

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2017

Miroslav Veselý



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

**BRNO UNIVERSITY OF TECHNOLOGY**

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

**FACULTY OF INFORMATION TECHNOLOGY**

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

**DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA**

**SYSTÉM SE SERVY ŘÍZENÝMI  
MIKROPOČÍTAČEM**

**SYSTEM WITH SERVOES CONTROLLED BY MICROCOMPUTER**

**BAKALÁŘSKÁ PRÁCE**

**BACHELOR'S THESIS**

**AUTOR PRÁCE**

**AUTHOR**

**MIROSLAV VESELÝ**

**VEDOUCÍ PRÁCE**

**SUPERVISOR**

**prof. Dr. Ing. PAVEL ZEMČÍK**

**BRNO 2017**

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií

Akademický rok 2016/2017

**Zadání bakalářské práce**

Řešitel: **Veselý Miroslav**

Obor: Informační technologie

Téma: **Systém se servy řízenými mikropočítačem**

**System with Servos Controlled by Microcomputer**

Kategorie: Uživatelská rozhraní

**Pokyny:**

1. Prostudujte způsob řízení servosystémů mikropočítačem, například embedded počítačem, PC, nebo mobilním zařízením. Zaměřte se na modelářské a jim podobné servosystémy.
2. Navrhněte jednoduchý přípravek, například pro nastavování vybraného optického zařízení a navrhněte způsob jeho ovládání.
3. Popište a diskutujte možnosti ovládání přípravku a jeho dosažitelné vlastnosti.
4. Navržený přípravek implementujte a demonstруйте jeho funkčnost v zamýšlené aplikaci.
5. Diskutujte dosažené výsledky a možnosti pokračování práce.

**Literatura:**

- Dle pokynů vedoucích

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 zadání

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

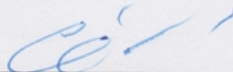
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Zemčík Pavel, prof. Dr. Ing.,** UPGM FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
612 66 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký  
vedoucí ústavu

## **Abstrakt**

Cílem této bakalářské práce bylo prozkoumání možností řízení modelářských serv mikropočítačem. Součástí práce bylo vytvoření zařízení pro procházení vzdálené plochy laserem. Vytvořené zařízení je postaveno na platformě Arduino a je možné jej vzdáleně ovládat pomocí počítačové aplikace.

## **Abstract**

The goal of this bachelor thesis was to investigate possibilities of controlling RC servos by a microcomputer. A part of this thesis was creation of a device for purpose of browsing distant surface by a laser. The created device is based on Arduino platform and it is possible to control this device remotely by a computer application.

## **Klíčová slova**

laser, servo, Arduino, mikropočítač, řízení, procházení, detekce

## **Keywords**

laser, servo, Arduino, microcomputer, control, browsing, detection

## **Citace**

VESELÝ, Miroslav. Systém se servy řízenými mikropočítačem. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Zemčík Pavel.

# **System se servy řízenými mikropočítačem**

## **Prohlášení**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana prof. Dr. Ing. Pavla Zemčíka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Miroslav Veselý  
17. května 2017

## **Poděkování**

Děkuji vedoucímu práce prof. Dr. Ing. Pavlu Zemčíkovi za poskytování cenných rad při tvorbě této práce a panu Tomáši Zemčíkovi za spolupráci při testování zařízení. Dále děkuji své rodině za podporu při tvorbě této práce.

# Obsah

Obsah.....	1
1 Úvod.....	2
2 Shrnutí současného stavu.....	3
2.1 Existující řešení.....	3
2.2 Platforma Arduino.....	3
2.3 Arduino Nano.....	9
2.4 Windows API.....	12
2.5 Windows Presentation Foundation.....	15
2.6 Laser.....	17
2.7 Modelářské servo.....	21
3 Současný stav a plán práce.....	23
3.1 Požadavky na zařízení a software.....	24
4 Popis práce.....	25
4.1 Konstrukce zařízení.....	26
4.2 Návrh softwaru.....	29
4.3 Testování a využití.....	38
5 Závěr.....	41
Literatura.....	42
Seznam příloh.....	44

# 1 Úvod

Cílem této práce je navrzení a vytvoření zařízení, které pomůže při zjišťování pozice na vzdálené ploše, na kterou je zaměřený detektor laserového paprsku v jiném zařízení. Pro vyvolání odezvy tohoto detektoru provede vytvářené zařízení průchod laserovým paprskem přes vzdálenou plochu.

V některých aplikacích může být požadováno zaměřování detektorů laserových paprsků na uživatelem vybraná místa. Detektor musí být zaměřený na konkrétní místo na vzdálené ploše, přičemž v aplikaci není dostupná možnost zjištění zaměření senzoru, avšak aplikace nabízí možnost úprav polohy senzoru. Určení pozice zaměření senzoru na vzdálené ploše je tedy nutné provést externím aplikací. Tato aplikace by měla zahrnovat samotný přístroj či přístroje, které budou použity k nalezení zaměření senzoru a dále softwarovou část, která bude použita uživateli k nastavování a ovládání tohoto zařízení. Softwarová část aplikace by měla pomáhat při automatizaci procesu zaměřování detektoru a celá aplikace by poté měla být snadno použitelná a reprodukovatelná při zachování nízké ceny.

Vytvářené zařízení by mělo sestávat z řídicího a komunikačního modulu. Tento modul bude ovládat celé zařízení a starat se o komunikaci s počítačovou aplikací. Pro vyzařování úzkého světelného paprsku je vhodný laserový modul. Pohyb laserovým modulem by mohla obstarat dvě modelářská serva.

Tato práce se bude zabývat problematikou návrhu a konstrukce zařízení, které bude pracovat výše uvedeným způsobem. Dále bude v práci rozebrán příslušný software, který bude nahrán v řídicím mikroprocesoru zařízení a software, který bude fungovat pro ovládání zařízení z počítače. V závěru práce budou zhodnoceno testování vytvořeného zařízení a navržen další postup pro případná vylepšení či změny v zařízení a softwaru.

Práce je zajímavá, protože kombinuje proces návrhu a tvorby fyzického zařízení a softwaru pro ovládání tohoto zařízení. Spojuje tak dvě oblasti, elektrotechniku a informační technologie, o které se zajímám.

## 2 Shrnutí současného stavu

V této kapitole budou rozebrány použité technologie hardwaru a softwaru. Jsou zde uvedeny informace o použité platformě Arduino a jejích vlastnostech, dále jsou zde uvedeny informace o aplikačním rozhraní Windows API a Microsoft .NET Framework. Z hardwaru jsou zde rozebrány principy laserových diod a modelářských serv. Kapitola neobsahuje veškeré informace o dané problematice, ale poskytuje pohled na možnosti použitých technologií.

### 2.1 Existující řešení

Zařízení, které by umožňovalo procházet vzdálenou plochu laserovým paprsek podle proměnného nastavení, které zadává uživatel, a které poskytuje sadu potřebných programů pro nastavování a řízení jsem v katalogích výrobců nenašel. Samozřejmě je možné, že podobné zařízení vyrábí firmy se zaměřením na automatizaci pouze na zakázku a na míru zákazníkovi. Avšak zařízení vyráběná na míru zákazníkovi bývají obvykle drahá a vnitřní řešení produktu je zákazníkovi často skryté.

Na druhou stranu je možné na internetu nalézt mnoho amatérsky vyráběných zařízení, včetně návodů na výrobu, která právě pomocí modelářského serva řídí například modul laseru. Obecně jde ale o zařízení, která mají například vykreslovat konkrétní obrazec na cílovou plochu a smysl těchto zařízení je především zábavní. Neumožňují tak proměnné nastavení uživatelem a autoři neposkytují rozhraní pro vzdálenou práci se zařízením. Taková zařízení proto zcela nesplňují potřebu, která je uvedena v úvodu práce.

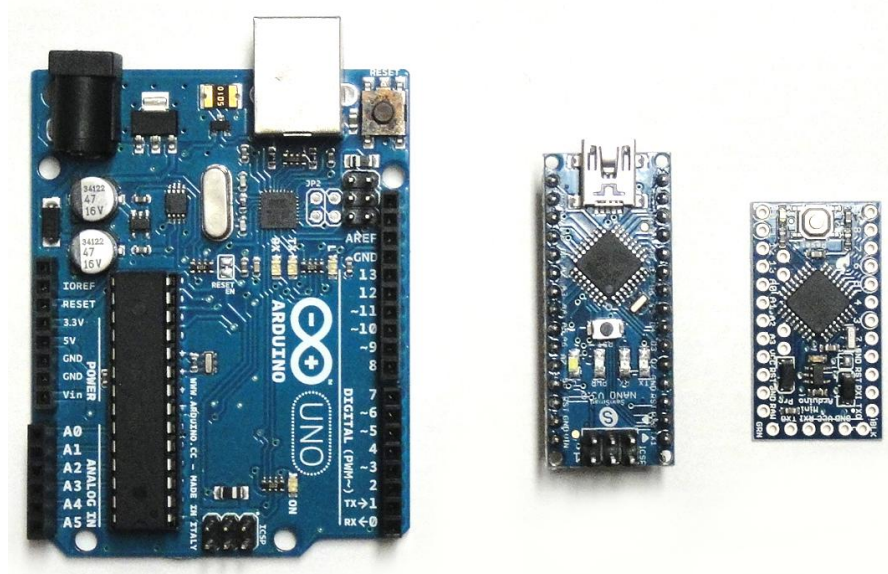
### 2.2 Platforma Arduino

Projekt je založený na elektronické platformě Arduino [1] a [2]. Tato platforma poskytuje otevřený hardware i software zaměřený na segment vestavěných mikropočítačů. Projekt Arduino byl představen v roce 2005 a zjednodušuje přístup k programování mikropočítačů a jejich periférií.

#### 2.2.1 Hardware

Hardware platformy Arduino se skládá z mikrokontroléru a podpůrných, rozšiřujících a napájecích obvodů. Konstrukce desek této platformy se liší podle verze, typu a výrobce, v zásadě ale většina desek poskytuje sadu napájecích obvodů, kterými je možné desku, zařízení na desce a periferie napájet. Mikrokontrolér platformy je programovatelný zvnějšku, typicky připojením vývojové desky k počítači přes rozhraní USB a naprogramováním mikrokontroléru z vývojového prostředí nebo lze použít technologii ISP (In-system programming) a pomocí šesti pinů na vývojové desce mikrokontrolér programovat a ladit. Deska často obsahuje převodníkový čip USB na RS-232 a je vytvořena virtuální sériová linka pro komunikaci mezi Arduino a počítačem. K některým verzím desek je třeba připojit externí převodník USB na RS-232 a programovat zařízení přes něj.





Obrázek 2.1: Srovnání desek platformy Arduino [18]

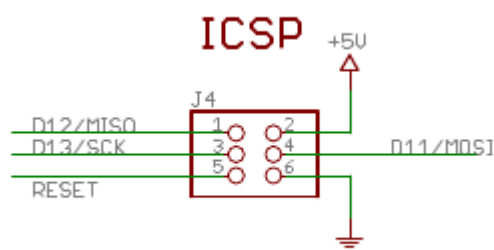
Nejdůležitější částí desky je programovatelný mikrokontrolér. V případě originálních desek Arduino jsou používány čipy od firmy Atmel [4]. V závislosti na typu desky jde o čipy s různou konfigurací a odlišnými pracovními frekvencemi, většinou v rozsahu jednotek až desítek MHz. Tyto čipy jsou postaveny na architektuře AVR RISC nebo ARM RISC s šířkou slova 8, 16 nebo 32 bitů. Operační paměť čipu je typu SRAM a její velikost se pohybuje v jednotkách až desítkách kilobajtů. Nevolatilní paměť programu je pak typu Flash a její velikost se pohybuje v jednotkách až stovkách kilobajtů. Čipy firmy Atmel dále obsahují druhou nevolatilní paměť typu EEPROM, velikost této paměti se pohybuje v řádech stovek až tisíců bajtů.

Rozšiřující periferie je možné k desce připojit přes vyvedené 2,54mm piny. V závislosti na typu desky se liší počet a typ vyvedených vstupně-výstupních pinů. Obecně ale Arduino poskytuje vyvedené digitální piny pro připojení digitálních periférií. A dále analogové piny s možností PWM (pulse width modulation) a A/D převodníkem. Některé dostupné piny jsou víceúčelové a poskytují možnost například běžné digitální komunikace s periférií nebo umožňují specializované využití, například pro SPI komunikaci. K platformě vzniklo mnoho rozšiřujících desek, které umožňují jednoduché připojení nových periférií k původní desce. Uvést lze například rozhraní TFT displeje, komunikační bezdrátová rozhraní, řadiče pro serva a jiné.

Napájení platformy je řešeno přes různá rozhraní, nejčastěji jde o rozhraní USB, 2,1mm jack nebo napájení přes vyvedené piny. Vestavěný napěťový stabilizátor umožňuje použití proměnného napájecího napětí a to typicky v rozsahu 6V až 20V a přizpůsobení na požadovaných 5V nebo 3,3V. Jako zdroj lze použít akumulátor nebo napájecí napětí ze sběrnice USB, například z hostitelského počítače. Proudové zatížení vyvedených pinů je různé a liší se podle typu desky, většinou se jedná o desítky mA na jeden pin.

Velikost desek platformy Arduino se pohybuje v řádech jednotek centimetrů a váha v jednotkách až desítkách gramů, platforma tak umožňuje použití i v rozměrově malých projektech, například v segmentu IoT (Internet of Things).

Zařízení platformy Arduino podporují uvedenou technologii ISP, zvanou také ICSP. Zkratka ISP v anglickém jazyce znamená In - System Programming. Jedná se o techniku pro ladění a programování již zabudovaného mikrokontroléru do vestavěné aplikace. Vzhledem k tomu, že měnit mikrokontrolér nebo část vestavěného zařízení kvůli aktualizaci řídicího programu by bylo neekonomické, je možné mikrokontrolér přeprogramovat nebo ladit přímo ve vestavěné aplikaci pomocí několika vyvedených pinů. V případě mikrokontrolérů firmy Atmel jde o 6 pinové rozhraní, pomocí něhož lze připojit jiné zařízení a komunikovat s ním pomocí sériového protokolu. V případě vývojových desek Arduino je možné pomocí ISP rozhraní obejít nebo naopak nahrát hlavní zavaděč mikrokontroléru a také může sloužit jako druhé rozhraní pro sériovou SPI komunikaci. Rozhraní využívá jeden pin pro externí reset, dva piny pro plně duplexní komunikaci, dva napájecí piny pro vstupní napětí a zemnění a jeden pin pro přenos hodinového signálu.



Obrázek 2.2: Schéma zapojení ISP [2]

Pro potřebu komunikace zařízení s jiným zařízením, typicky s počítačem, je možné využít sériové komunikace na pinech 0 a 1, které jsou zároveň připojeny k převodníku na rozhraní USB sběrnice. Pomocí tohoto rozhraní lze se zařízením vzdáleně komunikovat. Je tak možné implementovat oboustrannou komunikaci, například pro nastavení zařízení a získávání odezvy ze zařízení během operací, přičemž k tomu není zapotřebí další externí modul. K tomuto typu komunikace slouží v aplikačního knihovně platformy objekt *Serial*, pomocí něhož lze na sériové lince, mimo jiné, data zasílat i přijímat.

Podporovaný je standard RS-232 [9] pro sériovou komunikaci mezi dvěma zařízeními, které jsou propojeny minimálně třemi vodiči pro společné zemnění, vysílání a odesílání dat. V případě jednosměrné komunikace je možné využít zapojení se dvěma vodiči. Při použití více vodičů lze využít standardem definované signály pro řízení spojení, například vodič pro indikaci, že zařízení druhé strany je připraveno k přenosu dat. RS-232 definuje přenos dvou signálů, logické 0 a 1. Přičemž napětí jednotlivých stavů je záporné pro logickou 1 a kladné pro logickou 0. Možné je použít napětí v rozsahu  $\pm 3V$  až  $\pm 15V$ .

