JPG, EMF nebo také do formátu Gerber, který se používá pro profesionální zhotovení plošných spojů a používá ho většina výrobců plošných spojů. K tomuto formátu je také možný export dat pro automatické vrtání děr, kde se používá standardní formát Excellon. [7]

Software také podporuje výrobu plošných spojů pomocí CNC fréz. [7]

Při tisku lze nastavit měřítko, barvy jednotlivých vrstev, zrcadlové obrácení, rámeček a další velké množství možností.



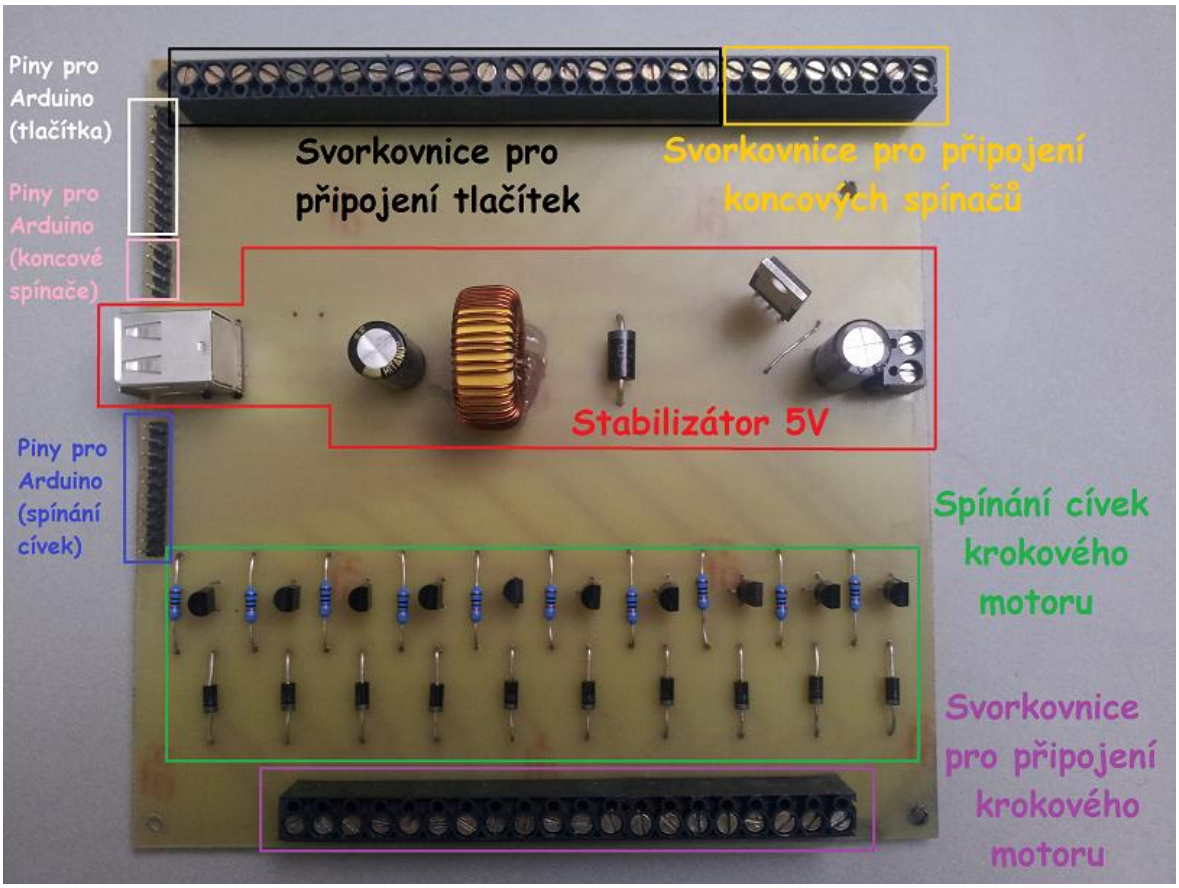
Obr. 3.4 *Navržený plošný spoj (strana mědi)*

3.2 Výroba plošného spoje

Navržený plošný spoj jsem si nechal vyrobit z důvodu spolehlivosti a přesnosti plošného spoje pomocí filmové předlohy. Předloha byla přenesena na fotocuprextit pomocí osvitu UV světlem. Působením UV světla došlo k narušení fotocitlivé vrstvy a následně se plošný spoj vyleptal v roztoku Chloridu železitého $FeCl_3$.

Následně jsem plošný spoj očistil a osadil součástkami.

3.3 Plošný spoj (strana součástek)



Obr. 3.5 *Popis plošného spoje*

3.3.1 Použité součástky

- ***Stabilizátor 5V***

Svorkovnice	
Stabilizátor	LM2576T-5
Elektrolytický kondenzátor	100uF/63V, low ESR
Dioda	1N5822
Toroidní tlumivka	DPO100A3
Elektrolytický kondenzátor	1000uF/10V, low ESR
Konektor USB	USB PCB 2AW

- ***Spínání cívek krokového motoru***

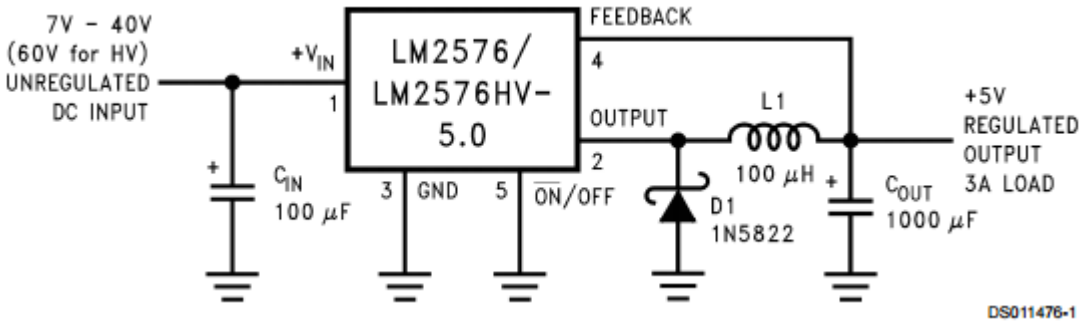
10 x Rezistor	820Ω
10 x Dioda	1N4007
10 x Tranzistor NPN	BC337-16

- ***Ostatní použité součástky***

Svorkovnice
Piny pro Arduino

3.3.2 Stabilizátor 5V

Stabilizátor slouží pro napájení dvou desek Arduino UNO pomocí kabelu USB, který se zapojuje do konektoru USB umístěného v levé části plošného spoje. Vstupní napětí, které je přiváděno na stabilizátor, může být v rozmezí 7V až 40V. Vstupní napětí je připojeno ze stabilizovaného zdroje na svorkovnici, která je umístěna vpravo na plošném spoji a v schématu je označena jako CON F. Stabilizátor má poměrně velký rozsah napájecího napětí. V tomto případě je ale využito stabilizované napětí 30V ze zdroje pro napájení cívek krokového motoru. Výstupní napětí stabilizátoru má hodnotu 5V. Maximální proudové zatížení je 3A. Tento stabilizátor je, co se týče maximálního výstupního proudu, předimenzovaný z důvodu případného pozdějšího vylepšení modelu závodu automobilů. Na plošném spoji je navíc vyvedeno napětí 5V i mimo konektor USB, kde se v případě pozdějšího využití může osadit svorkovnice, z které je tedy možné odebírat 5V. Stabilizátor je zapojen dle datasheetu, který poskytuje výrobce stabilizátoru. Schéma stabilizátoru je uvedeno na obr 3.6.