Komunikace je podle standardu RS-232 asynchronní a nevyžaduje tedy signál pro platnost dat. Rychlost přenosu závisí na kvalitě použitých vodičů, jejich délce a schopnostech koncových zařízení. V zásadě je možné komunikovat na vzdálenosti v řádech metrů až desítek metrů rychlostmi, které se pohybují v desítkách tisíc baudů za sekundu. Přenášená data mohou jako jednoduchou kontrolu dat používat paritní bit. Pro značení konce přenosu bajtu se používají jeden nebo dva stop bity. Délku bajtu v bitech lze definovat, typicky se používá standardních osm bitů na bajt, je ale možné použít například sedm bitů na bajt.

## 2.2.2 Software

K platformě Arduino [1] a [2] je vydáváno otevřené vývojové prostředí Arduino IDE [2] pro programování jednotlivých mikrokontrolérů. Programování těchto mikrokontrolérů probíhá v jazyce C/C++, je tak možné v projektech používat běžně známé konstrukce z těchto programovacích jazyků. S vývojovým prostředím je dodávána i sada rozšiřujících knihoven pro programování periférií. Například je možné pomocí nich ovládat serva, displeje nebo komunikovat přes sériovou linku. Zároveň jsou s prostředím dodávány i knihovny od firmy Atmel, které poskytují přístup vyšších úrovní programového kódu k perifériím na nižší úrovni. Část dodávaných knihoven poskytuje standardní funkce jazyka C pro práci s řetězci, matematické operace a další. Je důležité uvést, že většina mikrokontrolérů na deskách Arduino obsahuje hlavní zavaděč, který automaticky inicializuje mikrokontrolér a jeho periferie při resetu.

Arduino IDE používá pro překládání zdrojových kódů překladač GCC, který je nastavený pro překlad kódu pro čipy Atmel AVR a Atmel ARM [4]. Při linkování knihoven pro čipy Atmel se používají dodávané knihovny této firmy a linkování probíhá proti těmto knihovnám. Zdrojový kód těchto knihoven používá kombinaci jazyka C a vloženého jazyka symbolických adres, přičemž soubory obsahují podmíněná makra pro překlad na základě použité verze mikrokontroléru a optimalizaci výsledného binárního kódu.

Základní dostupné knihovny [3], které jsou dodávány firmou Atmel poskytují také standardní funkce jazyka C. Jde tedy o funkce pro práci s řetězci, matematické operace a další. Mimo to obsahují také knihovny definic datových typů, knihovny pro podporu práce s časem a další. Důležitou dodávanou knihovnou je podpora pro vytváření zavaděče pro mikrokontrolér.

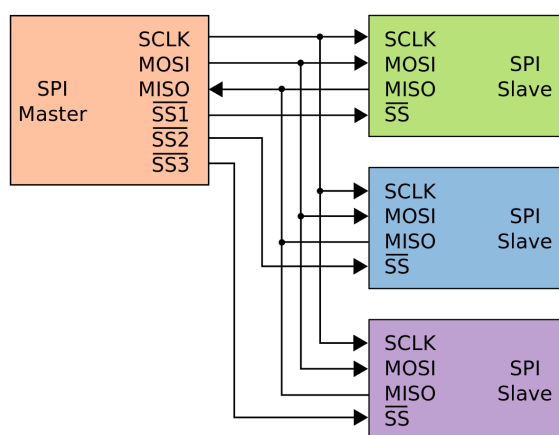
Ve zdrojovém kódu zavaděče je definováno základní nastavení pro inicializaci mikrokontroléru a jeho periférií po resetování. Zavaděč obsahuje také inicializaci komunikace s terminálem při připojení k počítači, nastavení hlídacího obvodu, zakázání maskovatelných přerušení a inicializace počátečních adres paměti na základě použití verze mikrokontroléru.

Verze desek Arduino, které jsou založeny na mikrokontrolérech Atmel AVR obsahují malou paměť EEPROM. Velikost této paměti se pohybuje v řádech stovek bajtů až jednotek kilobajtů. Tento typ paměti slouží pro uchování malého množství dat i po výpadku napájení nebo po resetování mikrokontroléru, kdy je obsah standardní paměti RAM ztracen nebo není definován. Pro práci s touto pamětí je dostupná knihovna, která tvoří obálku nad knihovnou firmy Atmel pro práci s pamětí EEPROM. Celou paměť EEPROM zde zastupuje jeden objekt a pomocí zaslání zpráv tomuto objektu lze pracovat s tímto typem paměti v řídicím programovém kódu mikrokontroléru.

Jako příklad lze uvést metodu pro automatické ukládání dat, která jsou přijata na konkrétním portu zařízení. Pomocí této metody lze Arduino naprogramovat tak, že přijatá data ukládá na určitou adresu v paměti EEPROM a to jen pokud jsou přijatá data z daného portu odlišná od uložených dat na cílové adrese paměti. Toto je vhodné například pro automatické sbírání aktualizovaných dat z čidel v určitých časových intervalech. Důvod pro ukládání pouze odlišných dat je ten, že životnost paměti EEPROM je omezena přibližně na 100 000 zápisových cyklů jedné paměťové buňky. Pomocí pouhého aktualizování dat se tak maximalizuje životnost paměti. Druhým důvodem je časová náročnost operace zápisu do paměti EEPROM, kdy zápis trvá 3,3 milisekundy. Z uvedeného zároveň vyplývá, že je možné šetřit energii minimalizací zápisů do paměti EEPROM.

Některé aplikace potřebují ukládat získaná data ve větším množství na trvalé úložiště, například do textového souboru. Pro tento účel je možné využít knihovnu pro práci s SD (Secure Digital) paměťovými kartami. Tato knihovna podporuje práci se základními souborovými systémy FAT16 a FAT32. Omezení se týká délky názvu souboru na osm znaků a tři znaky přípony, přičemž je podporováno procházení obsahu paměťové karty pomocí zadávání celých cest k souborům. Dostupné jsou základní operace pro práci se soubory, jedná se především o vytváření a mazání souborů, dále zápis do souboru a čtení ze souboru.

Mikrokontroléry Atmel AVR [4] poskytují možnost využití synchronního komunikačního protokolu SPI. Tento komunikační protokol umožňuje plně duplexní komunikaci mezi dvěma zařízeními, přičemž na SPI sběrnici může být v jeden okamžik připojeno i více zařízení.



Obrázek 2.3: Příklad zapojení více zařízení na SPI sběrnici [19]

Poskytnutá knihovna pro SPI komunikaci umožňuje rozšířené nastavování parametrů tohoto typu komunikace. Jde například o pořadí zasílaných bitů a rychlost komunikace. Dále lze uvést možnost konfigurace polarit hodin, výstupních bitů a vstupních bitů. Při vlastní komunikaci již samotná knihovna a SPI obvod mikrokontroléru zajišťuje správné nastavování vstupních a výstupních signálů v závislosti na zadané konfiguraci.

Knihovna Stepper je určena pro ovládání krokových motorů. Knihovna podporuje unipolární i bipolární krokové motory s možností připojení pomocí dvou nebo čtyř pinů. Každému objektu motoru lze určit rychlost otáčení v počtu otáček za sekundu. Následně lze připojené krokové motory řídit pomocí zasílání zpráv objektům serv s počtem kroků, které má motor provést.

Pro řízení řádkových textových displejů, které jsou založeny na technologii tekutých krystalů je určena knihovna LiquidCrystal. Knihovna umožňuje provádět s připojeným displejem běžné základní operace, například ovládání kurzoru displeje, zápis dat na displej a posun textu na displeji.

Ovládání grafických displejů je možné pomocí knihovny TFT LCD. Tato knihovna vyžaduje pro svoji funkčnost knihovnu SPI, jelikož k připojení modulu s displejem je použito rozhraní SPI. Pro ovládání lze použít hardwarové SPI nebo libovolné dostupné piny. Knihovna poskytuje možnost pro vykreslování obrázků, které mohou být uloženy na paměťové kartě. Dále knihovna poskytuje metody objektu pro výpis textu na připojený displej. Podporována je i možnost nastavování barev z modelu RGB pro vykreslování grafická primitiva, uvést lze obdélníky, kruhy, body a úsečky.

Zařízení platformy Arduino jsou schopná provádět sériovou komunikaci na pinech 0 a 1. Tyto piny jsou zároveň používány sběrnici USB pro komunikaci s počítačem a vlastní komunikace je podporována přímo hardwarem zařízení.

Pro sériovou komunikaci na jiných digitálních pinech je možné využít knihovny SoftwareSerial. Tato knihovna provádí asynchronní sériovou komunikaci softwarově, bez podpory hardwaru a umožňuje tak mít otevřených více nezávislých komunikačních kanálů. Pro komunikaci jsou zapotřebí dva digitální piny pro odesílání a příjem dat a vhodně konfigurovaná komunikační rychlost mezi zařízeními. Tato knihovna neumožňuje podrobnější transparentní nastavení komunikace mezi zařízeními, lze pouze zvolit rychlost komunikace a případně nastavit inverzní interpretaci přijatých bitů. Případné nastavení paritních bitů a stop bitů není umožněno.

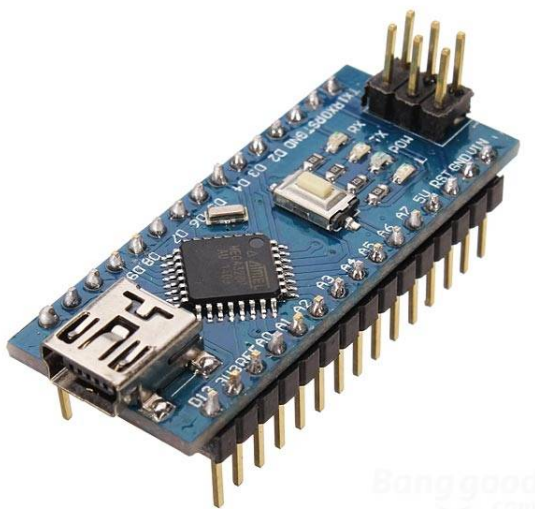
Pro potřeby řízení serv je k dispozici knihovna Servo. Tato knihovna umožňuje vytvářet objekty pro serva a pomocí zasílání zpráv těmto objektům serva řídit. Samotné řízení serv je založené na technice PWM. Vnitřní implementace této knihovny používá hardwarových čítačů/časovačů a jejich kanálů na čipu mikrokontroléru pro řízení střídavého řídicího pulsu pro servo.

Jednotlivá serva je nutné nejprve softwarově přiřadit k používaným pinům. Následně lze používat metody objektů pro ovládání těchto serv. Objektům, které reprezentují serva lze zasílat zprávy buď s hodnotou požadovaného úhlu natočení serva, a to v rozmezí 0 až 180 stupňů, nebo zprávy s šířkou pulsu v mikrosekundách. Přičemž druhý způsob je doporučený pro nestandardní serva, které nepoužívají běžnou šířku pulsu v intervalu 1000 až 2000 mikrosekund.

## 2.3 Arduino Nano

Pro projekt byla zvolena verze Arduino Nano a to především kvůli nízké ceně, malé velikosti vývojové desky, vybavenosti převodníkovým čipem USB na RS-232 pro komunikaci s hostitelským počítačem a dostatečnému počtu vstupně výstupních pinů. Vývojová deska obsahuje mikrokontrolér Atmel ATmega328 a sadu podpůrných obvodů. Programování se provádí přes mini-B USB konektor, v případě potřeby obejít zavaděč je možné mikrokontrolér programovat pomocí

technologie ISP. Deska poskytuje 8 vstupně výstupních analogových pinů a 14 vstupně výstupních digitálních pinů, z tohoto počtu 6 pinů podporuje PWM s rozlišením 8 bitů. Všechny 14 digitálních pinů umožňuje použít interní pull-up rezistor. Piny 2 a 3 na desce je možné konfigurovat pro použití externího přerušování. Všechny dostupné analogové piny poskytují 10 bitové rozlišení analogově digitálního převodníku, přičemž jako referenční napětí je možné použít vnitřní operační napětí 5V nebo napětí připojené na pin AREF vývojové desky. Reset pin umožňuje použití externího hardwarového resetování mikrokontroléru. Některé piny mohou být konfigurovány pro odlišné účely různým způsobem a mohou pracovat v analogovém nebo digitálním režimu.



Obrázek 2.4 Deska Arduino Nano [2]

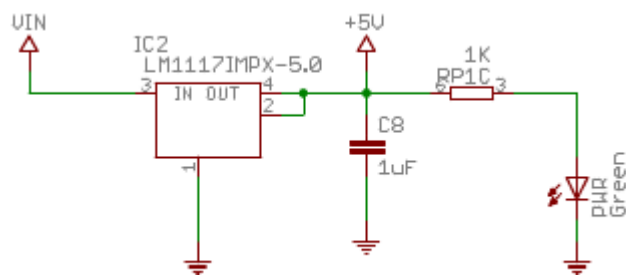
### 2.3.1 Atmel ATmega328

Atmel ATmega328 [4] je 8 bitový mikrokontrolér firmy Atmel, který je založený na instrukční sadě RISC. Typická pracovní frekvence tohoto mikrokontroléru je 16MHz, přičemž maximální frekvencí je 20MHz. Tento mikrokontrolér poskytuje 32 kilobajtů flash paměti pro program, 2 kilobajty operační paměti typu SRAM a 1 kilobajt paměti EEPROM. Operační napětí mikrokontroléru je v rozsahu 1,8V až 5,5V. Spotřeba tohoto mikrokontroléru se pohybuje v jednotkách miliampér. V závislosti na použití lze využít interní nebo externí oscilátor pro generování vnitřního hodinového signálu.

Mikrokontrolér obsahuje hardwarový blok pro SPI komunikaci. Technologie ISP je tímto mikrokontrolérem podporována. Většina instrukcí je vykonána v jednom taktu hodinového signálu mikrokontroléru. Výjimkou jsou například instrukce násobení, celočíselná násobička, která je integrovaná na čipu mikrokontroléru vyžaduje dva hodinové cykly pro operaci násobení. Dále jsou na čipu mikrokontroléru integrovány dva 8 bitové čítače/časovače a jeden 16 bitový čítač/časovač.

## 2.3.2 Napájení

Celá vývojová deska, včetně mikrokontroléru, pracuje s napětím 5V, pouze pro potřeby některých externích periférií je možné využít napětí 3,3V na pinu 3V3. Napájet vývojovou desku lze ze sběrnice USB při připojení vývojové desky k počítači. Další možností je využít integrovaný napěťový regulátor na desce, který přizpůsobí vstupní napětí na 5V pro napájení desky, a tímto způsobem zařízení napájet neregulovaným externím zdrojem s napětím v rozsahu 6V až 20V. Případně lze zařízení napájet regulovaným externím zdrojem o napětí 5V. Spotřeba vývojové desky s mikrokontrolérem a bez uvažování externích periférií je 19mA. Maximální vstupní nebo výstupní proud pro jeden pin desky je 40mA.



Obrázek 2.5: Zapojení napájení Arduino Nano [2]

## 2.3.3 Napěťový regulátor AMS1117

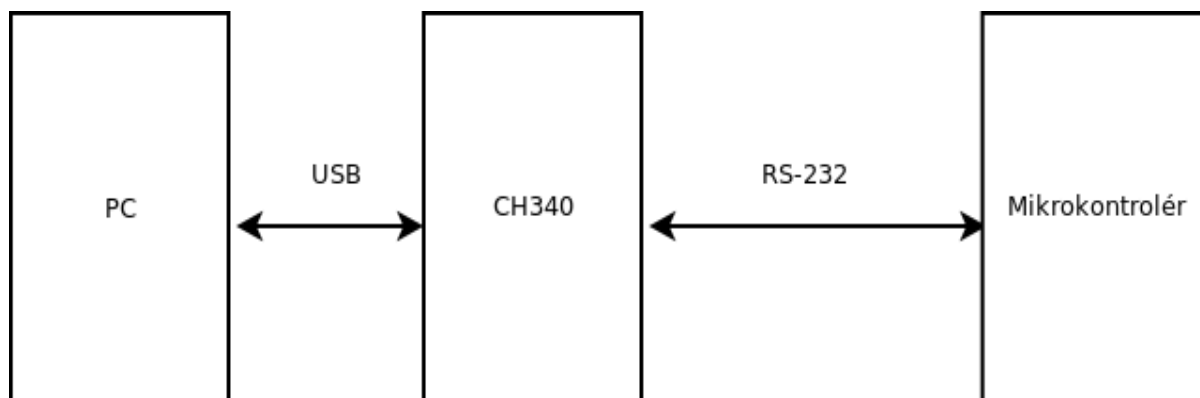
Na desce plošného spoje Arduino Nano je integrován lineární napěťový regulátor AMS1117 [13] vyráběný firmou Advanced Monolithic Systems. Tento napěťový regulátor se vyrábí v několika provedeních, uvést lze pouzdra typu SIOC, TO a SOT. V případě platformy Arduino je použita verze pouzdra SOT v konfiguraci se třemi piny pro vstupní napětí, výstupní napětí a společné zemnění.

Výrobce udává, že napěťový regulátor umožňuje šest úrovní výstupního napětí v rozsahu od 1,5V do 5,0V. Ztráta napětí na regulátoru se pohybuje na úrovni 1V, přičemž za typickou hodnotu se považuje 1,1V a za maximální 1,3V. Je tedy nutné, aby se vstupní a výstupní napětí lišilo alespoň o 1V. Avšak při nižším zatížení jsou ztráty také nižší. Udávané maximální vstupní napětí je 15V. Na výstupu regulátoru je možné odebírat maximální proud o hodnotě 1A, přičemž výrobce udává, že špičkovým proudem je 1,5A. Delší využívání vyššího proudu, než je hodnota 1A může způsobit vyšší zahřívání regulátor a případně může být regulátor poškozen. S tímto souvisí operační teplota, která se pohybuje v rozmezí -40°C až +125°C.

### 2.3.4 Převodník CH340

Pro převod mezi rozhraním USB a sériovou linkou standardu RS-232 je na plošném spoji Arduino Nano integrován převodník CH340G [12], který vyrábí čínská firma Nanjing Qin Heng Electronics. Tento převodník umožňuje připojení zařízení pracujících s paralelním nebo sériovým rozhraním, případně zařízení, které pracují se standardem IrDA. Převodník je uzavřen v pouzdře typu SSOP-20.

Převodník emuluje na sběrnici USB sériové rozhraní v plně duplexním režimu. Rychlost komunikace se může pohybovat od 50 baudů za sekundu do 2 megabaudů za sekundu. Sériovou linku standardu RS-232 je možné připojit v základní konfiguraci tří vodičů, společné zemnění, vysílací a přijímací vodič, nebo v rozšířené konfiguraci se zapojenými řídicími vodiči. Požadované napájecí napětí převodníku je 3,3V nebo 5V, přičemž nesmí přesáhnout 6,5V, kvůli poškození převodníku. Operační teplota převodníku je  $-40^{\circ}\text{C}$  až  $+85^{\circ}\text{C}$ . Spotřeba tohoto převodníku může být až 30mA, jako typickou spotřebu ale výrobce udává 12mA. Pro svoji funkci vyžaduje převodník vstupní hodinový signál 12MHz.

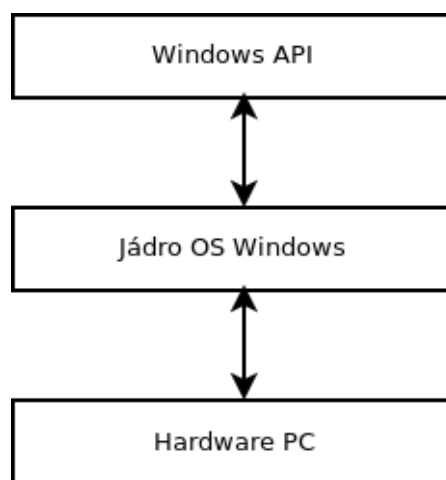


Obrázek 2.6: Blokové schéma zapojení CH340



## 2.4 Windows API

Windows API [8] je rozhraní služeb jádra systému Microsoft Windows. Toto aplikační programové rozhraní nabízí možnost používat prostředky operačního systému na nízké úrovni programového kódu. Rozhraní je vytvořeno v jazyce C a poskytuje rozsáhlou sadu maker, struktur, výčtů a funkcí. Knihovna Windows API navíc definuje vlastní datové typy ke standardním datovým typům jazyka C. Mezi tyto typy patří například datový typ znaku, který umožňuje změnu velikosti v závislosti na cílovém prostředí pro aplikaci. Windows API obecně poskytuje funkcionalitu pro práci se systémovými službami jádra, grafickým uživatelským rozhraním, síťovými službami a dalšími.



Obrázek 2.7: Umístění Windows

*API v hierarchii OS*

### 2.4.1 System Services

Systémové služby Windows API nabízí přístup ke zdrojům počítače a operačního systému. Mezi tyto zdroje se řadí správa paměti, přístup k souborovému systému, vytváření a řízení aplikačních vláken a procesů, přístup k zařízením, která jsou připojena k počítači a jiné.

Přístup k souborovému systému umožňuje správu zařízení pro ukládání dat se souborovým systémem typu FAT nebo NTFS. K tomuto se váže možnost správy logických svazků na médiích, mezi které se řadí pevné disky, optické disky a další úložná média. Aplikační rozhraní umožňuje tvorbu a správu souborů a složek v dostupných souborových systémech.

Windows API podporuje práci s odlišnými časovými formáty a konverze mezi nimi. Systémový a lokální čas je založený na práci s koordinovaným světovým časem, přičemž lokální čas pracuje s časovou zónou daného počítače. Při práci se souborovým systémem se využívají prostředky pro souborový čas, tento čas je reprezentován 64 bitovou hodnotou s přesností na 100 nanosekund. Systém Windows využívá tento formát při zápisu časových informací do souborového systému. Při využití souborového systému NTFS je zapisována hodnota koordinovaného světového času

a to bez závislosti na časové zóně prostředí operačního systému. Při zápisu časového razítka do souborového systému typu FAT je zapisován lokální systémový čas.

Pomocí Windows API lze dále spravovat služby a ovladače operačního systému. Funkce pro správu těchto komponent umožňují instalování, správu, vzdálené volání, ovládání a odstraňování služeb v systému.

Prostřednictvím funkcí Windows API je možné spravovat registry systému Windows. Funkce rozhraní dávají možnost zcela kontrolovat vytváření, správu a využití systémových registrů programátorem.

Využití části aplikačního rozhraní Compression API umožňuje provádět bezztrátovou kompresi a dekompresi dat s vlastní volbou kompresního algoritmu. Dostupných je několik kompresních algoritmů, uvést lze například algoritmus MSZIP nebo LZMS.

Správa paměti pod Windows API sestává z mnoha možností. Mimo funkcí jazyka C, které jsou definovány ve standardu tohoto jazyka a lze je bezpečně použít, je možné využívat specializované funkce z aplikačního rozhraní systému Windows. Funkce pro mapování souborů do paměti umožňují namapování souboru z disku počítače do fyzické paměti počítače a následně sdílet namapovaný soubor mezi více procesy s odlišnou virtuální pamětí. Případně lze tyto funkce využít pro vytvoření sdílené oblasti paměti bez mapování souboru do této paměti. Dynamická paměť na haldě může být alokována pomocí funkcí typu Heap, které umožňují detailní správu alokované paměti a to včetně vytváření a správy soukromých hald procesu, které bude proces spravovat ve vlastní režii.

Část Device Management Windows API umožňuje práci s ovladači zařízení a se zprávami a upozorněními o hardwarových komponentách, které jsou připojeny k počítači. Tyto zprávy například upozorňují na události jako je připojení nového zařízení, změna napájení a podobné. Uživatelské aplikace se mohou registrovat k příjmu zpráv o událostech v systému nebo mohou požádat o přímou komunikaci s ovladačem připojeného zařízení.

Knihovna obsahuje podporu pro správu programových procesů a vláken uživatelem. Procesům a vláknům lze, mimo jiné, přiřazovat priority a rozsah dostupných prostředků.

## 2.4.2 Communication Resources

Pro ovládání prostředků pro komunikaci se zařízeními, například přes sériové a paralelní porty, je možné použít rozhraní Communication Resources. Tato část rozhraní poskytuje funkce pro čtení a zápis dat do zařízení, která jsou připojena přes uvedené porty. Při konfiguraci připojení lze definovat omezení pro aplikaci na čtení nebo zápis do zdroje. Mimo jiné lze také definovat sdílený přístup ke zdroji, kdy jeden proces může ze zdroje číst a zapisovat do něj a další proces může například ze zdroje pouze číst. V případě komunikace přes sériový port lze nastavit rychlost přenosu v baudech za sekundu, počet bitů v přenášeném bajtu, počte stop bitů nebo typ kontroly parity.

## 2.4.3 Windows Sockets 2

Knihovna Windows Sockets 2 umožňuje pracovat se sítovou komunikací podobným způsobem, jako v případě knihovny BSD Sockets. Knihovna je zaměřena především na programování sítových služeb, které jsou založeny na modelu TCP/IP a podporuje implementaci obou protokolů IPv4 a IPv6. Na transportní vrstvě jsou podporovány oba protokoly TCP a UDP. Funkce, které jsou společné pro Windows Sockets 2 a BSD Sockets mají totožné prototypy, kvůli vzájemné kompatibilitě. V případě rozšiřujících funkcí je použit systém prototypů Windows API.

Knihovna umožňuje vytvářet schránky na základě typu síťového protokolu, například IPv4 a typu transportního protokolu. Přijmout data ze schránky nebo odeslat dat vzdálenému zařízení lze poté například pomocí funkcí *recv*, respektive *send*.

## 2.4.4 Windows User Interface

Windows User Interface je část Windows API, která slouží pro tvorbu a ovládání grafického uživatelského rozhraní. Programátor definuje vzhled aplikace a funkce, které slouží pro definování chování aplikace. Uživatelská aplikace nezískává přímý přístup k celé obrazovce, místo toho získává prostor pro vykreslování rozhraní ve svém okně. Při změně rozhraní aplikace, například zmenšení okna aplikace, je operačním systémem zaslána zpráva uživatelské aplikaci o dané události a aplikace má příslušným způsobem na zprávu reagovat.

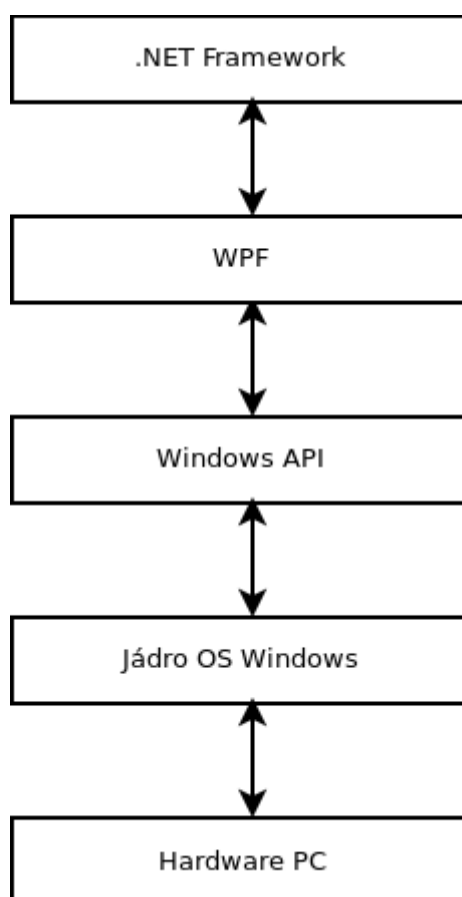
Operační systém udržuje pro každou aplikaci fronty zpráv o událostech. Zprávami informuje operační systém aplikaci o nastalých událostech. Mezi tyto události se řadí například stisk tlačítka myši, pohyb myši nebo stisk klávesy na klávesnici. Uživatelská aplikace může následně vybírat přijaté zprávy z fronty a provádět odpovídající akce. V případě stisku kombinace kláves je možné získávat od operačního systému zprávy o výsledném příkazu nebo definovat vlastní kombinace kláves.

Uživatelská aplikace má možnost vytvářet dialogová okna pro informování nebo dotazování uživatele. Do výsledných aplikací je možné vkládat potřebné zdroje, jako jsou například obrázky, ikony a další. Při vlastním běhu má poté aplikace možnost tato data získávat a používat.

Mezi prvky, které je možné v uživatelském rozhraní používat se řadí například tlačítka, seznamy a rámečky pro vkládání textu. Při použití textového kurzoru je možné vytvořit pouze jeden kurzor pro okno, přičemž by měl být kurzor po ztrátě zaměření okna odstraněn.

## 2.5 Windows Presentation Foundation

Windows Presentation Foundation (WPF) [10] je podsystém pro vytváření aplikací s grafickým uživatelským rozhraní pro operační systém Microsoft Windows. WPF je součástí platformy Microsoft .NET Framework. Technologie WPF odděluje tvorbu popisu vzhledu aplikace a programový kód. Aplikace WPF jsou vytvářeny v jazyce Visual Basic, C/C++ nebo C#, přičemž popis grafického uživatelského rozhraní může být zapsán ve značkovacím jazyce XAML. WPF aplikace umožňují použít přístup Model-View-Controller pro práci s daty v aplikaci. Pomocí vázání dat lze například navázat data aplikace k XAML prvků grafického rozhraní. Do vytvořených aplikací lze zahrnovat zdroje, což jsou řetězce, obrázky a další objekty, které budou distribuovány s aplikací, knihovny a lze je vyžít v programovém kódu. Technologie je zaměřena na snadno přenositelnost aplikací a možnost vytváření kvalitního grafického rozhraní aplikace.



Obrázek 2.8: Technologie WPF v hierarchii OS Windows

## 2.5.1 Grafické uživatelské rozhraní

Grafické uživatelské rozhraní je v aplikacích typu WPF vykreslováno pomocí technologie Direct3D a podporuje 2D i 3D grafiku. Tento přístup dovoluje využití grafického akcelerátoru v zařízení a snížit zatížení procesoru. Technologie dovoluje použití jednotek Device-independent pixels (DIPs), které definují obecné rozměry prvků aplikace bez závislosti na konkrétním zobrazovacím zařízení. Grafické rozhraní je tak možné tvořit pomocí zadání absolutních rozměrů, DIPs nebo pomocí zadání poměru rozměrů jednotlivých prvků a nechat přepočítání výsledných rozměrů prvků na grafickém podsystému.

## 2.5.2 Jazyk XAML

Deklarativní značkovací jazyk XAML [14] je založený na značkovacím jazyce XML. Jazyk XAML podporuje použití kódování UTF-8 a UTF-16, přičemž rozlišuje velikost znaků. Popis grafického rozhraní aplikace je hierarchický. Prvky jazyka XAML mohou obsahovat vlastnosti, přiřazené události a další vnořené prvky. Vlastnosti objektu mohou být plně definovány v jazyce XAML, přiřazené události musí být zpracované v programovém kódu aplikace. Párové značky nejsou vyžadovány, pokud prvek neobsahuje další prvky. Jazyk XAML podporuje jmenné prostory, vázání dat a připojování zdrojů aplikace.

Vytvořením nových tříd a jejich popisem v jazyce XAML lze vytváření nové prvky grafického rozhraní. Takové prvky mohou dědit od existujících prvků, být zcela nové nebo se skládat z několika existujících prvků. Takto vytvořené prvky lze následně používat v aplikacích stejným způsobem, jako prvky z knihoven .NET Framework.

Pro existující prvky XAML lze vytvářet šablony, které definují styly, tedy vzhled prvku a v něm zanořených prvků. Tyto šablony mohou být použité pro více prvků.

## 2.5.3 Prvky grafického rozhraní

Prvky grafického rozhraní jsou objekty, které je možné vytvářet dynamicky pomocí programového kódu nebo popisem v jazyce XAML. Základní prvky dostupné v knihovnách .NET Framework jsou vektorové a umožňují tak zachování kvality při změnách rozměrů grafického rozhraní. Každý prvek obsahuje vlastnosti a události. Mezi vlastnosti se řadí například rozměry objektu.

Události objektu obsahují přiřazené obslužné metody pro dané události, například kliknutí na objekt, změna rozměru objektu. Události musí být obslouženy programovým kódem, není možné definovat obsluhu v jazyce XAML.