Obr. 3.6 *Schéma zapojení stabilizátoru LM2576T-5 [6]*

3.3.3 Spínání cívek krokového motoru

Spínání jednotlivých cívek statoru je realizováno pomocí 10 bipolárních tranzistorů NPN. Na každý krokový motor je použito 5 tranzistorů, protože oba krokové motory mají 5 fází, respektive 5 statorových cívek. Je zde využito jednofázové unipolární řízení

krokového motoru. Báze je připojena přes rezistor o hodnotě 820Ω na I/O pin Arduino UNO, který je programově nastaven jako výstupní. V případě nastavení pinu Arduina na logickou 1 dojde k otevření tranzistoru a začne tranzistorem protékat proud. Mezi kolektorem a napájecím napětím zdroje je připojena cívka statoru a paralelně na ni je v závěrném směru zapojena dioda 1N4007. Tato dioda pohlcuje napěťové špičky vznikající na statorové cívce představující indukční zátěž. V případě, že by zde dioda nebyla, mohlo by dojít k průrazu tranzistoru. Emitor tranzistorů je spojen se zemí. Paralelně na statorové cívky jsou zapojeny LED diody s předřadnými rezistory. Pro červené LED je hodnota rezistoru 1200Ω a pro modré LED diody 1000Ω.

3.3.4 Svorkovnice pro tlačítka a koncové spínače

V horní části plošného spoje je svorkovnice, na kterou se připojují tlačítka a koncové spínače, viz obr. 3.5. Na svorkovnici pro tlačítka jsou některé svorky nevyužité. Původní řešení uvažovalo o připojení samostatných tlačítek. Později byl použit jeden společný vodič pro celou sadu tlačítek. Schéma zapojení je uvedeno v příloze této bakalářské práce. Svorkovnice pro tlačítka a koncové spínače je ve schématu označena jako CON D.

3.3.5 Svorkovnice pro připojení krokového motoru

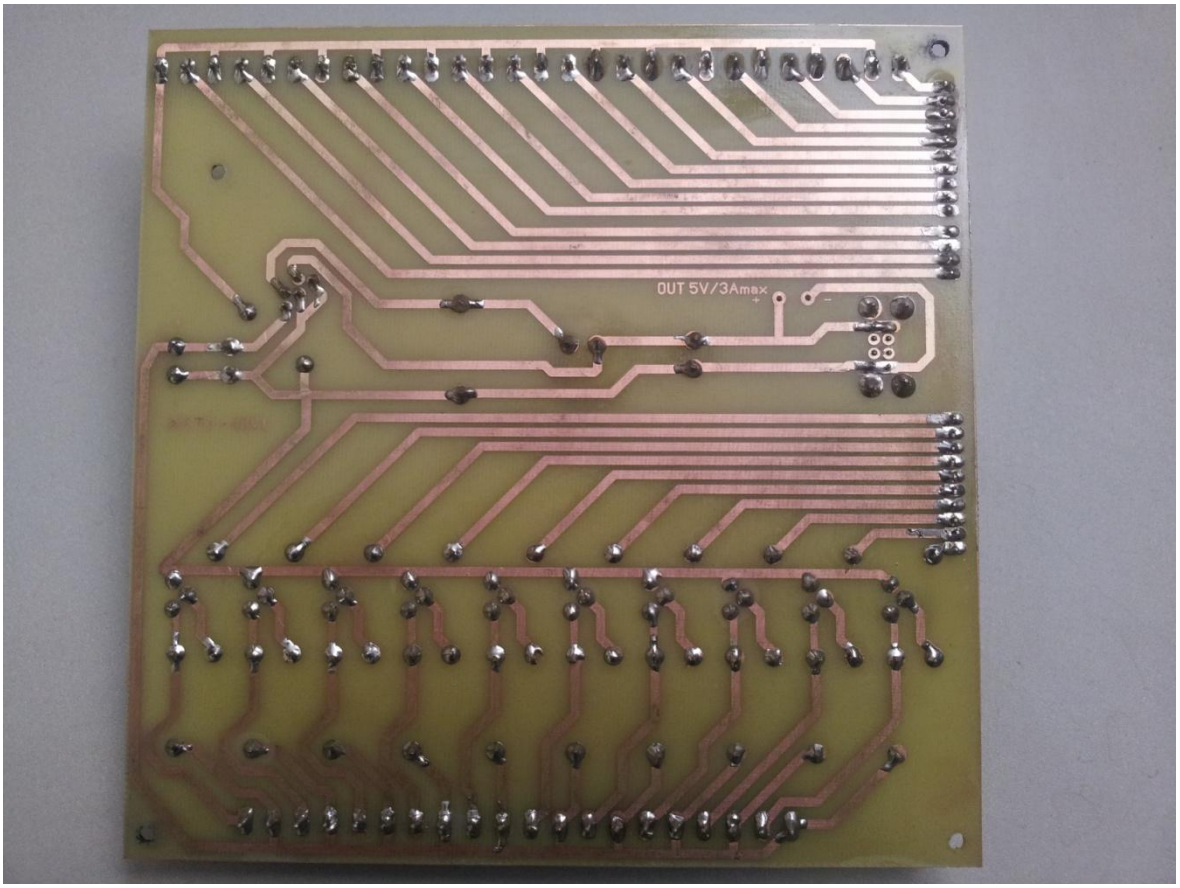
V dolní části plošného spoje se nachází svorkovnice pro připojení statorových cívek krokových motorů. Z důvodu propojení cívek u krokového motoru jsou některé části svorkovnice nevyužity. Svorkovnice je ve schématu označena jako CON E.

3.3.6 Piny pro Arduino

Piny pro Arduino slouží pro propojení navrženého plošného spoje se vstupně výstupními piny desek Arduino UNO. Piny pro tlačítka jsou na Arduinu zapojeny do pinů označených číslicemi 2, 3, 4, 5, 6 a na plošném spoji na piny označené CON A. Piny pro koncové spínače jsou přivedeny na Arduino, kde na pin číslo 7 je přiveden dolní koncový spínač a na pin 8 horní koncový spínač. Na plošném spoji jsou piny koncových spínačů přivedeny na piny označené CON B. Piny pro spínání statorových cívek jsou na Arduinu označeny číslicemi 9, 10, 11, 12, 13 a jsou připojeny na piny označené CON C.

3.4 Plošný spoj (strana mědi)

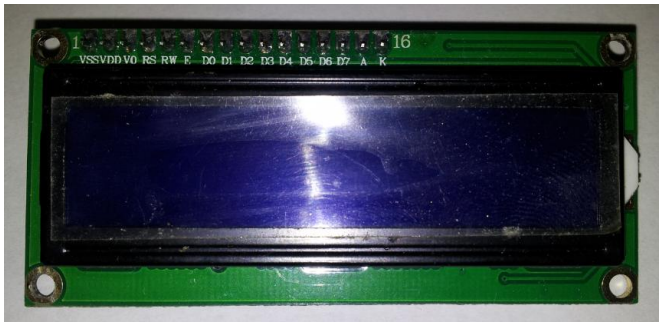
Všechny pájené spoje byly vizuálně zkontrolovány, zda došlo ke spojení cínu a mědi. Následně byly všechny spoje proměřeny pomocí multimetru, zda jsou spolehlivě vodivé. Strana mědi byla po připájení součástek natřena roztokem isopropylalkoholu a kalafuny, aby se na měděných cestičkách nevytvářela oxidace. Velikost napětí 30V ze stabilizovaného zdroje napětí je využita jako vstupní napětí pro stabilizátor a pro spínání cívek statoru. Pro tlačítka je využita záporná polarita napětí zdroje. Jelikož jsou v programu zapnuty PULLUP rezistory, tak zde není pro tlačítka potřeba žádný rezistor umístěný na desce plošného spoje. Spíná se tedy záporná polarita napětí, která je přiváděna na vstupně výstupní piny Arduina, které jsou programově nastavené jako vstupní.