## 2.5.4 Programový kód

Programový kód aplikace může být zapsán v libovolném jazyce, který podporuje Common Language Runtime (CLR), jedná se například o jazyky C# a Visual Basic. WPF aplikace používají jako výchozí přístup takzvaný „managed code“ [11], ale programátor může využít i takzvaný „unmanaged code“.

„Managed code“ je programový kód, který je spravován virtuálním strojem CLR. Kompilací kódu, který podporuje CLR vzniká Intermediate Language (IL). Jde o přenositelný binární kód pro virtuální stroj CLR, tento kód bude zpracováván při běhu aplikace tak, že bude kompilován Just-in-Time (JIT) technikou do nativního kódu cílového stroje. CLR má za cíl zvýšení efektivity programu a bude podle informací získaných za běhu aplikace kód spravovat.

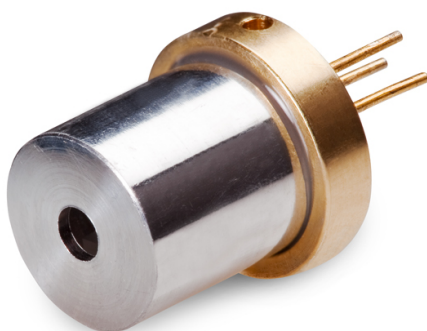
„Unmanaged code“ je programový kód, který není spravován virtuálním strojem CLR. Tento druh programového kódu zároveň umožňuje programátorovi použít knihovny, které nejsou vytvořené pro CLR. Umožňuje například použití adresových ukazatelů v jazyce C#. Taková část kódu se označuje jako „unsafe context“. Není zaručeno bezpečné vykonání kódu. Kód v jazyce C/C++ může být vytvořen pro CLR jako IL nebo jako nativní „unmanaged code“.

## 2.6 Laser

Laser je optickým zdrojem monochromatického světla, které je z tohoto zdroje vyzařováno v úzkém svazku fotonů s malou rozbíhavostí. Vyzařované světlo nemusí být ve viditelném spektru, může jít i o rentgenové záření nebo ultrafialové světlo. Laser se dnes využívá v mnoha oborech a zařízeních. Jmenovat lze například laserové tiskárny, optické mechaniky, síťové optické spoje, obor bioinženýrství a další. Existuje mnoho typů laserů, dnes se používají velmi často lasery polovodičové a plynné. Dále budeme uvažovat lasery polovodičové, jelikož je tento typ laseru použit v tomto projektu.

### 2.6.1 Laserová dioda

Laserová dioda [5] je polovodičový optoelektronický prvek, který je schopen při průchodu proudu PN přechodem vyzařovat světlo v úzkém pásmu vlnových délek, jedná se tedy o monochromatické světlo. Laserová dioda je zdrojem koherentního záření, pokud je provozována na prahovém a vyšším proudu. Při nižších proudech se stává zdrojem nekoherentního záření. Na rozdíl od LED je laserová dioda zdrojem záření, které má menší rozbíhavost výstupního svazku, i přesto je ale třeba použít čočky pro odstranění rozbíhavosti a usměrnění výstupního záření.



Obrázek 2.9: Laserová dioda [17]

Dalším problémem je chlazení, jelikož při delším kontinuálním vyzařování se laserová dioda značně zahřívá, je třeba ji dostatečně chladit, například kovovým chladičem, jinak hrozí její zničení. V případě diod s nízkým výkonem postačí jako chladič pouzdro diody, které bude odvádět vytvářené teplo. Laserové diody jsou schopné dosahovat vysokých frekvencí spínání v řádu GHz, toto se uplatňuje například v optických spojích. Sítě, které jsou postavené na optických prvcích mohou dnes dosahovat rychlostí přenosu dat v řádech terabitů za sekundu.

## 2.6.2 Pouzdro

Laserová dioda bývá zapouzdřena v kovovém obalu společně s diodou pro monitorování výkonu. Toto je dáno tím, že laserová dioda vyzařuje světlo oběma směry a asi 10% světla dioda vyzařuje dozadu. Toto světlo dopadá na monitorovací diodu, což bývá fotodioda, a je tak možné monitorovat průběh výkonu laserové diody. Při zapouzdření se vyvádí kontakty obou diod odděleně a není tak třeba využívat možnosti monitorování výkonu diody.

V pouzdře s laserovou diodou bývá také zabudována čočka a odrazové zrcátko pro směrování vyzařovaného světla, není to ale podmínkou. Podmínkou je to především v optických spojích. V těchto případech bývají diody zapouzdřené ještě v dalším, větším pouzdře, většinou již plastovém, a připravené k provozu. Pouzdro diody je uzavřené hermeticky, především kvůli nečistotám.

## 2.6.3 Napájení

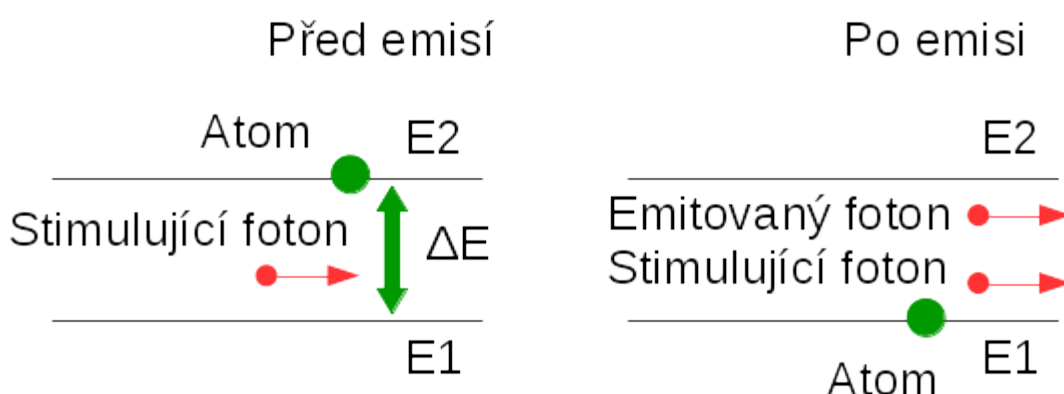
K napájení laserových diod je nutné použít proudový zdroj nebo napěťový zdroj s ochranným rezistorem. Zdroj by měl mít dostatečně velkou výstupní impedanci a měl by poskytovat stabilní proud. Výše zmíněné chlazení laserové diody je nutné, protože při vysokých teplotách dioda mění svůj odpor, důsledkem čehož jí prochází vyšší proud a tím může dojít ke zničení diody. Pro napájení laserové diody je nutné zvolit správný pracovní bod podle katalogového listu. Přičemž je nutné vzít v úvahu, že při zahřívání, i když bude dioda chlazená, se zvyšuje potřebný prahový proud. Před napětím v závěrném směru je vhodné laserovou diodu chránit směrovou diodou antiparalelně, toto napětí se pohybuje okolo hodnoty 1V.

## 2.6.4 Princip funkce

V principech funkce laserových diod se uplatňuje kvantová mechanika a principy polovodičových prvků. Funkce laserové diody je založena na stimulované emisi fotonů. Aktivní prostředí je v oblasti PN přechodu, při průchodu proudu přechodem se využívá injekce elektronů a děr. Doba jejich života se pohybuje v nanosekundách, následně rekombinují, přičemž je vyzářen foton, který má náhodnou fázi, směr i polarizaci. V některých okamžicích je energie z rekombinace pohlcena krystalovou mřížkou polovodiče a způsobuje zahřívání diody. V případě nízkého napětí nebo proudu, který prochází diodou, nedochází ke stimulované emisi záření, ale pouze ke spontánní emisi nekoherentního záření.

## 2.6.5 Stimulovaná emise

Pokud do prostředí PN přechodu těsně před rekombinací a spontánní emisí vstoupí foton, nazýváme jej stimulujícím fotonem, s energií rovnou energii, která je potřebná pro přechod mezi energetickými hladinami přechodu, tak je nový foton emitován ještě před samotnou spontánní emisí. Tento nový foton bude mít energii rovnou energii stimulujícího fotonu, stejný směr šíření a stejnou polarizaci a fázi. Při uvažování kvantové mechaniky není stimulující foton pohlcen, ale projde soustavou. Ve výsledku jsou vyzářeny dva fotony. Vzniká koherentní záření, které je z části přiváděné na vstup a stimuluje další emisi fotonů. Zpětná vazba je tvořena optickým rezonátorem, kterým může být difrakční mřížka polovodiče. V případě laserových diod je převládajícím typem emise stimulovaná emise.



Obrázek 2.10: Emise fotonu stimulací PN přechodu

Na obrázku výše je ilustrován princip stimulace PN přechodu fotonem. V tomto případě uvažujeme pouze dvě energetické hladiny E2 a E1. Soustava přejde po vyzáření nového fotonu do stavu s nižší energií E1.

## 2.6.6 Spontánní emise

Ke spontánnímu záření dochází v laserové diodě v případě nízkého vstupního proudu, napětí nebo pokud není soustava stimulována jinými fotony. V takovém případě elektrony přecházejí do vyšší energetické hladiny a buď vyzáří foton nebo je energie po rekombinaci pohlcena diodou a způsobuje zahřívání. V případě vyzáření fotonu se tento nešíří směrově, ale náhodným směrem, s náhodnou fází a polarizací. Světelné záření je nekoherentní. Při dostatečném napětí a proudu soustava přechází do stimulovaného záření, které je popsáno výše.

## 2.6.7 Záření

Koherentní záření je jedním ze způsobů vyzařování světla. Takové záření má stejnou fázi, polarizaci a směr. Vlnová délka tohoto typu záření je v úzkém intervalu hodnot. V případě koherence



časové jde o uvedené a stejný časový rozestup vyzařování. V laserech vzniká koherentní záření díky stimulované emisi a monochromatickosti.

Nekoherentní záření vzniká v laserech při spontánní emisi bez monochromatického záření. Fotony tohoto záření nemají stejný směr, fázi nebo polarizaci. Vlnová délka tohoto záření se tedy nepohybuje v úzkých intervalech, ale může mít široký rozsah.

Podle způsobu vyzařování z PN přechodu se laserové diody dělí na dva druhy. Hranově vyzařovací laserová dioda je dnes více používaný typ laserové diody, jelikož technologie je dostupná a vyzkoušená. Vyzařování probíhá z hrany PN přechodu polovodiče. Vyzařovaný paprsek má větší rozptyl a dioda má nižší účinnost, než plošně vyzařovací dioda.

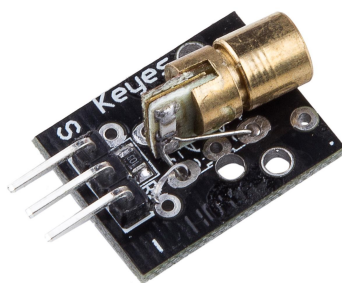
U plošně vyzařovací laserové diody probíhá vyzařování z plochy PN přechodu. Má vyšší účinnost než hranově vyzařovací dioda, ale vyžaduje složitější technologie pro výrobu. PN přechod je možné uložit mezi odrazivé plochy, pomocí nichž je dosaženo vyšší efektivity. Pro efektivitu je tato technologie vhodnější pro optické spoje.

## 2.6.8 Použití laserové diody

Pomocí současných technologií lze vyrobit laserové diody s životností v řádech stovek tisíc hodin. Široké využití mají laserové diody v oboru elektrotechniky a informačních technologií. Používají se například v síťových optických spojích, laserových tiskárnách a pro záznam a čtení optických disků. Laserové ukazovátka je příkladem využití laserové diody s nízkým výkonem, typicky do 5mW. Dále se laserové diody využívají pro měření vzdálenosti, ve čtečkách čárových kódů, v průmyslu pro řezání materiálů, ve zdravotnictví a ve vědeckých oborech.

## 2.6.9 Laserový modul KY-008

Pro projekt byl zvolen laserový modul KY-008. Tento modul sestává z laserové diody a čočky pro usměrnění vyzařovaného paprsku. Vyzařovaný laserový paprsek je červený o vlnové délce 650nm. Modul navíc obsahuje základní elektronický obvod, včetně ochranného rezistoru, pro snadné zapojení do obvodu. Napájecí napětí je 5V.



Obrázek 2.11 Laserový  
modul KY-008

## 2.7 Modelářské servo

Modelářské servo [7] a [15] je specifický druh elektrického motoru, který, právě na rozdíl od klasických elektrických servomotorů, umožňuje přesně nastavovat pozici natočení rotoru. Zde je právě rozdíl oproti klasickým servomotorům, které po přivedení napájení otáčejí rotorem a jejich řízení závisí pouze na pulsně šířkové modulaci. Modelářské servo obsahuje také servomotor, ale ten je řízen složitější elektronikou, která na základě vstupního signálu a aktuální pozice rotoru provádí požadované natočení rotoru a na této pozici rotor drží. Velkého uplatnění našla serva v robotice, automatizaci, modelářství a v počítačových pevných discích. Serva typicky pracují v nízkých otáčkách, na rozdíl od běžných stejnosměrných servomotorů a zapouzdří se do obalu společně s řídicí elektronikou a převodovými ozubenými koly. Obal může být plastový nebo kovový. Běžné modelářské servo má vyvedené tři vodiče pro napájení, zemnění a řízení, zde je opět rozdíl oproti servomotorům, které používají pouze dva napájecí vodiče a žádné dedikované řízení. Převodový systém ozubených kol může být celý plastový, jako u nejlevnějších serv, nebo mohou být použita kovová ozubená kola, v případě dražších modelů.

### 2.7.1 Stejnosměrný motor

Stejnosměrný motor [7] je základním druhem elektrického motoru. Motory tohoto druhu lze řídit změnou napájecího napětí. Těmito změnami dochází k lineární regulaci otáček motoru. Nejčastěji se používají stejnosměrné motory s buzením permanentními magnety ve statoru. V případě stejnosměrných motorů s vysokým výkonem se může používat ventilace k chlazení motoru. V ostatních případech se používá chlazení pouze kostrou motoru. Pro výrobu magnetů statoru se používá magneticky tvrdých materiálů, například sloučeniny neodymu, bóru a železa. Rotor obsahuje vinutí s vyvedením na komutátor, na který doléhají uhlíkové kartáče.

### 2.7.2 Řízení serv

Serva lze řídit [6] a [7] analogově nebo digitálně. V případě modelářských serv se používají oba přístupy pro řízení. Digitální serva obsahují přídavnou elektroniku pro práci s přijatým digitálním řídicím signálem, kvůli tomu je obvykle i vyšší jejich cena. Naproti tomu analogově řízená serva používají jednodušší vnitřní elektroniku, jejich cena je pak nižší. Řízení serv je například možné pomocí předem naprogramovaného mikrokontroléru, lze snadno a dynamicky měnit programově a provádí se pomocí pulsně šířkové modulace, která je především známá pod anglickou zkratkou PWM (Pulse Width Modulation).

### 2.7.3 Pulsně šířková modulace

Tato technika se používá i mimo oblast řízení servomotorů a modelářských serv [6], například pro nastavování jasu LED. V každém případě se řídicí elektronice zařízení zasílají pulsy proměnné délky. Délka tohoto pulsu udává v případě řízení serv řídicí elektronice serva úhel natočení rotoru serva. Každé servo má různě definované délky pulsů, obvykle jde o délky v řádech stovek až tisíc mikrosekund. V případě řízení LED jde o rychlé vypínání a zapínání, které se při vhodné frekvenci jeví lidskému oku jako nižší jas LED. Poměr doby aktivity pulsu k době celého cyklu se označuje jako střída nebo také anglicky jako duty cycle.

### 2.7.4 Modelářské servo Tower Pro SG90

Jedná se o jedno z nejvíce rozšířených a cenově dostupných serv pro modelářství. T-Pro SG90 [16] váží 9 gramů a jedná se o mikro servo. Toto servo umožňuje natočení rotoru v rozsahu 0 až 180 stupňů. Požadované napájecí napětí je v rozsahu 4,8V až 7,2V. Převody jsou plastové a točivý moment může dosáhnout na 1,8kg na délce páky 1cm. Toto servo se vyrábí v digitálně i analogově řízené verzi. Používaná šířka pulsu řízení je 20ms s délkou střídý v rozsahu 1ms až 2ms.



Obrázek 2.12 Modelářské servo Tower Pro SG90

### 3 Současný stav a plán práce

Cílem práce je vytvořit zařízení a softwarovou sadu pro práci s tímto zařízením. Účelem zařízení bude prosvícení vzdálené plochy podle uživatelem zadaných mezí, zařízení například projde laserem po řádcích vzdálenou plochu. Takové zařízení bude sestávat z pohyblivé základny, v níž budou umístěna dvě serva pro ovládání v horizontálním a vertikálním směru. Na základně bude umístěn modul s laserovou diodou, která bude použita pro prosvětlování plochy. Zařízení bude řízeno pomocí platformy Arduino a zároveň bude možné přes tuto platformu zařízení ovládat z počítače.

Softwarová část práce bude sestávat z programu pro řízení zařízení. Tento program bude umožňovat vzdálené nastavení zařízení a jeho ovládání. Ovládání zařízení z počítače bude zajišťovat aplikace s grafickým uživatelským rozhraním, pomocí které bude uživatelům umožněno vytvářet nastavení, zasílat nastavení do zařízení a ukládat nastavení do souborů v počítači pro pozdější použití. Pro automatizaci použití zařízení bude vytvořena softwarová knihovna, která umožní aplikacím třetích stran volat obsluhu pro zaslání nastavení do zařízení.

Komunikace cílového zařízení s platformou PC bude realizována pomocí emulace sériové linky přes sběrnici USB. Napájení celého zařízení bude taktéž zajištěno z hostitelského počítače přes sběrnici USB.

V práci využiji znalosti získané studiem technologií, které jsou uvedené v druhé kapitole. Zcela jistě uplatním znalosti ze studia platformy Arduino a softwaru k této platformě. V tomto ohledu bude důležitá oblast řízení serv a vzdálená komunikace s počítačem. V případě softwarové knihovny využiji získané znalosti o Windows API se zaměřením na oblast Communication Resources. Pro tvorbu aplikace s grafickým uživatelským rozhraním využiji Microsoft .NET Framework a technologii WPF s jazyky C# a XAML. V softwaru pro řízení mikropočítače by jsem mohl využít paměť EEPROM pro ukládání přijatého nastavení z počítače. Pravděpodobně nevyžiji technologii ISP, protože zvolenou verzi Arduino Nano je možné programovat přes sběrnici USB.

### 3.1 Požadavky na zařízení a software

Zařízení by mělo být cenově dostupné a jednotlivé součásti zařízení by mělo být možné běžně zakoupit v internetovém nebo kamenném obchodu.

Zařízení by mělo být schopné směřovat laserový modul v horizontálním a vertikálním směru. Není nutné pohybovat se v rozsahu 360°, cílovým smyslem zařízení je prosvícení plochy před zařízením, z tohoto důvodu postačí směřování v rozsahu 180° horizontálně. Vzhledem k tomu, že není třeba vyplnit celou plochu pomyslné polokoule, tak úhly natočení, které se blíží k 180° nebo 0° ve vertikálním směru nejsou pro smysl použití zařízení potřebné.

Jako laserový modul může být použit běžný červený laser. Zařízení musí být schopné komunikace s počítačem. Napájení zařízení není nutné řešit zvlášť a je možné zařízení napájet přímo ze sběrnice USB, pomocí které bude zařízení připojováno k počítači.

Mikropočítač zařízení bude obsahovat řídicí software, který by měl umožňovat vzdálené nastavování z počítače pomocí proměnných parametrů, komunikace s počítačem bude realizována například přes sériovou linku.

Software mikropočítače by měl umožňovat více způsobů průchodu laserem vzdálenou plochou, například po řádcích nebo po sloupcích. Mělo by být možné nastavovat různou rychlost průchodu. Krajní limity vzdálené plochy, čtyři body, bude zadávat uživatel, budou moci být proměnné a obrazec vymezený limity nemusí být nutně pravidelný čtyřúhelník.

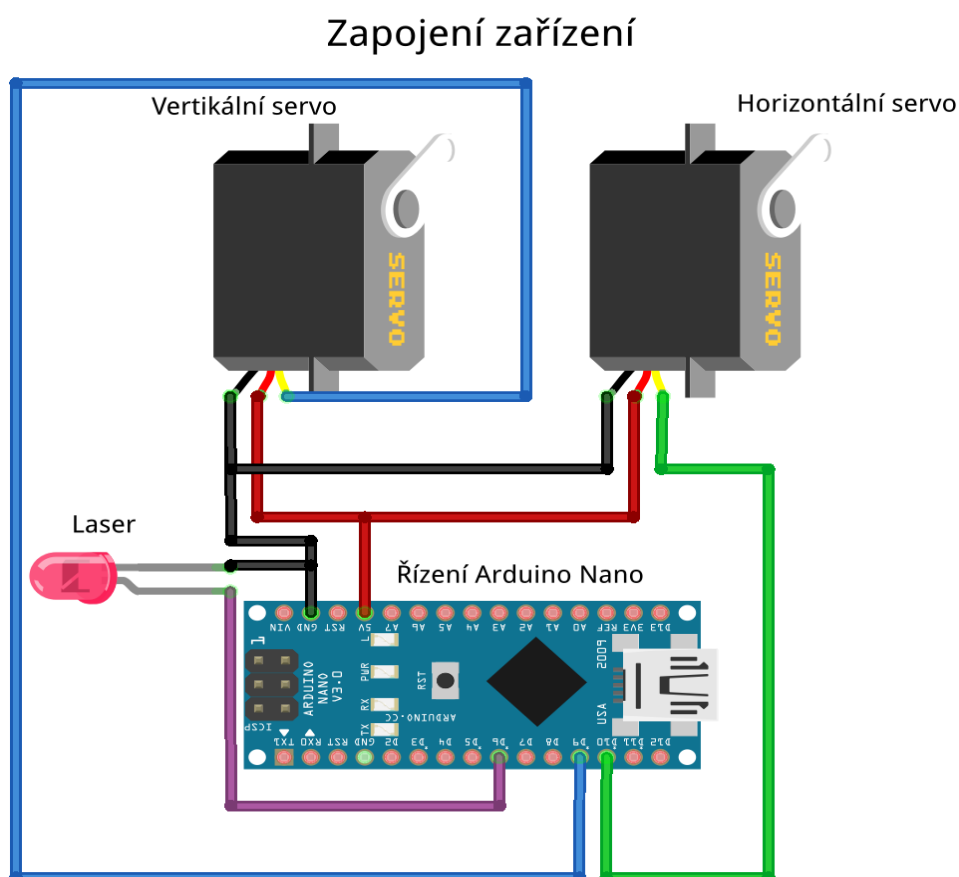
Pro ovládání zařízení by měla být vytvořena aplikace s grafickým uživatelským rozhraním pro platformu PC. Přes tuto aplikaci bude možné vytvořit nastavení pro zařízení, toto nastavení do zařízení odeslat a spustit průchodu. Nastavení krajních limitů obrazce k procházení bude probíhat interaktivně označením krajních bodů laserem na zařízení. Dále bude aplikace umožňovat ukládání jednotlivých nastavení do souborů pro pozdější použití.

Pro potřeby automatizace ovládání zařízení by měla být vytvořena dynamická knihovna pro platformu PC a operační systém Windows. Tato knihovna bude umožňovat zasílání uložených nastavení do zařízení, vytvářet nové nastavení a zaslat jej do zařízení, spouštění průchodů a zastavení průchodu.

## 4 Popis práce

V této kapitole bude vysvětleno vlastní řešení práce na základě požadavků, které byly uvedeny v předchozí kapitole. Bude oddělen popis výsledného zařízení a softwaru pro jeho řízení.

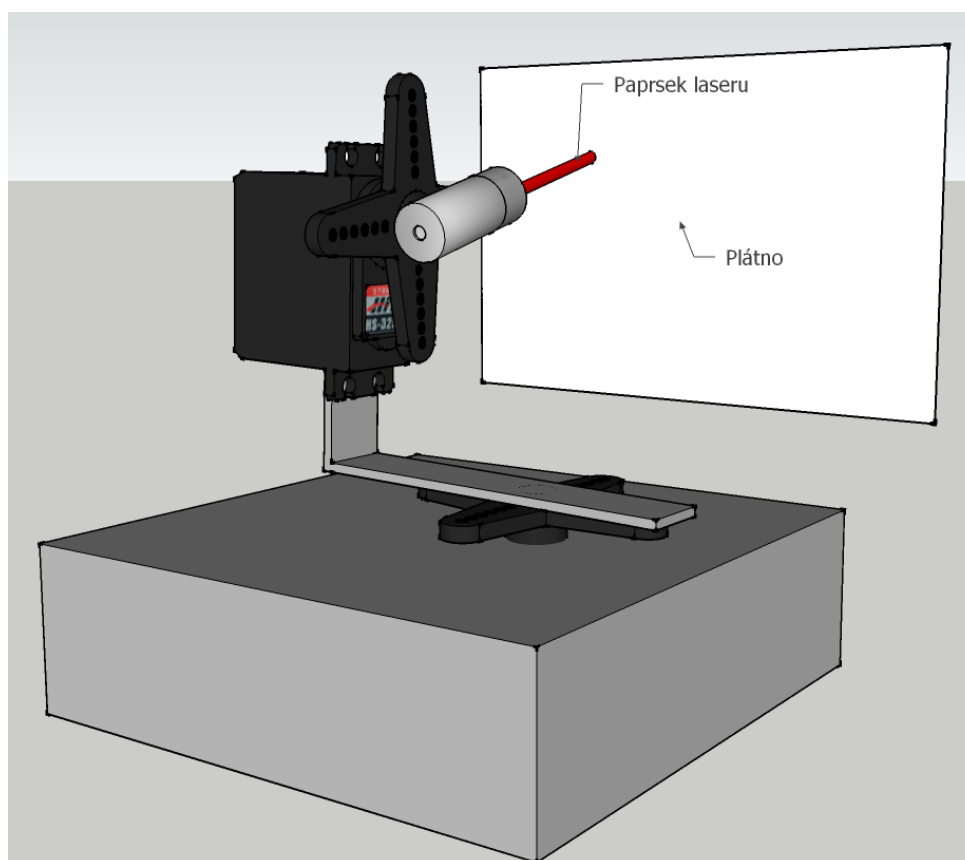
Pro tvorbu zařízení s laserem je třeba řešit problém pohybu v horizontálně-vertikálním směru a řízení. Bude potřeba vyřešit uložení serv tak, aby umožňovala horizontálně-vertikální pohyb s modulem laseru a aby se zařízení samovolně nepohybovalo nebo nepřevracelo. Dále bude třeba vyřešit problém řízení a napájení serv a laseru. K řízení se váže problematika komunikace zařízení s počítačem a nastavování zařízení. Bude třeba vytvořit aplikaci s grafickým uživatelským rozhraním, která přehledným způsobem umožní nastavovat potřebné parametry průchodu a správu uložených nastavení průchodu. Pro komunikaci se zařízením bude třeba řešit problém spojení zařízení s počítačem.



Obrázek 4.1 Navrhované zapojení zařízení

## 4.1 Konstrukce zařízení

Z uvedeného vyplývá, že bude třeba využít dvou serv pro horizontální a vertikální směrování, tato serva by měly být na sobě závisle mechanicky spojená.



Obrázek 4.2: Model konceptu zařízení

Výrazná přesnost serv není nutná, protože pokud uživatel nastaví zpoždění přesunu mezi procházenými body na nízké hodnoty v řádech milisekund až desítek milisekund, tak bude pohyb laserového paprsku prakticky plynulý a nebude třeba pohyb provádět po malých krocích. Vzhledem k výše uvedeným závěrům a požadavku na minimální cenu zařízení, jak bylo uvedeno na úvod práce, jsou použita serva Tower Pro SG90. Jako laserový modul je použit jednoduchý a běžně dostupný modul KY-008.

Serva jsou ve výsledku umístěna v typově nepojmenované pohyblivé plastové základně, která umožní upevnění modulu s laserem na vrchol základny. Základna bude připevněna na plastovém boxu, ve kterém bude uložena řídicí elektronika a podpůrné obvody. Tato základna řeší problém původního konceptu zařízení, kterým bylo vytvoření páky vertikálním servem.

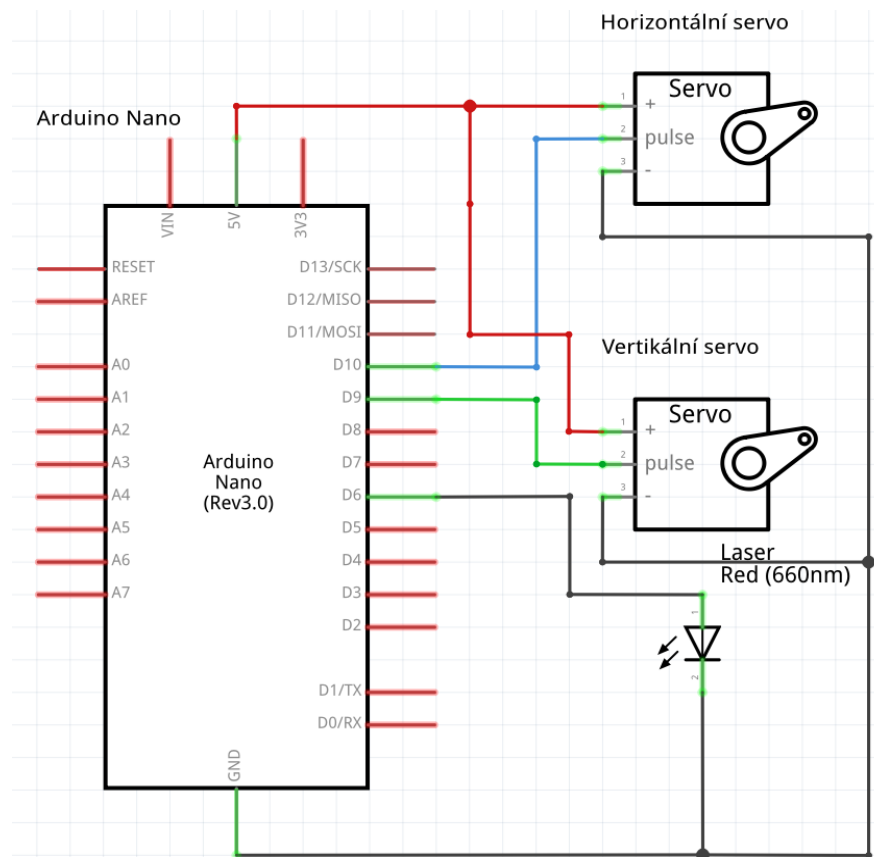


*Obrázek 4.3 Základna se servy*

Pro řízení zařízení je zvoleno Arduino Nano s mini-USB portem. Mikrokontrolér této vývojové desky je naprogramován řídicím programem, který umožňuje komunikaci s počítačem přes sériovou linku emulovanou přes sběrnici USB. Vývojová deska Arduino Nano je vložena do pinového pole na tištěném spoji. Toto umožní případnou výměnu řídicí vývojové desky v případě jejího poškození. Na příslušné napájecí a řídicí piny desky jsou připojena serva a laserový modul.