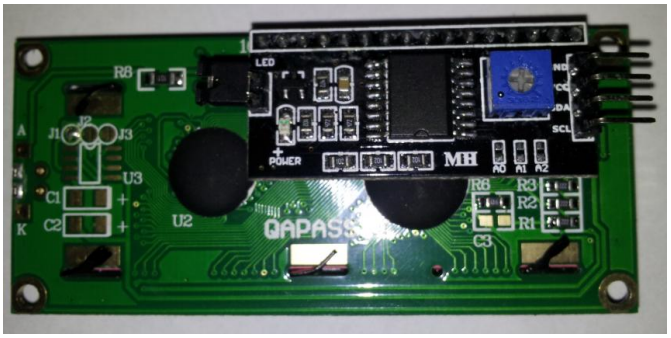
Obr. 3.7 *Plošný spoj (strana mědi)*

3.5 LCD Displeje

Pro každé Arduino UNO je použit jeden LCD displej, který má 2 řádky a do každého řádku je umístěno 16 znaků. Proto je tento displej většinou označován jako LCD displej 16x2. Vývody displeje jsou připájeny na seriový převodník I2C. Díky převodníku je možné displej propojit s Arduinem pomocí čtyř vodičů. Dva vodiče jsou určeny pro napájení displeje. Displej je napájen napětím 5V přímo z pinů Arduina, které jsou označeny na Arduinu nápisem 5V a GND a na displeji jsou označeny jako Vcc a GND. Další dva vodiče jsou využívány pro posílání dat na displej a jsou označeny na displeji jako SDA a SCL. SDA se zapojuje na Arduinu na pin, který je označen nápisem A4. SCL se zapojuje na pin A5.



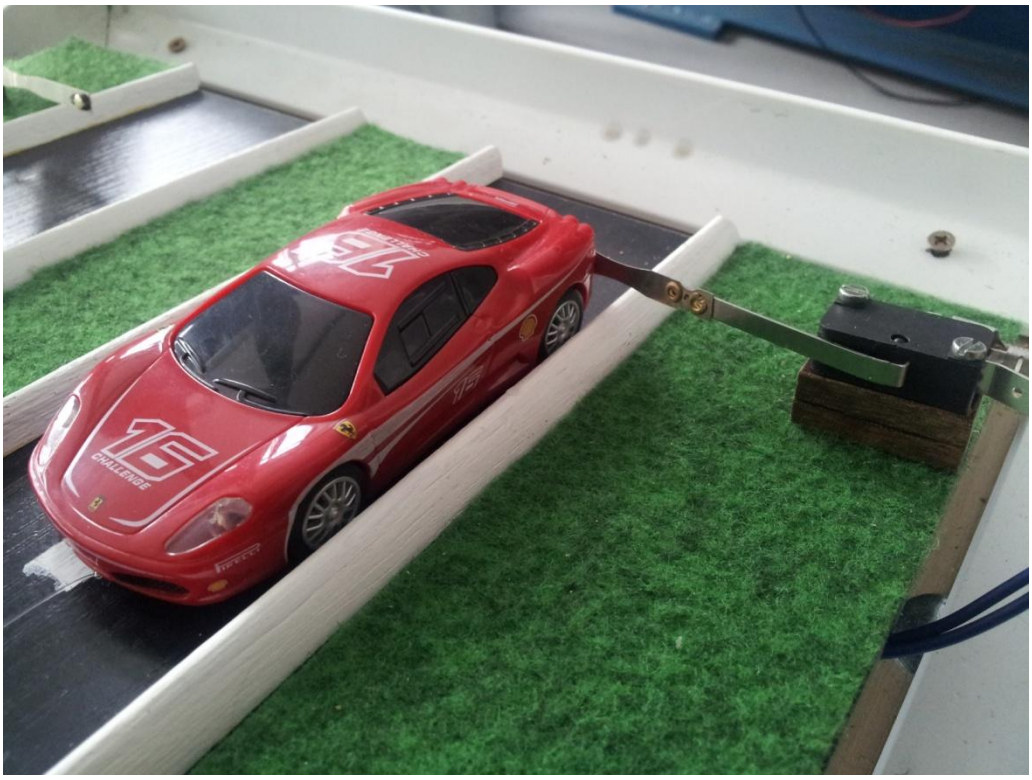
Obr. 3.8 D*isplej*



Obr. 3.9 Zadní strana d*ispleje s převodníkem*

3.6 Koncové spínače

Pro koncové spínače byl použit mikrospínač 5/250VAC s dlouhou páčkou. Protože model autíčka má malou hmotnost, musela být ovládací páčka prodloužena. Tím se snížila potřebná síla na sepnutí mikrospínače. Prodloužená páčka je viditelná na obr. 3.10.



Obr. 3.10 *Koncový spínač s modelem autíčka*

4 Možné vylepšení závodu automobilů

Závod automobilů jsem vylepšil řídící elektronikou a zobrazováním času na displejích. Při návrhu řídící elektroniky jsem počítal s pozdějším vylepšením modelu závodů automobilů. Z tohoto důvodu byl předimenzován stabilizátor. Napětí 5V ze stabilizátoru je možné odebírat z konektoru USB. Na plošném spoji je připravena možnost odebírat napětí stabilizátoru i mimo konektor USB. V případě potřeby lze připájet svorkovnici. Na displejích se zobrazuje čas, za který se autíčko dostane z dolní do horní polohy nakloněné roviny.

Mezi vylepšení modelu závodu automobilů, které by mohlo být v budoucnu realizováno, je volba závodních režimů. Uživatel by pomocí tlačítek a displejů zvolil režim jízdy. Mezi režimy jízdy by bylo možné vybrat z několika možností. Jeden z režimů by byl navržen tak, aby jedno z autíček bylo automaticky ovládáno pomocí Arduina UNO. Druhé autíčko by ovládal uživatel pomocí tlačítek.

Dalším vylepšením by byl závodní semafor, který by odstartoval závod. Tlačítka by byla blokována do doby, než se závod odstartuje. K odblokování tlačítek by došlo, až by na semaforu zhasla dioda značící start závodu.

K další možnosti vylepšení modelu závodu automobilů by patřila komunikace mezi deskami Arduino UNO. Dosažené časy by se porovnávaly. Na displejích by se zobrazil vítěz a dosažený čas. Tyto údaje by se ukládaly do paměti a porovnávaly by se s nejlepším časem, který byl dosáhnut.

Mezi konstrukční vylepšení by mohlo být vylepšení krokového motoru. Vhodným zmenšením vzduchové mezery nebo přidáním ocelových plechů na statorové cívky by se dosáhlo zvýšení kroutícího momentu.

Model závodu automobilů má především prezentovat funkci krokového motoru. Další vylepšení by spočívalo v tom, že by se mohl zvolit režim řízení krokového motoru. Mohlo by se vybrat mezi jednofázovým unipolárním řízením a mezi dvoufázovým unipolárním řízením. Případně by každý z motorů měl jiný režim řízení.

Závěr

Cílem této bakalářské práce bylo vylepšit původní model závodu automobilů. Původní model byl ovládán spínáním stabilizovaného zdroje napětí přes tlačítka. Při stisku tlačítek se přímo napájely jednotlivé statorové cívky. Nevýhodou bylo, že krokový motor nedržel po uvolnění tlačítek sepnuté cívky.

Provedl jsem vylepšení spočívající v ovládání krokového motoru řídícím programem Arduina. Autíčka se po spuštění automaticky nastaví do startovací pozice. Model závodu automobilů byl také vylepšen o zobrazování času na displeje. Po startu je měřen čas.

Během realizace praktické části této bakalářské práce jsem se seznámil s profesionálním softwarem určeným pro návrhy desek plošných spojů, poznal proces výroby plošných spojů. Naučil se základy programování Arduina v programovém prostředí a prakticky si ověřil funkci a možnosti řízení krokového motoru.

V první kapitole je popsán mikrokontroler Arduino UNO a jeho vývojové prostředí IDE. Následně jsou uvedeny základní vlastnosti dalších oficiálních desek Arduino, které jsou dostupné na trhu.

Ve druhé kapitole jsou uvedeny základní druhy krokových motorů. Jsou zde uvedeny možnosti řízení krokových motorů. Výhody a nevýhody jednotlivých druhů řízení a přehledné tabulky spínaných cívek. Je zde uvedeno jednofázové unipolární řízení, které je využito v praktické části této bakalářské práce.