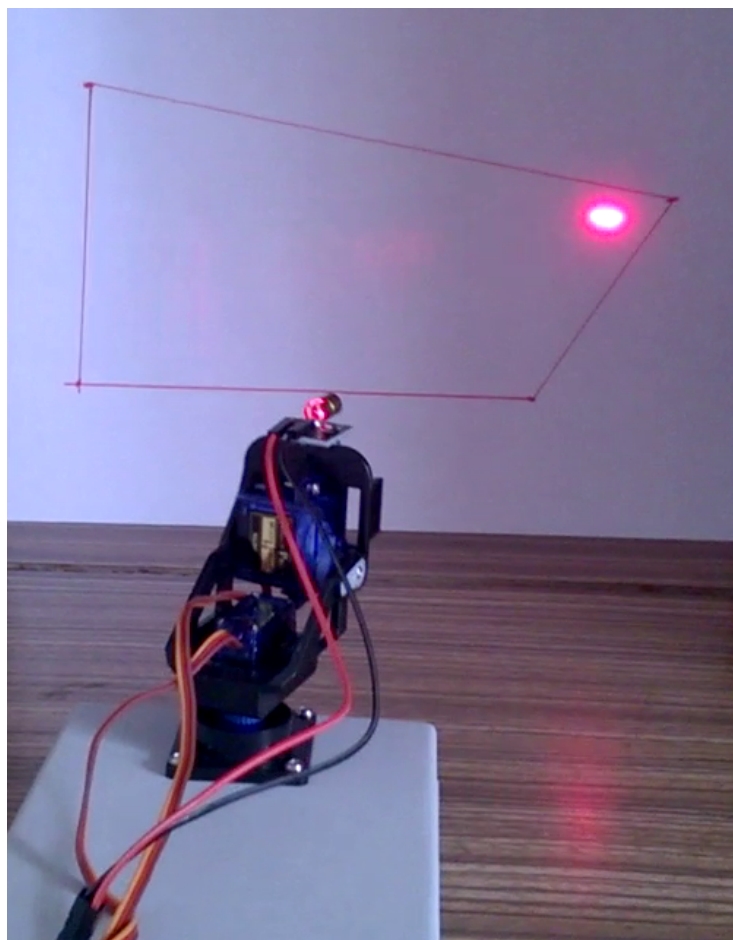
Řízení zařízení je uloženo v plastovém boxu a z boxu jsou vyvedeny pouze potřebné vodiče k servům a laserovému modulu. Pro připojení ke sběrnici USB je umožněn vnější přístup k mini-USB konektoru na desce. Na následujícím obrázku je výsledné schéma zapojení zařízení.





Obrázek 4.4: Výsledné schéma zapojení

Na následujícím obrázku je možné vidět výslednou konstrukci zařízení. Čtenář si může povšimnout použitých serv a jejich uložení. Servo pro horizontální pohyb je uloženo v základně rotorem směrem dolů a pohybuje tak celou konstrukcí základny. Servo pro vertikální směrování je upevněno v horní konstrukci základny a pohybuje připevněným panelem s laserovým modulem. Od zařízení vedou vodiče do boxu, ve kterém jsou tyto připojeny k desce Arduino Nano. Ve výchozí konfiguraci je pro připojení laserového modulu použit pin číslo 6, pro serva to jsou piny 9, pro vertikální servo, respektive pin 10 pro horizontální servo. Analogový šum pro inicializační hodnotu generátoru pseudonáhodných čísel se získává z pinu číslo 7.



*Obrázek 4.5: Výsledné zařízení při průchodu  
nepravidelného čtyřúhelníku*

## 4.2 Návrh softwaru

V této kapitole bude rozebrán návrh softwaru pro mikroprocesor, aplikace pro platformu PC s grafickým uživatelským rozhraním a knihovny pro automatizaci ovládání zařízení.

Pro komunikaci se zařízením je použita sériová linka o rychlosti 9600 baudů za sekundu s jedním stop bitem, osmi datovými bity a bez paritních bitů.

Je nutné poznamenat, že vzhledem k upevnění serv odpovídá maximálnímu natočení doleva úhel  $180^\circ$  a  $0^\circ$  doprava. Natočení zcela nahoru odpovídá úhlu  $0^\circ$  a dolů  $180^\circ$ . Celý program pracuje s těmito hodnotami jako s limity natočení a podle souřadnicového systému, který tyto hodnoty tvoří je také program vytvořen.

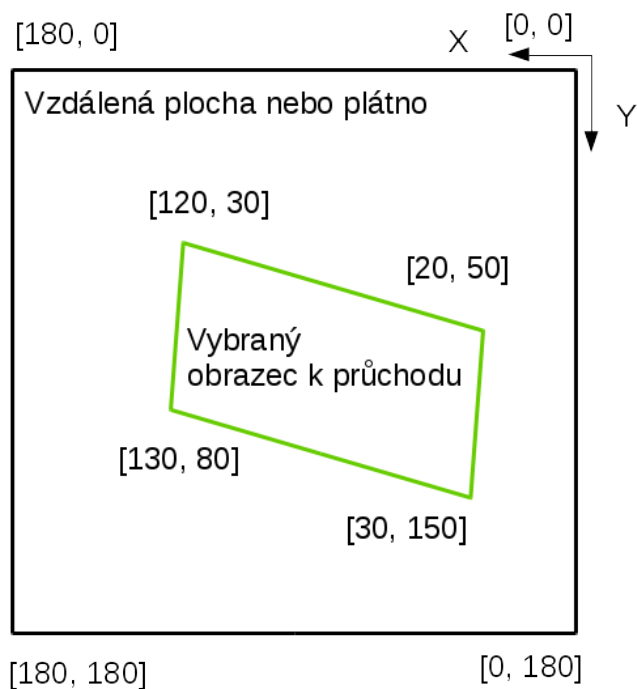
## 4.2.1 Řídící software

Software pro mikropočítač, který řídí celé zařízení je napsán v jazyce C/C++ s využitím knihoven pro platformu Arduino. V tomto případě se jedná především o knihovnu Servo a Arduino. Program je třeba překládat v Arduino IDE. Vlastní program se skládá ze čtyř zdrojových textů. Konkrétně se jedná o soubory **main.ino**, **common.h**, **alg.h** a **alg.cpp**.

Řídící software zajišťuje dvě důležité úlohy. První z nich je komunikace s počítačem a příjem nastavení. Druhou úlohou je řízení serv a laserového modulu.

Řízení laserového modulu je jednoduché a spočívá pouze v zapínání a vypínání laseru podle situace, ve které se zařízení nachází. Typicky se tedy při začátku procházení laser zapne a po skončení zase vypne.

Nejdůležitější částí řídicího softwaru je samotné řízení serv. Software umožňuje přizpůsobení nastavení procházení podle několika základních parametrů. Především jde o definování oblasti procházení, kdy není nutné procházet celý souřadnicový systém serv, ale je možné čtyřmi body definovat čtyřúhelník v procházené oblasti, a v takto definované oblasti čtyřúhelníku bude průchod laserem proveden.



Obrázek 4.6: Souřadnicový systém zařízení a poloha ilustračního obrazce k průchodu

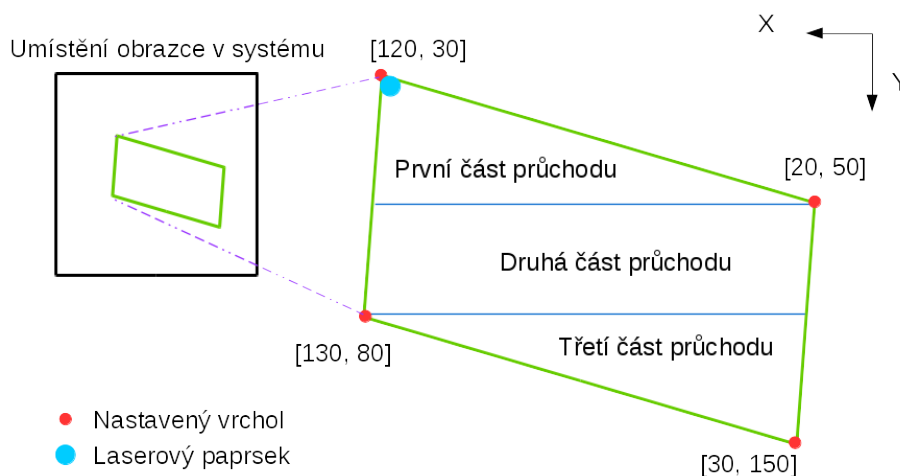
Další možností je nastavení zpoždění průchodu v milisekundách mezi dvěma sousedními body na vzdálené ploše. Tímto má uživatel možnost regulovat rychlost procházení obrazce na vzdálené

ploše. Aby nebylo nutné jedno nastavení opakovaně spouštět manuálně, tak možné nastavit počet opakování průchodu obrazcem. Rychlost a přesnost průchodu je také možné regulovat pomocí nastavení počtu průchodů v jednom obrazci bez závislosti na rozměrech obrazce.

Software umožňuje použít několik implementovaných algoritmů průchodu obrazcem. Tyto algoritmy definují, jakým způsobem se bude laser pohybovat po vzdálené ploše. Uvést lze například horizontální průchod shora dolů, kdy laser bude po řádcích procházet vymezený obrazec na vzdálené ploše od nejnvýše položeného bodu po nejnižší. Vybrané algoritmy umožňují průchod obecným konvexním čtyřúhelníkem. Algoritmy, u nichž není důvod k tomuto přístupu umožňují procházení pouze obdélníkem.

Komunikace s počítačem je založena na spojení přes emulovanou sériovou linku pomocí rozhraní USB. Pro zařízení jsem vytvořil jednoduchou binární instrukční sadu pro řízení a nastavování z počítače.

Významnou funkcí je **getQuadSet**, jelikož na základě přijatého nastavení, především pozic vrcholů čtyřúhelníku, vypočítá všechny potřebné parametry pro průchod obrazcem. Díky tomu je možné, aby byly algoritmy průchodu univerzální a nezávisely na obecných a pevně nastavených parametrech. Celý výpočet cílových parametrů je založený na vzdálenostech souřadnic vrcholů, podobnosti trojúhelníků a poměru stran trojúhelníků. V zásadě jde o potřebu získat koeficienty, které se následně v algoritmech použijí k postupnému výpočtu pozic bodů, mezi kterými se bude procházet laserovým paprskem.



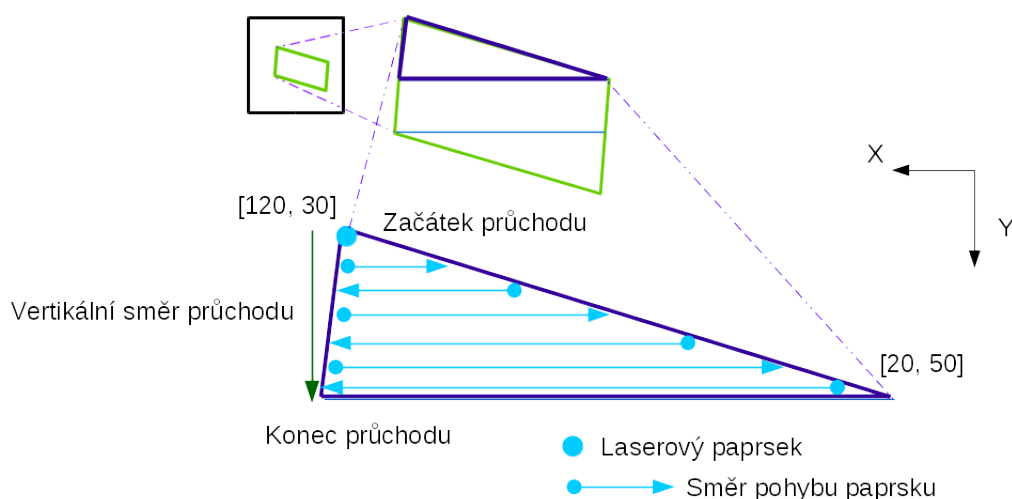
*Obrázek 4.7: Obrazec k průchodu laserem a jeho rozdělení na tři klíčové části*

Na zde uvedené ilustraci je zobrazená procházená část v celém souřadnicovém systému zařízení a rozdělení zadaného obrazce na tři klíčové části. Pro každou takovou část jsou vypočítány vlastní parametry průchodu.

Celé řízení procházení obrazce je možné implementovat několika způsoby, uvést lze například použití prediktorů rozšiřování souřadnic. Na tomto základu by bylo možné konkrétní souřadnici

rozšiřovat vždy, když by byl proveden patřičný počet kroků. V tomto projektu byla zvolena druhá možnost, která využívá aritmetiky v plovoucí řádové čárce. K počátečním souřadnicím jsou přičítány koeficienty přírůstku, tyto koeficienty jsou kladné nebo záporné, podle toho, zda souřadnice roste nebo klesá. Tímto způsobem se dosahuje implicitního omezení nárůstu hodnot souřadnic přes limity obrazce vzhledem k počtu kroků, kterými je obrazec procházen.

Níže je uvedena ilustrace průchodu první části obrazce laserem. Průchod začíná v nejvýše položeném bodě  $[120, 30]$ , vertikální souřadnice stoupá s každým procházeným řádkem a horizontální úsečka, kterou laser prochází se postupně rozšiřuje.



Obrázek 4.8: Ilustrace průchodu laserem první části vymezeného obrazce

Implementované algoritmy pro procházení vymezených obrazců mají za úkol projít vymezený obrazec podle nastavených parametrů patřičným způsobem. Když například uživatel zadá krajní body, které tvoří čtverec a vybere vertikální algoritmus průchodu zleva doprava, tak řídicí software zajistí, že serva budou směřována pouze v limitech tohoto čtverce a zařízení projde laserovým paprskem tento obrazec po sloupcích zleva doprava.

Algoritmy pro vertikální a horizontální průchod sdílí stejný zdrojový kód. To je možné díky výměně souřadnic vrcholů čtyřúhelníku a programové výměně serv. Algoritmy pro průchod laseru po soustředných obdélnících a algoritmus náhodného průchodu vyžadují, aby vrcholy obrazce tvořily obdélník.

Původní souřadnice vrcholu	Nové souřadnice vrcholu
Levý horní $[x, y]$	Pravý horní $[y, x]$
Pravý horní $[x, y]$	Pravý dolní $[y, x]$
Levý dolní $[x, y]$	Levý horní $[y, x]$
Pravý dolní $[x, y]$	Levý dolní $[y, x]$

Tabulka 4.1 Výměna souřadnic vrcholů čtyřúhelníku pro vertikální průchod

## 4.2.2 Instrukční sada zařízení

Pro potřeby vzdáleného řízení zařízení byla vytvořena jednoduchá instrukční sada. Instrukce se předávají v binárním formátu. Každá instrukce může přijímat jiné parametry různé délky, včetně žádných parametrů. Kompletní tabulku instrukcí, včetně podrobného popisu, formátu a kódu instrukcí obsahuje soubor **README.txt** ve složce **main** se zdrojovými kódy programu řízení zařízení. Níže budou tedy uvedeny pouze nejvýznamnější instrukce a informace k nim. Pokud není uvedeno jinak, tak instrukce parametry nepřijímá. Zaslání jakékoli instrukce do zařízení přerušit aktuálně probíhající proces procházení, pokud je spuštěn.

- Instrukce **STOP** (kód 0x00) způsobí zastavení průchodu, instrukce je formální, vzhledem k tomu, že jakákoli přijatá data na sériové lince způsobí zastavení průchodu. Nepřijímá žádné parametry.
- Instrukce **START** (kód 0x01) zadá zařízení příkaz ke kontrole přijatého nastavení a pokud nastavení neobsahuje chyby, tak bude spuštěn nastavený průchod.
- Instrukce **LSTOP** (kód 0x02) zastaví průchod a vypne laser.
- Instrukce **LSTART** (kód 0x03) spustí laser.
- Instrukce **DELAY** (kód 0x04) zašle do zařízení 16 bitové číslo, které reprezentuje zpoždění v milisekundách při průchodu serva mezi dvěma body. Parametrem je 16 bitové celé číslo bez znaménka.
- Instrukce **LT** (kód 0x20) nastaví pozici levého horního vrcholu čtyřúhelníku k procházení. Společně s instrukcí se zasílají dvě osmi bitová čísla bez znaménka, která reprezentují horizontální a vertikální souřadnici vrcholu, v tomto pořadí. Ve výchozím nastavení musí být hodnoty v mezích 0 až 180. Pokud hodnota přesáhne limit, tak bude oznámena chyba instrukcí **RANERR** (kód 0xFE). Stejně pracují instrukce **RT**, **LB**, **RB**.
- Instrukce **GLT** (kód 0x30) používá se, když uživatel pouze zaměřuje body, které chce označit za vrcholy, v tomto případě pro levý horní vrchol. Instrukce má stejné parametry jako instrukce **LT**, způsobí však pouhé nasměrování laseru na zadaný bod.
- Instrukce **OK** (kód 0xE0) oznamuje úspěch provedení akcí.
- Instrukce **HTB** (kód 0x50) nastaví algoritmus průchodu na horizontální shora dolů.
- Instrukce **HBT** (kód 0x51) nastaví algoritmus průchodu na horizontální zdola nahoru.
- Instrukce **VLR** (kód 0x52) nastaví algoritmus průchodu na vertikální zleva doprava.
- Instrukce **VRL** (kód 0x53) nastaví algoritmus průchodu na vertikální zprava doleva.
- Instrukce **REM** (kód 0x54) nastaví algoritmus průchodu na soustředné obdélníky od středu ke krajům.
- Instrukce **REC** (kód 0x55) nastaví algoritmus průchodu na soustředné obdélníky od krajů ke středu.
- Instrukce **RAN** (kód 0x56) nastaví algoritmus průchodu na náhodný.

### 4.2.3 Knihovna pro vzdálené řízení

Za účelem možnosti integrovat ovládání zařízení do aplikace třetí strany byla vytvořena sdílená dynamická knihovna pro operační systém Microsoft Windows. Tato knihovna je zároveň použita pro komunikaci se zařízením prostřednictvím vytvořené aplikace. Pro tvorbu knihovny byl použit programovací jazyk C/C++ a knihovna využívá prostředky z Windows API pro komunikaci se zařízením přes sériovou linku. Návod na přeložení knihovny je zapsán v souboru **ReadMe.txt** ve složce se zdrojovými texty knihovny. Pro následné použití je třeba knihovnu zkopírovat do složky s cílovou aplikací.

Aplikace třetí strany má možnost importovat sdílenou knihovnu **servolaser\_lib.dll**. Po importování dostává taková aplikace přístup ke sdíleným funkcím knihovny a má tak možnost pomocí volání těchto funkcí ovládat vytvořené zařízení. Je tak možné, aby volala například funkci **SLStartFile** a spustila procházení na základě souboru s nastavením.

Funkcionalita knihovny obsahuje možnost spuštění procházení na základě přijatého nastavení nebo podle nastavení, které je uloženo v konfiguračním souboru. Takto lze proces procházení automatizovat, aniž by uživatel použil aplikaci s grafickým uživatelským rozhraním. V souvislosti s tímto obsahuje knihovna funkci pro jednoduché zaměřování laseru na zadanou pozici a to kvůli vizualizaci zaměření. Data použitá pro zaměřování nejsou ukládána, je pouze zaslána instrukce do zařízení a to zaměří laser na zadanou pozici.

Možné je také vytvořené nastavení uložit do konfiguračního souboru. Uživatel tak má možnost libovolná nastavení ukládat a to bez závislosti na grafické aplikaci. Data, se kterými se pracuje, ať už při ukládání do souboru nebo při odesílání do zařízení, jsou kontrolována tak, aby splňovala všechny podmínky pro korektní funkčnost zařízení. Knihovna tak kontroluje rozsahy data, ještě před odesláním do zařízení.

Knihovna umožňuje současné použití ve více aplikacích. Je tak možné, aby jedna aplikace spustila procházení laserem a druhá aplikace tento průchod ukončila. Tento přístup závisí na kooperativním sdílení sériové linky více procesy. Proces, který chce zařízení ovládat má možnost otevřít spojení se zařízením a následně s tímto pracovat, pokud jej neovládá jiný proces. Po vzdání se kontroly nad zařízením nebo zániknutí procesu je možné, aby o přístup požádal jiný proces. Pro kontroly aktivity zařízení se používají knihovnou sdílená data pro všechny procesy.

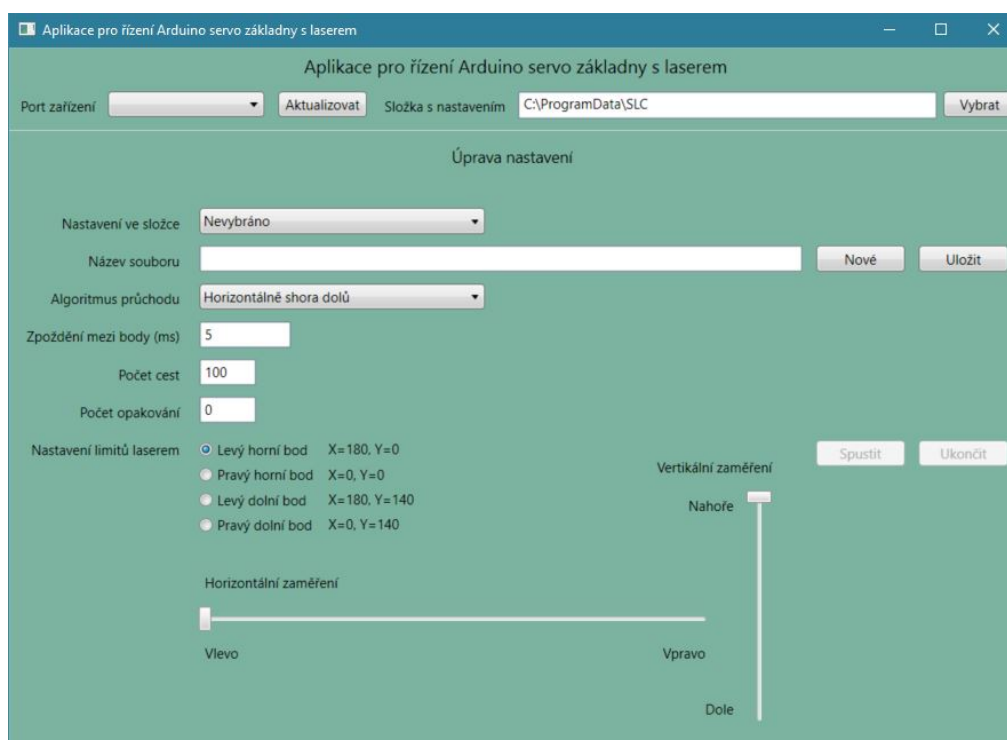
Při navazování spojení se zařízením je třeba vždy nejdříve vyprázdnit vstupní a výstupní vyrovnávací paměť sériové linky a před následným odesláním nových dat do zařízení čekat alespoň 400 milisekund. Tento proces ustavení spojení je implementován v metodě objektu, který používá knihovna používá pro řízení komunikace na sériové lince a jej třeba provádět explicitně.

### 4.2.4 Aplikace pro vzdálené řízení

Účelem aplikace s grafickým uživatelským rozhraním je poskytnout uživateli snadný způsob rychlého, interaktivního nastavení a spuštění zařízení. A také uživateli zpřístupnit funkcionalitu pro vytváření a ukládání vlastního nastavení do počítače. Uživatel má tak možnost organizovat

svoje nastavení uložením do souboru v libovolné složce a pojmenováním tohoto souboru. Aplikaci je možné provozovat i bez připojeného zařízení. V takovém případě aplikace umožňuje spravovat uložená nastavení a vytváření nových. Při vytváření aplikace bylo záměrně cíleno na zcela jednoduché uživatelské rozhraní, které neobsahuje rušivé prvky.

Aplikace je založena na platformě .NET Framework s využitím technologie Windows Presentation Foundation a jazyka C#, cílí tedy na operační systém Microsoft Windows. Ovládání zařízení je zprostředkováno sdílenou dynamickou knihovnou, která je popsána v předchozí podkapitole.



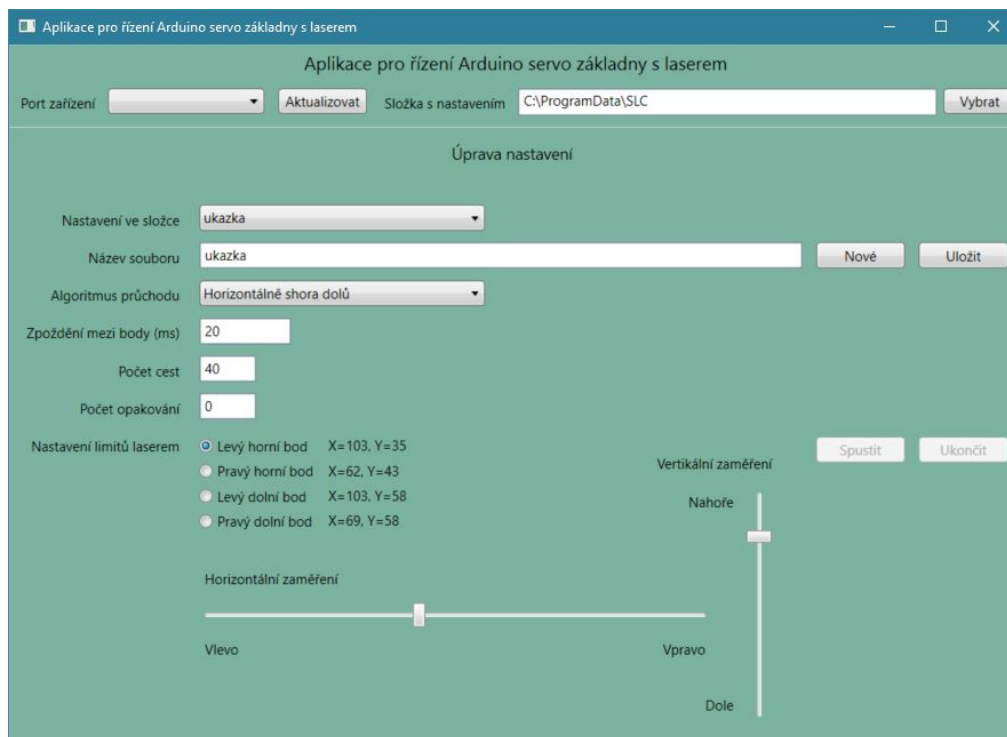
Obrázek 4.9 Aplikace pro vzdálené řízení ve výchozím nastavení po spuštění

Grafické uživatelské rozhraní aplikace umožňuje uživateli interaktivní nastavování limitních bodů obrazce v případě připojeného zařízení. Tímto se rozumí to, že je pohyb prvků grafického rozhraní převáděn na příkazy, které jsou následně zasílány přes sériovou linku do zařízení. Zařízení nastaví zaměření laseru na daný bod a uživatel tak může přímo vidět, v jakých mezích bude průchod laserem probíhat.

Dalšími důležitými ovládacími prvky jsou textová pole pro podrobnější parametrizování průchodu laseru. Lze zadat zpoždění v milisekundách mezi dvěma body, toto zpoždění se používá pro zpomalení pohybu serv mezi sousedními body. Počet cest udává počet procházených přímků v obrazci. Tento parametr umožňuje více členité dělení obrazce. Počet opakování rovnající se nule znamená jeden průchod, je tedy bez opakování, vyšší hodnoty parametru znamenají skutečný počet



opakování průchodu. Limity laseru se zadávají jako body, které lze zaměřovat posuvníky pro horizontální a vertikální zaměřování.



Obrázek 4.10 Aplikace pro vzdálené řízení při nastavování parametrů průchodu

Při spuštění programu jsou parametry inicializované na výchozí hodnoty a není třeba je zadávat. Grafické uživatelské rozhraní uživateli oznamuje případné chybně zadané hodnoty a rozsahy hodnot, které lze zadat. Na výše uvedeném obrázku je vidět nastavení, které bylo použito pro tvorbu souboru s nastavením **ukazka.dat**. Tento soubor je uložen na paměťové kartě, přiložené k této práci, ve složce **bin**. Zároveň je na zmíněném médiu uloženo video **ukazka.mp4**, ve složce **video**, které zachycuje chování zařízení při takto naprogramovaném průchodu.

## 4.2.5 Formát souborů s nastavením

Vytvořené nastavení je možné ukládat do jednoduchých konfiguračních souborů. Data v těchto souborech jsou uložena v textovém formátu, vždy jedna položka konfigurace na jednom řádku a typ konfigurační položky a data této položky jsou oddělena dvojtečkou. Cílem tohoto přístupu je snadná přenositelnost a čitelnost zapsaných dat.

Pro ilustraci lze uvést příklad zapsání souřadnic jednoho z limitních bodů obrazce k procházení. Řetězec „LTC:103:35“ udává, že souřadnicí levého horního rohu LTC obrazce je bod **103** horizontálně a **35** vertikálně. Následující tabulka zobrazuje možná konfigurační pole souboru

a jejich rozsahy. Ve všech případech se jedná o celočíselné hodnoty. Na základě znalosti těchto informací lze libovolně generovat vlastní konfigurační soubory.

<b>Data pole</b>	<b>Význam</b>	<b>Rozsah</b>
ALG	Typ algoritmu procházení	<0;6>
DEL	Zpoždění mezi dvěma body v milisekundách	<0;5000>
PAT	Počet cest v obrazci	<1;255>
REP	Počet opakování průchodu	<0;255>
INF	Povolení informačních výpisů	<0;1>
LTC	Souřadnice levý horní bod	<0;180>
RTC	Souřadnice pravý horní bod	<0;180>
LBC	Souřadnice levý dolní bod	<0;180>
RBC	Souřadnice pravý dolní bod	<0;180>

*Tabulka 4.2: Datová pole konfiguračního souboru*

V další tabulce jsou uvedena číselná vyjádření algoritmů průchodů, která je nutné znát pro případné vytváření konfiguračních souborů aplikací třetí strany. Zde je nutné uvést, že v poslední tři algoritmy vyžadují souřadnice, které tvoří obdélník. Kontrola na geometrický útvar, který body tvoří je prováděna i samotným zařízením a v případě chybných dat nebude procházení zahájeno.

<b>Číselné vyjádření</b>	<b>Odpovídající algoritmus průchodu</b>
0	Horizontálně shora dolů
1	Horizontálně zdola nahoru
2	Vertikálně zleva doprava
3	Vertikálně zprava doleva
4	Po obdélnících od středu ke krajům
5	Po obdélnících od krajů ke středu
6	Náhodně v zadaných limitech

*Tabulka 4.3: Číselná vyjádření algoritmů*

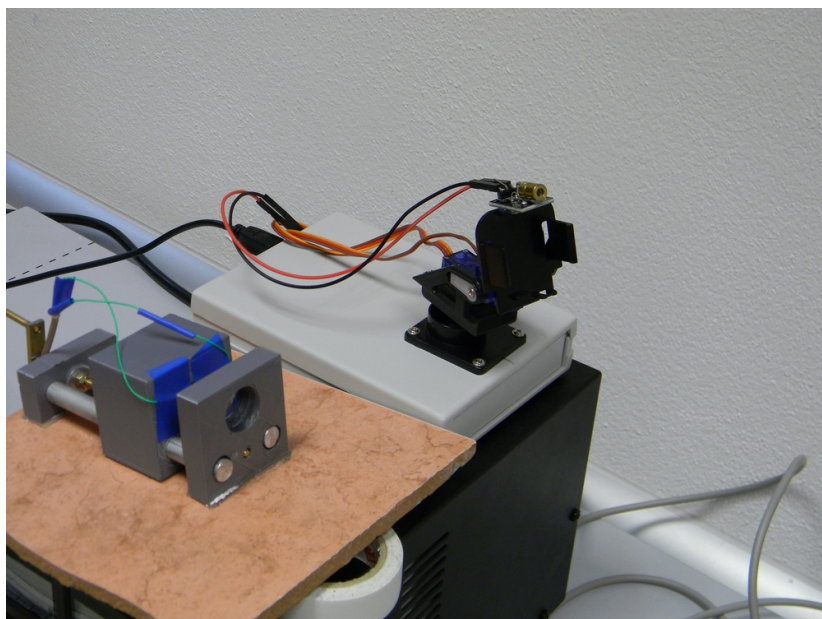
## 4.3 Testování a využití

Základní testování probíhalo způsobem, který je zachycen na přiložené video nahrávce **ukazka.mp4**. Zařízení bylo položeno na rovnou plochu a nasměrováno přední stranou s laserovým modulem na plochu, na které bylo testováno procházení čtyřúhelníkových obrazců s tupými i ostrými úhly mezi přilehlými stranami a s odlišnými délkami stran. Na těchto obrazcích byly zároveň testovány i dostupné algoritmy procházení s různým nastavením. V tomto ohledu se jednalo o nastavení počtu cest k procházení v obrazci, počet opakování algoritmu a především testování nastavení zpoždění serv při průchodu mezi dvěma body. Tento parametr je velmi důležitý, jelikož umožňuje regulovat plynulost průchodu obrazce. V tomto ohledu bylo dosaženo zjištění, že plynulý průchod umožňuje nastavení zpoždění v širším rozsahu 3 až 70 milisekund.