Třetí kapitola se zabývá praktickou částí této bakalářské práce. Popisuje vytvořenou desku plošného spoje a použitou řídící elektroniku. Kapitola také obsahuje popis funkce celého modelu závodů automobilů.

Ve čtvrté kapitole jsou uvedeny možnosti vylepšení modelu závodů automobilů.

Seznam literatury a informačních zdrojů

[1] VODA, Zbyšek. *Průvodce světem Arduina*. Bučovice: Martin Stříž, 2015. ISBN 978-80-87106-90-7.

[2] *What is Arduino* [online]. 2016 [cit. 2016-04-22]. Dostupné z: *https://www.arduino.cc/en/Guide/Introduction*

[3] *Arduino UNO* [online]. 2016 [cit. 2016-04-22]. Dostupné z: https://www.arduino.cc/en/Main/ArduinoBoardUno

[4] VODA, Zbyšek. *Arduino Linux-Software* [online]. 2016 [cit. 2016-05-30]. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=1881

[5] VODA, Zbyšek. *Programujeme arduino* [online]. 2016 [cit. 2016-05-30]. Dostupné z: http://arduino.cz/programujeme-arduino/

[6] *Datasheet LM2576T*-5 [online]. 2016 [cit. 2016-05-30]. Dostupné z: http://www.datasheetdir.com/LM2576T-ADJ+Switching-Regulators

[7] *Sprint-Layout* [online]. 2016 [cit. 2016-05-30]. Dostupné z: http://www.elvo-plzen.cz/layout.html

[8] *Arduino Leonardo* [online]. 2016 [cit. 2016-05-30]. Dostupné z: http://www.hwkitchen.com/products/arduino-leonardo-with-headers-/

[9] *ATmega168/328-Arduino Pin Mapping* [online]. 2016 [cit. 2016-05-30]. Dostupné z: https://www.arduino.cc/en/Hacking/PinMapping168

[10] BARTOŠ,V., ČERVENÝ,J., HRUŠKA, J., KOTLANOVÁ,A., SKALA,B.: *Elektrické stroje.* Vydavatelství ZČU v Plzni, 2006

[11] ŘEZÁČ, Kamil. *Krokové motory* [online]. 2016 [cit. 2016-05-30] Dostupné z: https://robotika.cz/articles/steppers/cs

[12] *Pohony s krokovými motorky* [online]. 2016 [cit. 2016-05-30] Dostupné z: http://fei1.vsb.cz/kat410/studium/studijni_materialy/se/cast_C_el_pohony/se_eph_c1_krokac_02_teorie.pdf

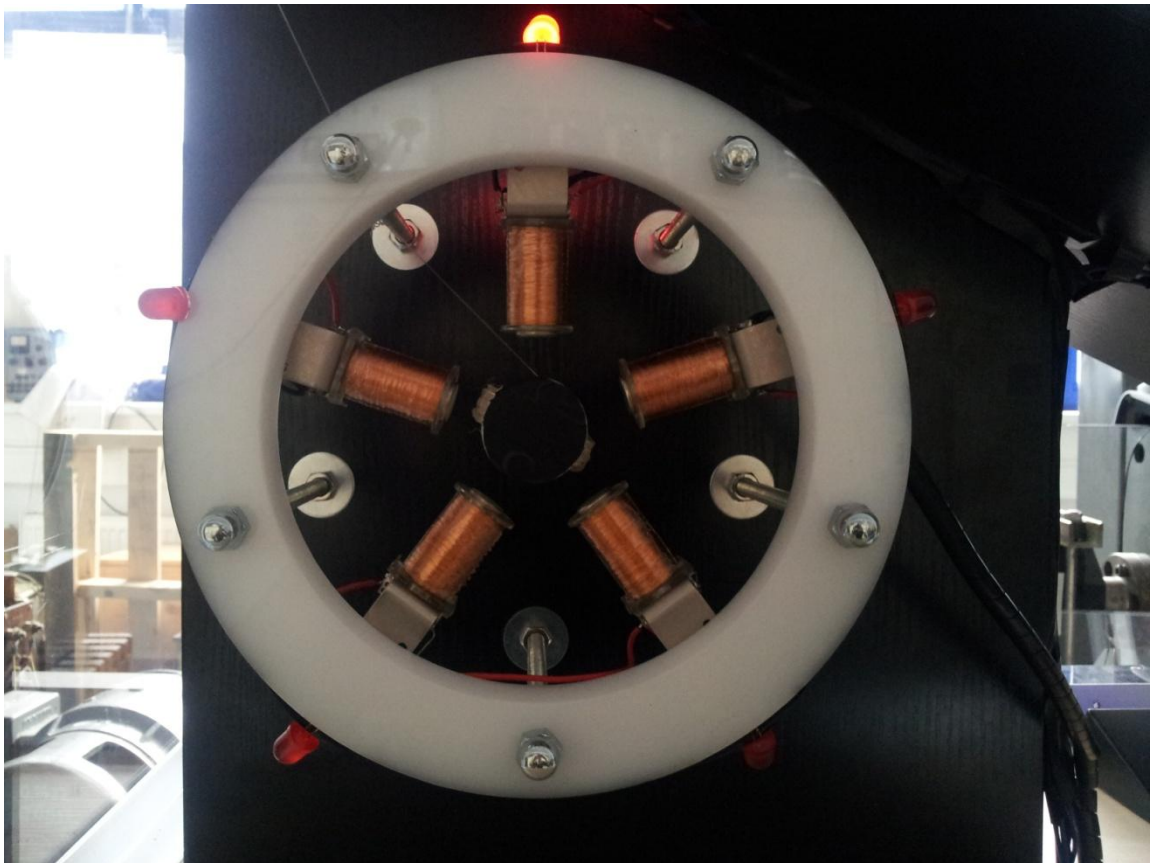
[13] *Mikrokrokování: sny a skutečnost* [online]. 2016 [cit. 2016-05-30] Dostupné z: http://robodoupe.cz/2016/mikrokrokovani-sny-a-skutecnost/

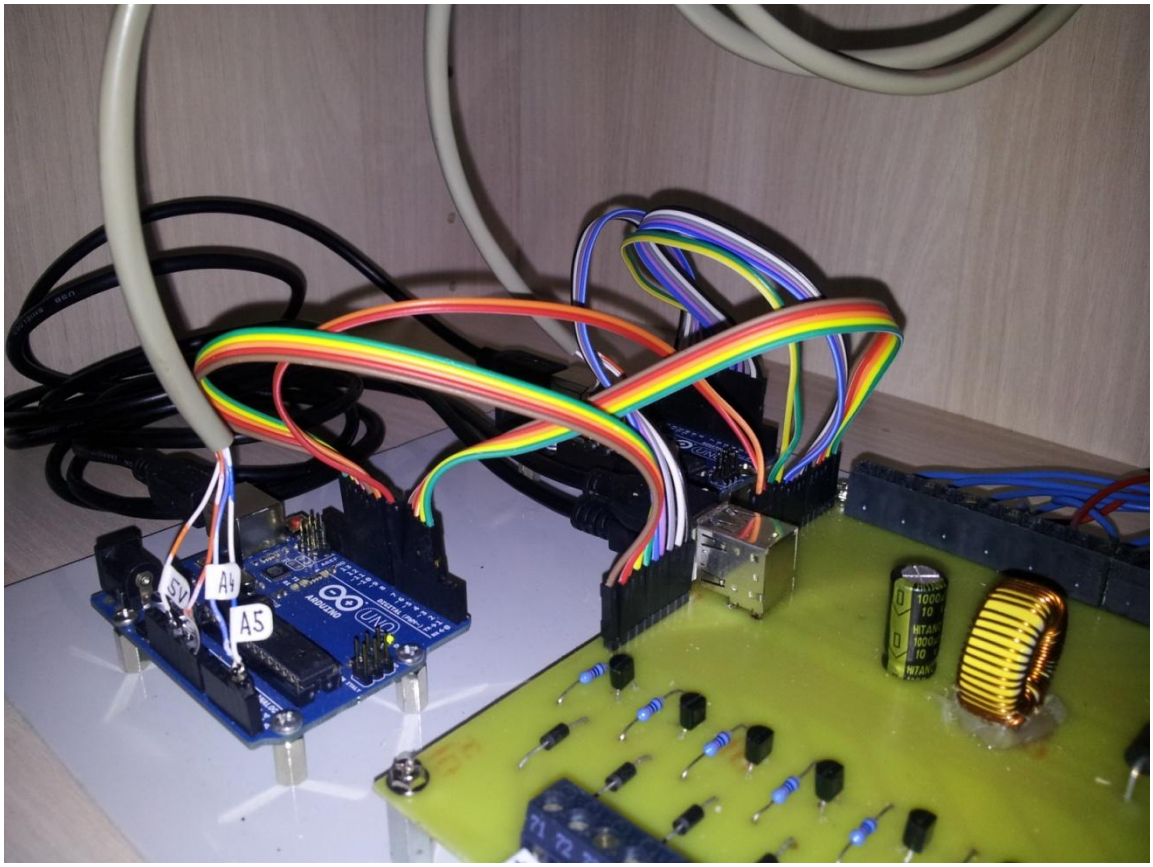
[14] *Druhy krokových motorů* [online]. 2016 [cit. 2016-05-30] http://www.kvd.zcu.cz/cz/materialy/POS/HTML/57/default.htm

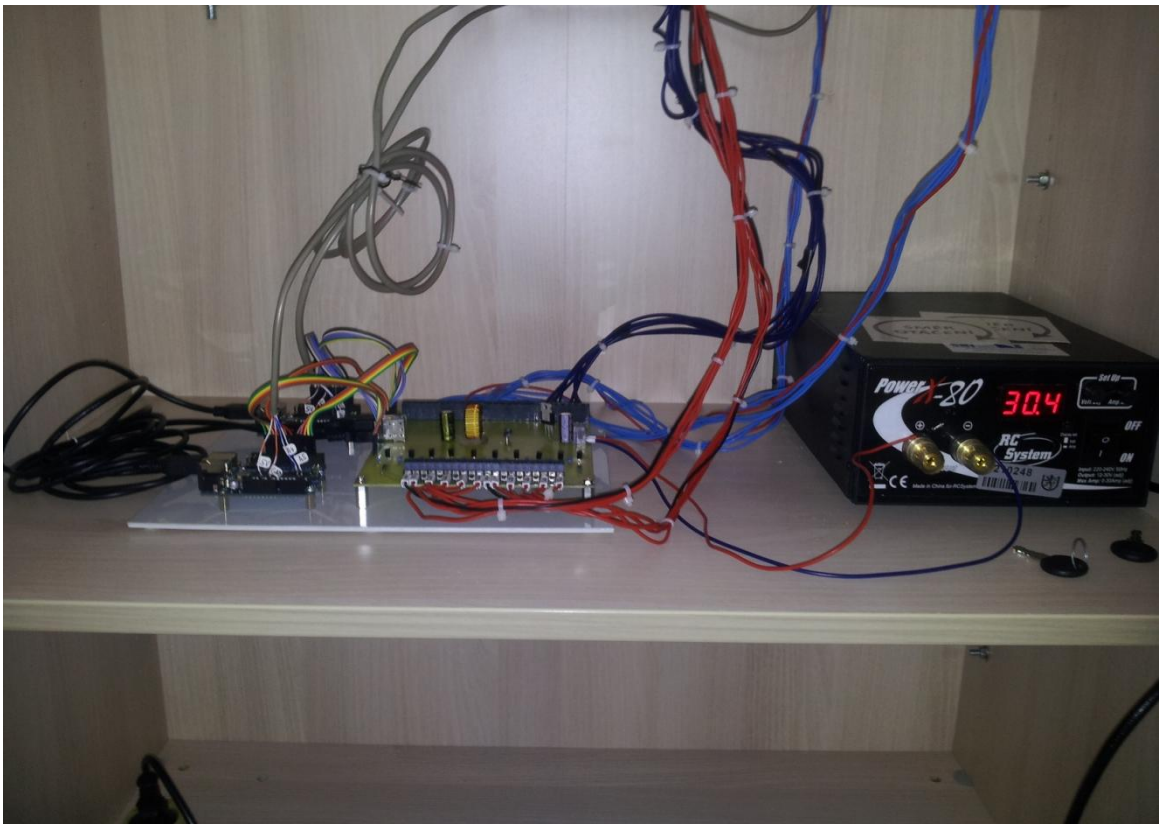
Přílohy

Příloha A – Fotografie

Vas cas:
00:16:8




5V
A4
A5
UNO


30.4
RC System

Příloha B – Program

```
/*
--------------------------------------------------------------------------
Ovládání krokového motoru v modelu závodů automobilů pomocí
mikrokontroleru Arduino
--------------------------------------------------------------------------
  Digital pin:  tlacitka:               2,3,4,5,6
                koncovy spinac dolni:   7
                koncovy spinac horni:   8
                civky:                  9,10,11,12,13

  Display:      SDA
                SCL
                Ucc
                Gnd
--------------------------------------------------------------------------
*/

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

const int civka_1 = 9;
const int civka_2 = 10;
const int civka_3 = 11;
const int civka_4 = 12;
const int civka_5 = 13;

const int tlacitko_1 = 2;
const int tlacitko_2 = 3;
const int tlacitko_3 = 4;
const int tlacitko_4 = 5;
const int tlacitko_5 = 6;
const int koncak_dolni = 7;
const int koncak_horni = 8;

int tlacitko=1;
int poc=0;
int krok=0;
int e=0;
boolean run = false;

long timer = 0;
int stavtlacitka;
int stavtlacitka1;
int stavtlacitka2;
int stavtlacitka3;
int stavtlacitka4;
int stavtlacitka5;
int stavkoncaku;
int koncak_dolni_novystav = HIGH;
int koncak_dolni_poslednistav = HIGH;
int koncak_horni_novystav = HIGH;
int koncak_horni_poslednistav = HIGH;
int sekundy = 0;
int minuty = 0;
int desetiny = 0;

void setup(){
```

```
pinMode(civka_1, OUTPUT);
pinMode(civka_2, OUTPUT);
pinMode(civka_3, OUTPUT);
pinMode(civka_4, OUTPUT);
pinMode(civka_5, OUTPUT);

pinMode(tlacitko_1, INPUT_PULLUP);
pinMode(tlacitko_2, INPUT_PULLUP);
pinMode(tlacitko_3, INPUT_PULLUP);
pinMode(tlacitko_4, INPUT_PULLUP);
pinMode(tlacitko_5, INPUT_PULLUP);
pinMode(koncak_dolni, INPUT_PULLUP);
pinMode(koncak_horni, INPUT_PULLUP);


//=========================================================

  Serial.begin(9600);
  lcd.begin(16,2);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("  ZAPADOCESKA");
  lcd.setCursor(0,1);
  lcd.print("   UNIVERZITA");
  delay(3000);
  lcd.clear();
  lcd.print("    Fakulta");
  lcd.setCursor(0,1);
  lcd.print("Elektrotechnicka");
  delay(3000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Vas cas:");
  lcd.setCursor(0,1);
  lcd.print("00:00:0");
  do{
  stavkoncaku = digitalRead(koncak_dolni);
  jeddolu();
  }while (stavkoncaku==HIGH);
  }

//============================================================
// PODPROGRAMY
//------------------------------------------------------------

/*
 * Jizda nahoru
 */
void jednahoru () {

stavtlacitka = digitalRead(tlacitko_1);
    if (stavtlacitka == LOW && tlacitko==1){
      digitalWrite(civka_1, HIGH);
      zpozdeni();
      digitalWrite(civka_2, LOW);
      digitalWrite(civka_3, LOW);
      digitalWrite(civka_4, LOW);
      digitalWrite(civka_5, LOW);
      tlacitko=2;
    }
```

```
stavtlacitka = digitalRead(tlacitko_2);
    if (stavtlacitka == LOW && tlacitko==2){
      digitalWrite(civka_2, HIGH);
      zpozdeni();
      digitalWrite(civka_1, LOW);
      digitalWrite(civka_3, LOW);
      digitalWrite(civka_4, LOW);
      digitalWrite(civka_5, LOW);
      tlacitko=3;
    }

  stavtlacitka = digitalRead(tlacitko_3);
    if (stavtlacitka == LOW && tlacitko==3){
      digitalWrite(civka_3, HIGH);
      zpozdeni();
      digitalWrite(civka_1, LOW);
      digitalWrite(civka_2, LOW);
      digitalWrite(civka_4, LOW);
      digitalWrite(civka_5, LOW);
      tlacitko=4;
    }

   stavtlacitka = digitalRead(tlacitko_4);
    if (stavtlacitka == LOW && tlacitko==4){
      digitalWrite(civka_4, HIGH);
      zpozdeni();
      digitalWrite(civka_1, LOW);
      digitalWrite(civka_2, LOW);
      digitalWrite(civka_3, LOW);
      digitalWrite(civka_5, LOW);
      tlacitko=5;
    }

  stavtlacitka = digitalRead(tlacitko_5);
    if (stavtlacitka == LOW && tlacitko==5){
      digitalWrite(civka_5, HIGH);
      zpozdeni();
      digitalWrite(civka_1, LOW);
      digitalWrite(civka_2, LOW);
      digitalWrite(civka_3, LOW);
      digitalWrite(civka_4, LOW);
      tlacitko=1;
    }
}
//-------------------------------------------------------------