Z tohoto důvodu je nastaveno 5 milisekund jako výchozí hodnota zpoždění ve všech zdrojových textech. Pomocí kombinace uvedených nastavení se podařilo ověřit, že zařízení skutečně korektně prochází laserem zadaný obrazec. V případě testování algoritmů, které umožňují procházení pouze obdélníků, respektive čtverců, bylo dosaženo stejných zjištění. Specifickým algoritmem je algoritmus náhodného průchodu. Tento algoritmus byl testován ještě při vytváření řídicího algoritmu, kvůli generovaným náhodným proměnným.

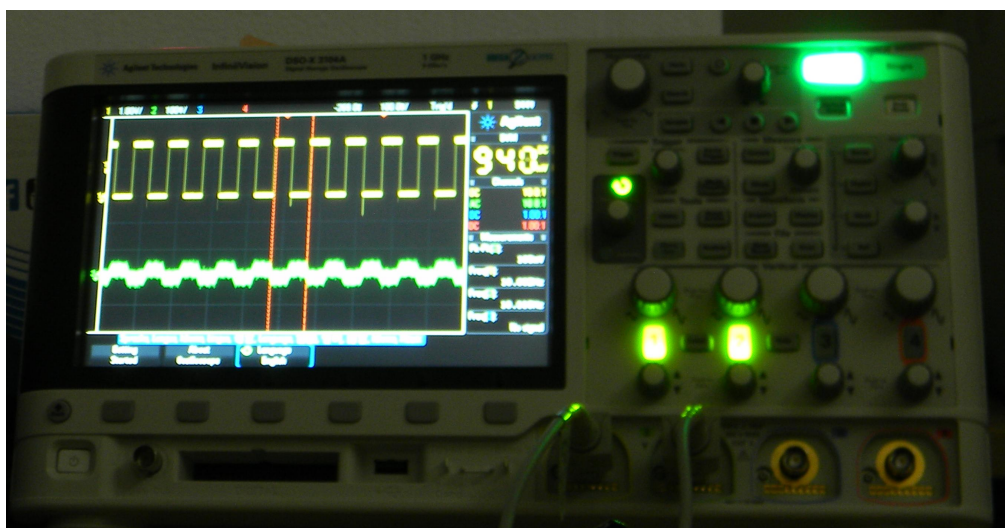
Nepřesnosti průchodu závisí především na kvalitě fyzického provedení zařízení. Nebudeme-li uvažovat rozmítanost laserového paprsku, tak se jedná především o kvalitu provedení serv a převodů v těchto servech. Z důvodu cílení na nízkou cenovou hladinu jsou v projektu použity levná serva s nízkým rozlišením a plastová převodová soustava. Podle dokumentačního listu k servu Tower SG90 je rozlišení tohoto serva jeden stupeň. Tuto přesnost bude ovšem navíc degradovat fakt použití nepřesné plastové převodové soustavy. Zde lze uvažovat nepřesnost výroby a opotřebení celé soustavy. Řešením by bylo použití kovové převodové soustavy a serv s vyšším rozlišením. Toto by ale zvyšovalo cenu zařízení. Při plynulém průchodu není přesnost serv výrazně důležitá. Částečnou náhradou za přesnost může být možnost nastavení více cest pro procházení v obrazci, než je výška obrazce. Nižší přesnost zařízení se může výrazně projevit při procházení vzdálenějších ploch, v tomto ohledu by se mohlo jednat o vzdálenosti větší, než 3 metry.

Druhé testování potvrdilo, že zařízení ve spolupráci s laserovým detektorem korektně pracuje. Tato část testování je zdokumentována na přiložených video nahrávkách a fotografiích ve složce **video/test2**. Vytvořené zařízení bylo připojeno k počítači a ovládáno přes vytvořenou aplikaci s grafickým uživatelským rozhraním. Vedle zařízení byl položen senzor laserového paprsku **obrázek 4.11**, který byl zaměřen na neznámé místo na jeden metr vzdálené ploše.



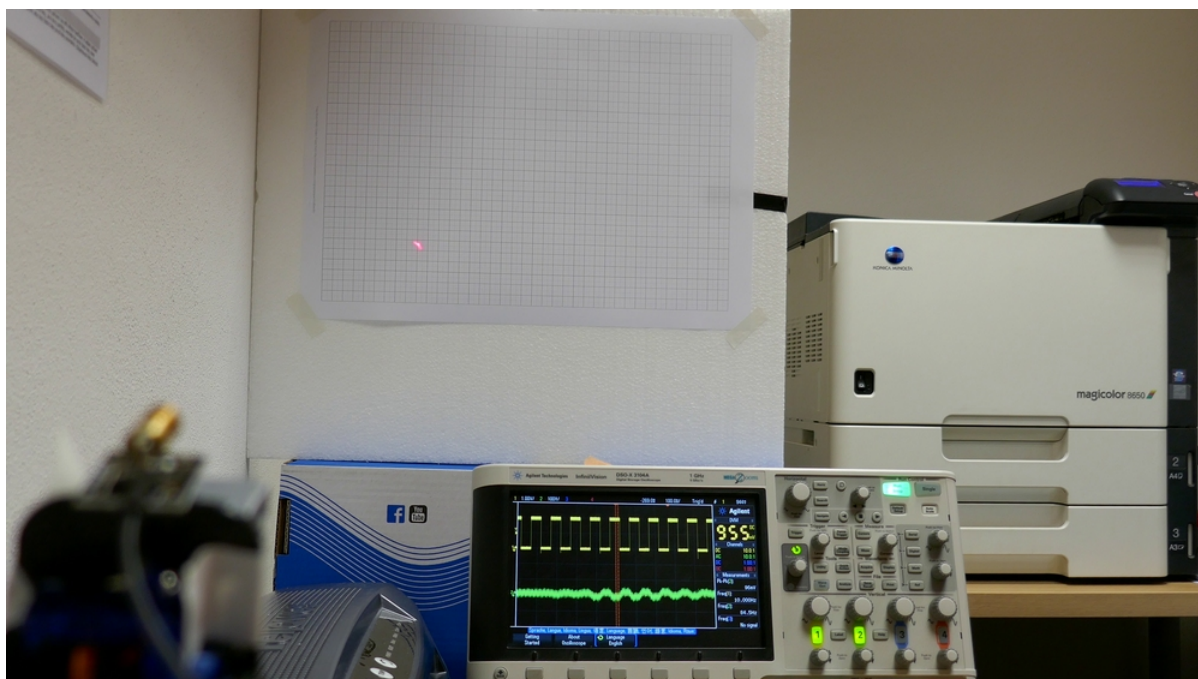
Obrázek 4.11: Vytvořené zařízení společně s detektorem laseru  
(šedé zařízení)

Pro další testování byl laser modulován frekvencí 10Hz. Detektor laserového paprsku byl připojen na osciloskop. Při zaznamenání průchodu laserového paprsku přes místo na vzdálené ploše, na které byl zaměřen detektor, se na osciloskopu zobrazují zákmity (dolní průběh), jak ukazuje **obrázek 4.12**. Na základě odezvy detektoru je tak možné zjistit, jakou oblast pokrývá detektor.



Obrázek 4.12: Odezva na osciloskopu při průchodu laserem přes  
zaměřené místo

Při testování bylo měněno zaměření detektoru i oblast průchodu laserem. Při dalším průchodu laserem tak celá aparatura vypadala tak, jak zachycuje následující obrázek. Zaměření detektoru je přibližně v oblasti, na kterou je zaměřen laser (červená tečka). Toto viditelně detekuje osciloskop (dolní průběh). Snímek je pořízen z videa **P1000749.MP4** v čase 1 minuta a 10 sekund.



*Obrázek 4.13: Aparatura při nalezení zaměření detektoru*

Testování potvrdilo již uvažované nedostatky zařízení, které byly, kvůli zaměření na nízkou cenu, zanedbány. Oblast zaměření detektoru lze zjistit a následně upravovat, kvůli hrubému kroku serv není ale možné detekovat oblast s takovou jemností, aby bylo přesně možné definovat detektorem zaměřenou oblast. Řešení tohoto problému je uvedeno na začátku této podkapitoly. Při testování se také ukázalo, že při průchodu je vhodné použít vyšší zpoždění mezi kroky serv. Experimentálně bylo zjištěno, že vhodné hodnoty se pohybují v rozsahu 100 až 500 milisekund.

## 5 Závěr

Cílem této práce bylo nastudování literatury o laserových diodách, modelářských servech a jejich řízení s pomocí mikropočítače. Následně bylo třeba navrhnout zařízení s podpůrným softwarem, které by usnadnilo zaměřování pozice detektoru laserového paprsku na vzdálené ploše. Tento cíl byl splněn, jak dokazuje **kapitola 4.3** o testování zařízení. V rámci této práce tedy vzniklo zařízení pro pohyb laserovým modulem za pomoci serv a software pro řízení tohoto zařízení a pro vzdálené ovládání zařízení z počítače.

V rámci práce a jejího zadání jsem prostudoval texty, které se zabývají především řízením modelářských serv a jejich vlastnostmi. Díky tomuto jsem se dozvěděl, jak serva pracují, jaká jsou jejich omezení a jak je možné je řídit pomocí mikropočítačů a vzdáleně pomocí počítače. Na základě těchto získaných znalostí jsem navrhl zamýšlené zařízení z úvodní kapitoly tohoto textu a způsob ovládání tohoto zařízení, jak z hlediska řídicího softwaru, tak z hlediska vzdálené komunikace s tímto zařízením. Na tomto základu jsem vytvořil zařízení a podpůrný software pro ovládání tohoto zařízení. Zařízení, jeho vlastnosti a řízení jsem popsal v této práci a software jsem zdokumentoval ve zdrojových textech.

Další práce a vylepšování zařízení by mohlo spočívat ve zvýšení kvality jeho hardwaru. Zde by se jednalo o odstraňování nedostatků, které jsou uvedeny v předcházející kapitole o testování. Současná serva a převodové soustavy by se nahradila za kvalitnější modely. Odstranily by se tak problémy s nepřesností a hrubým průchodem, bylo by pak možné přesněji definovat oblast zaměření detektoru. Určitou formou vylepšení by také mohly být vyměnitelné laserové moduly. Řešení by spočívalo ve vytvoření mechanismu, do kterého by se moduly vkládaly a upevnění modulu by mohlo být realizováno jeho sevřením v tomto mechanismu. Podmínkou by byla nízká váha a možnost upevnění mechanismu na vrchol pohyblivé základny. Základnu by také bylo možné vymodelovat na 3D tiskárně a společně s výše uvedenými vylepšeními by bylo zařízení celkově kvalitnější po stránce hardwaru. Při případných vylepšeních hardwaru by bylo možné zachovat řídicí software. Vylepšení řídicího softwaru by mohlo spočívat v přidávání nových algoritmů procházení a řízení modulace laseru. Počítačová aplikace pro řízení zařízení by mohla být vylepšena například grafickým znázorněním útvaru, který je tvořen zaměřenými body.

Následujícím krokem by mohlo být spojení a zajištění synchronizace detektoru s vytvořeným zařízením. Cílem by bylo automatizovat úpravu zaměření detektoru na základě informace o pozici detekovaného laserového paprsku a požadovaném zaměření detektoru.

# Literatura

- [1] Arduino. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-05-15]. Dostupné z: <https://cs.wikipedia.org/w/index.php?title=Arduino>
- [2] *Arduino* [online]. Ivrea: Arduino, 2017 [cit. 2017-05-15]. Dostupné z: <https://www.arduino.cc>
- [3] AVR Libc. *AVR Libc Home page* [online]. Boston: Free Software Foundation, 2016 [cit. 2017-05-15]. Dostupné z: <http://www.nongnu.org/avr-libc/user-manual/index.html>
- [4] *Atmel* [online]. San Jose: Atmel Corporation, 2017 [cit. 2017-05-15]. Dostupné z: <http://www.atmel.com/default.aspx>
- [5] Laserové diody. *Elektrorevue* [online]. Brno: International Society for Science and Engineering, o.s., 2001 [cit. 2017-05-15]. Dostupné z: <http://www.elektrorevue.cz/clanky/01034/index.html>
- [6] Řízení serva. *Robotika.cz* [online]. Robotika.cz, 2005 [cit. 2017-05-15]. Dostupné z: <http://robotika.cz/guide/servo/cs>
- [7] SKALICKÝ, Jiří. Elektrické servopohony. Vyd. 2. Brno: Vysoké učení technické, 2001. ISBN 80-214-1978-4.
- [8] *MSDN Library* [online]. Redmond: Microsoft Corporation, 2017 [cit. 2017-05-15]. Dostupné z: <https://msdn.microsoft.com>
- [9] HW server představuje - Sériová linka RS-232. *HW server* [online]. Praha: HW server, 2005 [cit. 2017-05-15]. Dostupné z: <http://vyvoj.hw.cz/rozhrani/hw-server-predstavuje-seriova-linka-rs-232.html>
- [10] Introducing Windows Presentation Foundation. *MSDN Library* [online]. Redmond: Microsoft Corporation, 2006 [cit. 2017-05-15]. Dostupné z: <https://msdn.microsoft.com/en-us/library/aa663364.aspx>
- [11] What is “managed code”? *Docs Microsoft* [online]. Redmond: Microsoft Corporation, 2016 [cit. 2017-05-15]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/articles/standard/managed-code>
- [12] *Jiangsu Qinheng* [online]. Nanjing: Jiangsu Qinheng Co., 2016 [cit. 2017-05-15]. Dostupné z: <http://wch-ic.com/>
- [13] *Advanced Monolithic Systems* [online]. New York: Advanced Monolithic Systems, 2017 [cit. 2017-05-15]. Dostupné z: <http://ams-semitech.com/page2.php>
- [14] XAML Overview (WPF). *MSDN Library* [online]. Redmond: Microsoft Corporation, 2017 [cit. 2017-05-15]. Dostupné z: [https://msdn.microsoft.com/en-us/library/ms752059\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms752059(v=vs.110).aspx)
- [15] What is the difference between a DC motor and servo motor? *Handy Board* [online]. Massachusetts: GODADDY.COM, 2017 [cit. 2017-05-15]. Dostupné z: <http://handyboard.com/hb/faq/hardware-faqs/dc-vs-servo/>
- [16] *Tower Pro* [online]. Taiwan: Tower Pro Pte, 2014 [cit. 2017-05-15]. Dostupné z: <http://www.towerpro.com.tw/>

- [17] SureLock™ 633nm Wavelength Stabilized Laser Diode. *Ondax* [online]. Monrovia: Ondax, 2011 [cit. 2017-05-15]. Dostupné z: <http://www.ondax.com/products/surelock-wavelength-stabilized-lasers/surelock-wavelength-stabilized-collimated-to-can-lasers/surelock-633nm-wavelength-stabilized-laser-diode>
- [18] Arduino Pro Mini. In: *So-net* [online]. Tokio: Sony Network Communications, 2014 [cit. 2017-05-15]. Dostupné z: [http://makers-with-myson.blog.so-net.ne.jp/upload/detail/m\\_Arduino20UNO20vs20Nano20vs20Pro20Mini.JPG.html](http://makers-with-myson.blog.so-net.ne.jp/upload/detail/m_Arduino20UNO20vs20Nano20vs20Pro20Mini.JPG.html)
- [19] BURNETT, Colin M.L. Jedno řídicí (master) a tři podřízená (slave) zařízení na sběrnici SPI. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-05-15]. Dostupné z: [https://en.wikipedia.org/wiki/File:SPI\\_three\\_slaves.svg](https://en.wikipedia.org/wiki/File:SPI_three_slaves.svg)



# Seznam příloh

Příloha 1. Paměťová karta typu SD s tímto textem, spustitelným programem, zdrojovými kódy a video a foto ukázkami práce zařízení.