/*
 * Jizda dolu
 */
void jeddolu() {
      e=1;
      stavkoncaku = digitalRead(koncak_dolni);
   if (stavkoncaku == HIGH && krok==0){
      zpozdeni();
      digitalWrite(civka_1, LOW);
      digitalWrite(civka_2, LOW);
      digitalWrite(civka_3, LOW);
      digitalWrite(civka_4, LOW);
      digitalWrite(civka_5, HIGH);
      krok=1;
    }
```

```
      stavkoncaku = digitalRead(koncak_dolni);
   if (stavkoncaku == HIGH && krok==1){
      zpozdeni();
      digitalWrite(civka_1, LOW);
      digitalWrite(civka_2, LOW);
      digitalWrite(civka_3, LOW);
      digitalWrite(civka_4, HIGH);
      digitalWrite(civka_5, LOW);
      krok=2;
    }

      stavkoncaku = digitalRead(koncak_dolni);
   if (stavkoncaku == HIGH && krok==2){
      zpozdeni();
      digitalWrite(civka_1, LOW);
      digitalWrite(civka_2, LOW);
      digitalWrite(civka_3, HIGH);
      digitalWrite(civka_4, LOW);
      digitalWrite(civka_5, LOW);
      krok=3;
    }

      stavkoncaku = digitalRead(koncak_dolni);
   if (stavkoncaku == HIGH && krok==3){
      zpozdeni();
      digitalWrite(civka_1, LOW);
      digitalWrite(civka_2, HIGH);
      digitalWrite(civka_3, LOW);
      digitalWrite(civka_4, LOW);
      digitalWrite(civka_5, LOW);
      krok=4;
    }

      stavkoncaku = digitalRead(koncak_dolni);
   if (stavkoncaku == HIGH && krok==4){
      zpozdeni();
      digitalWrite(civka_1, HIGH);
      digitalWrite(civka_2, LOW);
      digitalWrite(civka_3, LOW);
      digitalWrite(civka_4, LOW);
      digitalWrite(civka_5, LOW);
      krok=0;
    }
}

//---------------------------------------------------------------------------

/*
 * Kontrola dolniho koncoveho spinace
 */

void checkkoncakdolni() {
  if(koncak_dolni_novystav != koncak_dolni_poslednistav) {
  if(koncak_dolni_novystav == HIGH ) {
 smazvse();
  run =true;
  }
  }

  koncak_dolni_poslednistav = koncak_dolni_novystav;
```

```
  }

//--------------------------------------------------------------------------
/*
 *   Kontrola horniho koncoveho spinace
 */

void checkkoncakhorni() {
  if(koncak_horni_novystav != koncak_horni_poslednistav) {

  if(koncak_horni_novystav == LOW ) {
   run=false;
   delay(5000);

 stavkoncaku = digitalRead(koncak_dolni);
    if (stavkoncaku == HIGH ){
      tlacitko=1;
      e=0;
      do{


      jeddolu();
      }while (stavkoncaku==HIGH);
  }
    }
  }
}

//--------------------------------------------------------------------------
 /*
 * Pro každou desetinu sekundy vyvola podprogram tikni()
 */

void hodiny() {
  Serial.println(millis() /100);
  if((timer - millis()/100) >= 100 || timer == 0) {
    tikni();
    timer = millis()/100;
  }
}

/*
 * Zobrazovani casu na displeji
 */

void tikni() {

  if(run) {
    aktualizujLCD();

    if(desetiny == 9) {
      desetiny = 0;

      if(sekundy == 59) {
        sekundy = 0;
        minuty++;

      } else {
        sekundy++;
      }
    } else {
```

```
    desetiny++;
    }
  }
}
//---------------------------------------------------------------
/*
 * Aktualizace dat displeje
 */

void aktualizujLCD() {
  lcd.setCursor(0,1);
  if(minuty < 10) {
    lcd.print("0");
  }
  lcd.print(minuty, DEC);
  lcd.print(":");

  if(sekundy < 10) {
    lcd.print("0");
  }
  lcd.print(sekundy, DEC);
  lcd.print(":");

  lcd.print(desetiny, DEC);

}
//---------------------------------------------------------------

/*
 * Vymazani displeje a promenych do vychozich hodnot
 */

void smazvse() {
  run = false;
  sekundy = 0;
  minuty = 0;
  desetiny = 0;
  aktualizujLCD();

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Vas cas:");
  lcd.setCursor(0,1);
  lcd.print("00:00:0");
}
//---------------------------------------------------------------

/*
 * Zpozdeni mezi spinanim civek pomoci aktualizace displeje
 */

void zpozdeni() {
 for(poc=0;poc<50;poc++){
    hodiny();
    }
      poc=0;
}

//=============================================
//=============================================
/*
```

```
 * Hlavni smycka programu
 */
void loop() {

  if (e==0){
  jeddolu();}
  hodiny();  //start hodin
  koncak_dolni_novystav = digitalRead(koncak_dolni);
  koncak_horni_novystav = digitalRead(koncak_horni);
  checkkoncakhorni();
  checkkoncakdolni();
  jednahoru();

}
```

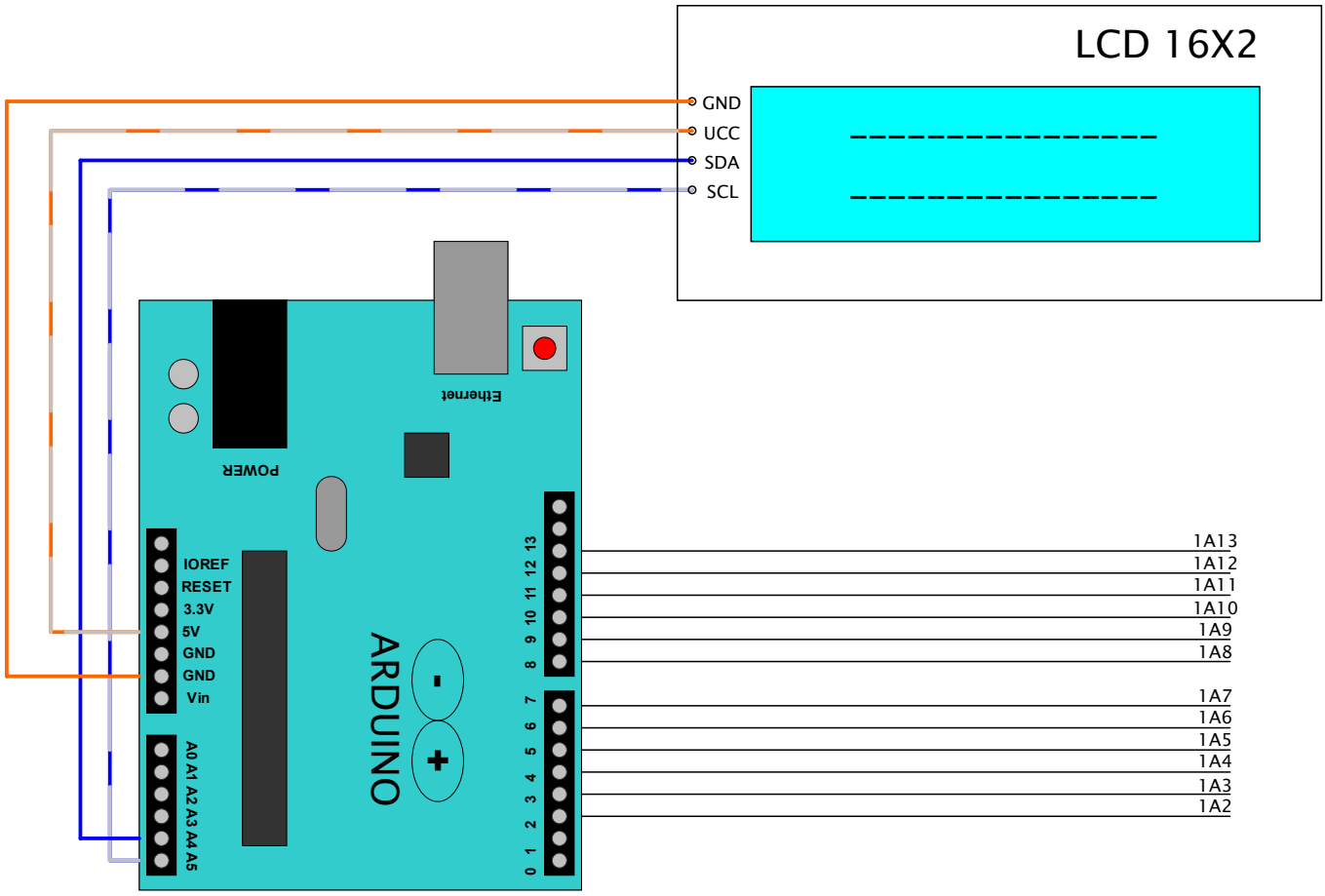
Příloha C – Dokumentace zapojení

Schéma zapojení

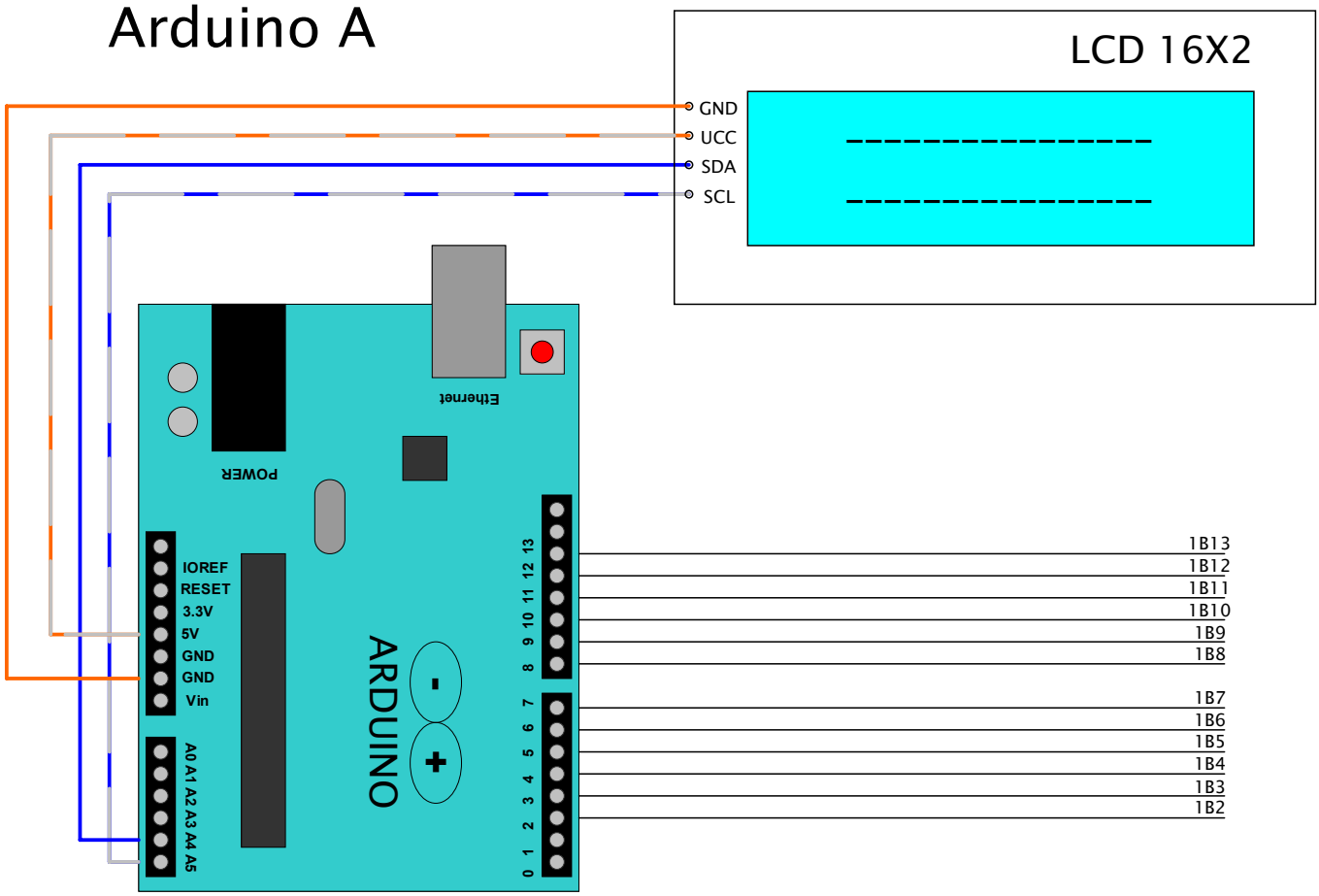
Ovládání krokového motoru v modelu závodů automobilů pomocí mikrokontroleru Arduino

Václav Frélich

2016

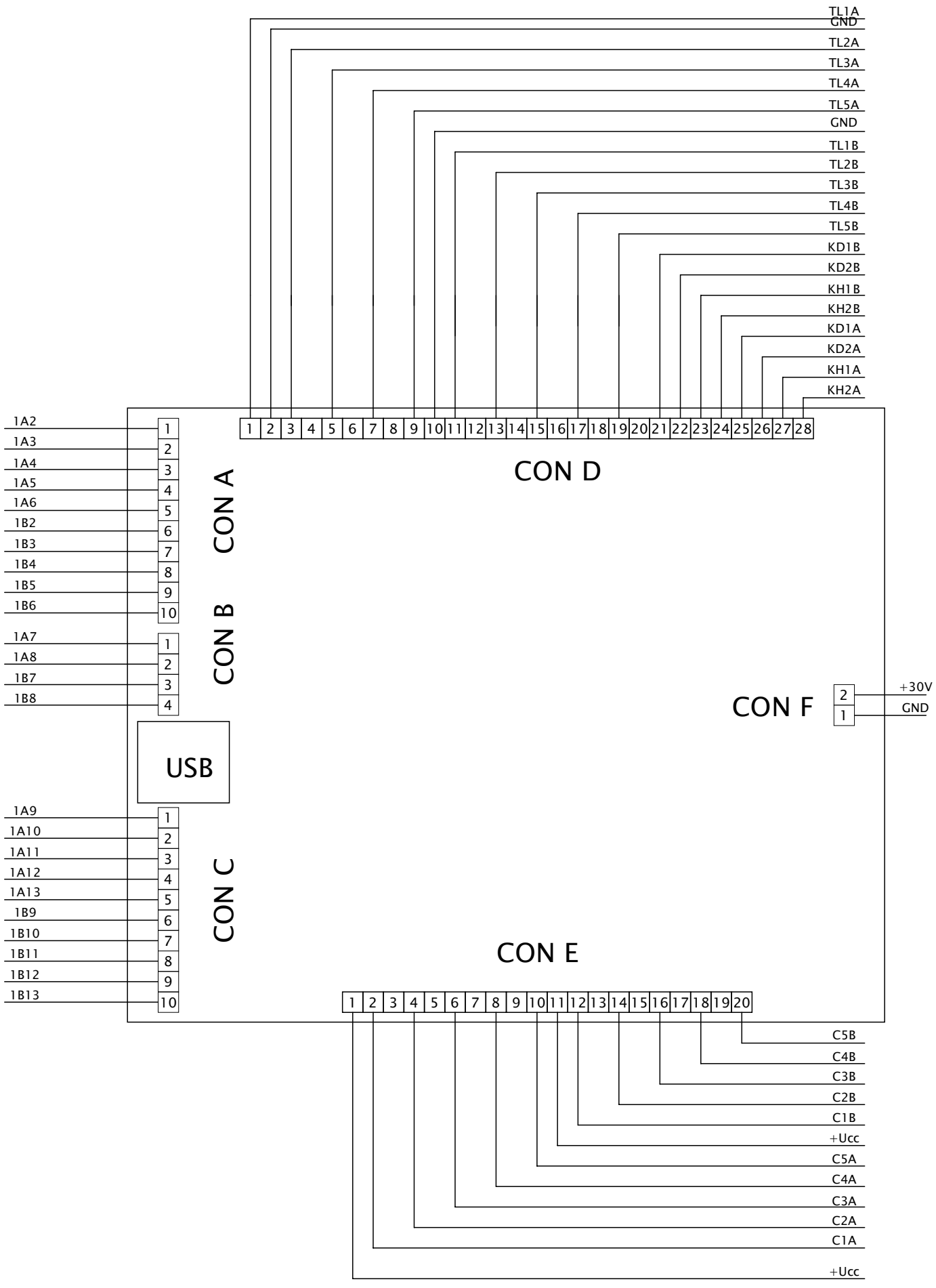


Arduino A

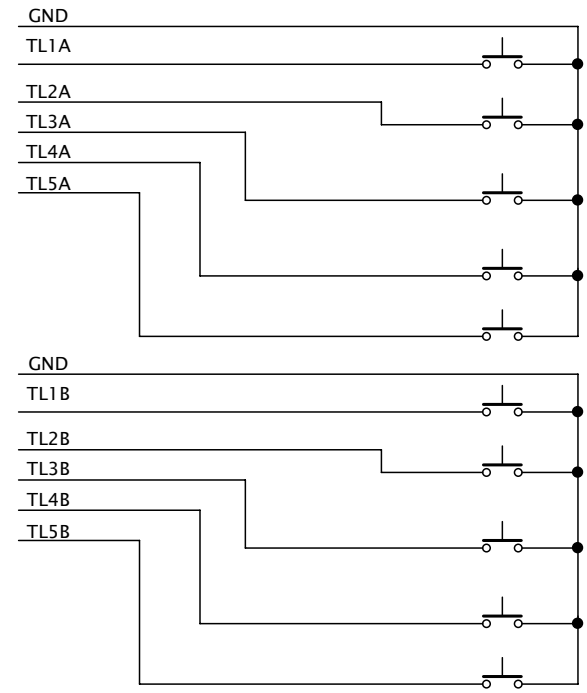


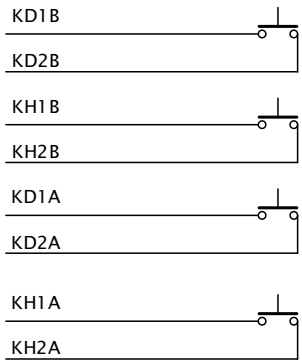
Arduino B

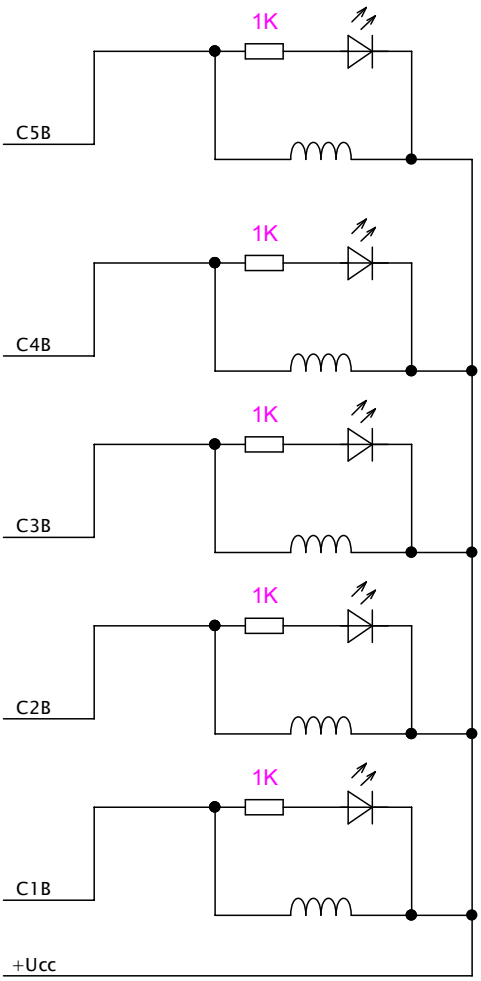
Název: Ovládání krokového motoru v modelu závodů automobilů pomocí mikrokontroleru Arduino		
Index:	Datum: 21.5.2016	List: 1/3
Autor: Václav Frélich		
Revize:		

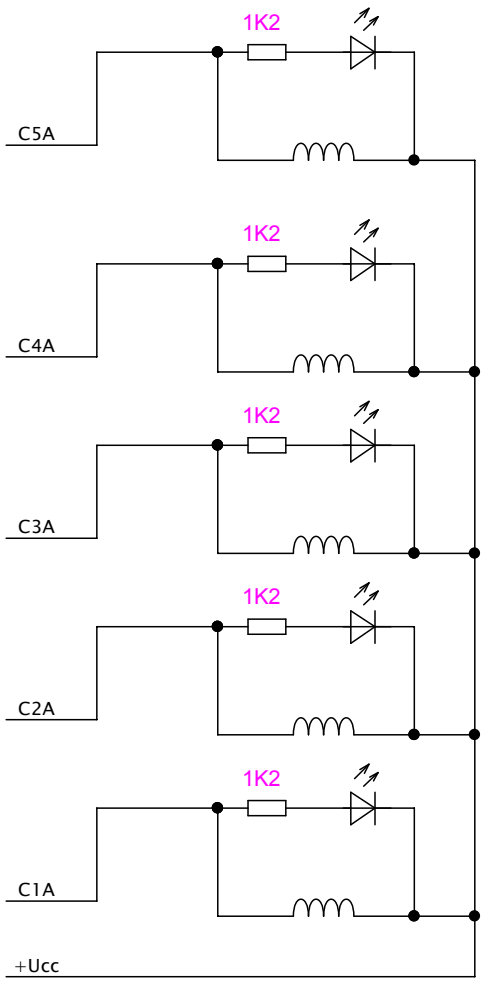


Název: Ovládání krokového motoru v modelu závodů automobilů pomocí mikrokontroleru Arduino		
Index:	Datum: 21.5.2016	List: 2/3
Autor: Václav Frélich		
Revize:		









Název: Ovládání krokového motoru v modelu závodů automobilů pomocí mikrokontroleru Arduino		
Index:	Datum: 21.5.2016	List: 3/3
Autor: Václav Frélich		
Revize:		

Příloha D – Datový nosič CD

Datový nosič CD obsahuje: bakalářskou práci, vytvořený program pro mikrokontroler Arduino UNO včetně použitých knihoven, schéma zapojení, navržený plošný spoj v měřítku 1:1