

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická

BAKALÁŘSKÁ PRÁCE

Laserový projektor

2008

Jan Dvořáček

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze, dne 27. 6. 2008


.....
podpis

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Jan Dvořáček**

Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný
Obor: Kybernetika a měření

Název tématu: **Laserový projektor**

Pokyny pro vypracování:

1. Seznamte se procesory řady Atmel AVR.
2. Navrhněte vhodné řešení, které umožní rozmítání paprsku laserového projektoru. Řešení bude sloužit k promítání obrázků nebo textů.
3. Navrhněte a realizujte vhodný HW s procesorem Atmel AVR, který umožní řízení na základě příkazů po sériové lince.
4. Implementujte SW pro mikrokontrolér případně i pro PC, který umožní demonstraci celého zařízení.

Seznam odborné literatury:

Dodá vedoucí práce

Vedoucí: Ing. Pavel Němeček

Platnost zadání: do konce zimního semestru 2008/2009


prof. Ing. Michael Šebek, DrSc.
vedoucí katedry




doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 25. 2. 2008

Abstrakt

Tato práce popisuje problematiku projekce maticového obrazu či textu pomocí jednoho bodového zdroje světla a realizaci projektoru pracujícím na tomto principu. Jako zdroj světla byl použit LED laser. Zařízení řídí procesor řady AVR od firmy Atmel. K rozmitání laserového paprsku do matice bodů slouží kotouč se zrcátky po obvodu. K ovládání zařízení slouží sériová linka. V práci je popsán princip projekce a dokumentace k demonstračnímu modelu.

Klíčová slova

Laser, projekce, maticový text.

Abstract

In this bachelor thesis is described research and realization of projection images or text by matrix of dots using one light source. On functional model of this device were used LED laser like light source and AVR Microcontroller. For sweeping laser beam were used wheel with small mirrors on perimeter. Device can be controlled by serial line. This thesis contains description of projection principle and documentation of functional model.

Keywords

Laser, projection, matrix text.

Obsah

1 Úvod	7
1.1 Rozbor zadání.....	7
1.2 Pokyny.....	7
1.3 Upřesnění funkcí.....	7
2 Návrh řešení.....	8
2.1 Princip vykreslování.....	8
2.2 Diagram časování.....	10
2.3 Blokové schéma.....	11
2.4 Spínání laseru.....	11
2.5 Pohon.....	12
2.6 Synchronizace.....	13
2.7 Mikrokontrolér AVR.....	14
2.8 Vývojové prostředí CodeVision.....	16
2.9 Komunikace.....	17
3 Realizace HW.....	18
3.1 Mechanické provedení.....	18
3.2 Elektronika.....	18
4 Implementace SW mikrokontroléru.....	20
4.1 PWM.....	20
4.2 Regulace otáček.....	20
4.3 Modulace laseru.....	21
4.4 Struktura dat.....	21
4.5 Komunikace s PC.....	22
5 Obslužný SW pro PC.....	23
5.1 Úvod.....	23
5.2 Požadavky.....	23
5.3 Řešení.....	23
5.4 Odeslání dat.....	24
6 Závěr.....	25
7 Použité zdroje a software.....	26
8 Přílohy.....	27
8.1 Rozpiska součástek.....	27
8.2 Schéma zapojení.....	28
8.3 Výkresy DPS.....	29
8.4 Výkres mechaniky.....	30
8.5 Výpis programu mikrokontroléru.....	31
8.6 Obrazová dokumentace.....	35

Obsah CD

Program pro PC
Program mikrokontroléru
Bakalářská práce v PDF
Obrazová dokumentace

Seznam obrázků

obr. 1: náhled rotující mechaniky.....	8
obr. 2: vertikální vychylování paprsku.....	9
obr. 3: horizontální vychylování paprsku.....	9
obr. 4: Diagram časování.....	10
obr. 5: Blokové schéma.....	11
obr. 6: Architektura AVR.....	14
obr. 7: Komunikace standartu SPI.....	15
obr. 8 : Generátor kostry programu a inicializace.....	16
obr. 9: konstrukce optické brány.....	19
obr. 10: Generování PWM.....	20
obr. 11: náhled přípravy dat k vykreslení.....	24
obr. 12: Schéma zapojení.....	28
obr. 13: plošný spoj elektroniky.....	29
obr. 14: konstrukce optické brány.....	29
obr. 15: převodník USB-UART.....	29
obr. 16: Výkres mechaniky.....	30
obr. 17: Hello world.....	35
obr. 18: Funkce.....	35
obr. 19: Fungující prototyp.....	35
obr. 20: Detail mechaniky.....	36

1 Úvod

1.1 Rozbor zadání

Cílem práce bylo rozebrat možnosti pro promítání maticového textu či obrazu pomocí jednoho zdroje světla. K rozmítání světla do bodů obrazu bylo použito několik zrcadel po obvodu rotujícího kotouče. Celé zařízení řídí procesor AVR a je možné jej ovládat sériovou linkou. Jako zdroj světla slouží LED laser.

1.2 Pokyny

1. Seznamte se procesory řady Atmel AVR.
2. Navrhněte vhodné řešení, které umožní rozmítání paprsku laserového projektoru. Řešení bude sloužit k promítání obrázků nebo textů.
3. Navrhněte a realizujte vhodný HW s procesorem Atmel AVR, který umožní řízení na základě příkazů po sériové lince.
4. Implementujte SW pro mikrokontrolér případně i pro PC, který umožní demonstraci celého zařízení.

1.3 Upřesnění funkcí

Rozmítání laserového paprsku by mělo umožňovat vykreslení textu, z toho plyne požadavek alespoň šesti řádků výsledné matice bodů. Matici bodů je nutno vykreslovat alespoň 25 krát za vteřinu, aby bylo dosaženo plynulosti zobrazeného obrazu.

SW mikrokontroléru musí být schopen vykreslit dostatečně rychle celou matici bodů. Dále musí kompenzovat případné nepřesnosti mechanické části vhodnou kalibrací.

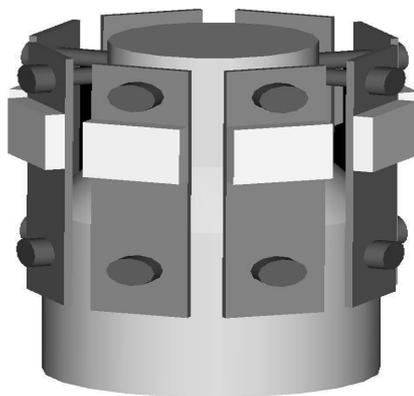
SW pro PC bude sloužit k ovládání zařízení, přípravě a nahrávání dat ke zobrazení a také k nastavení kalibračních konstant zařízení.

2 Návrh řešení

Tato kapitola popisuje technologie a nástroje použité a uvažované při realizaci práce. Některé postupy a řešení byly prostudovány a z různých důvodů nepoužity. Výhody a nevýhody jednotlivých technologií jsou popsány níže.

2.1 Princip vykreslování

Pro vychylování laserového paprsku slouží soustava zrcadel. Po uvážení několika jiných konfigurací byla vybrána konfigurace s takovým počtem zrcadel, kolik bude mít výsledná matice řádků. Zrcadla byla umístěna po obvodu kruhu. Každé zrcadlo pak vychýleno o jiný úhel.



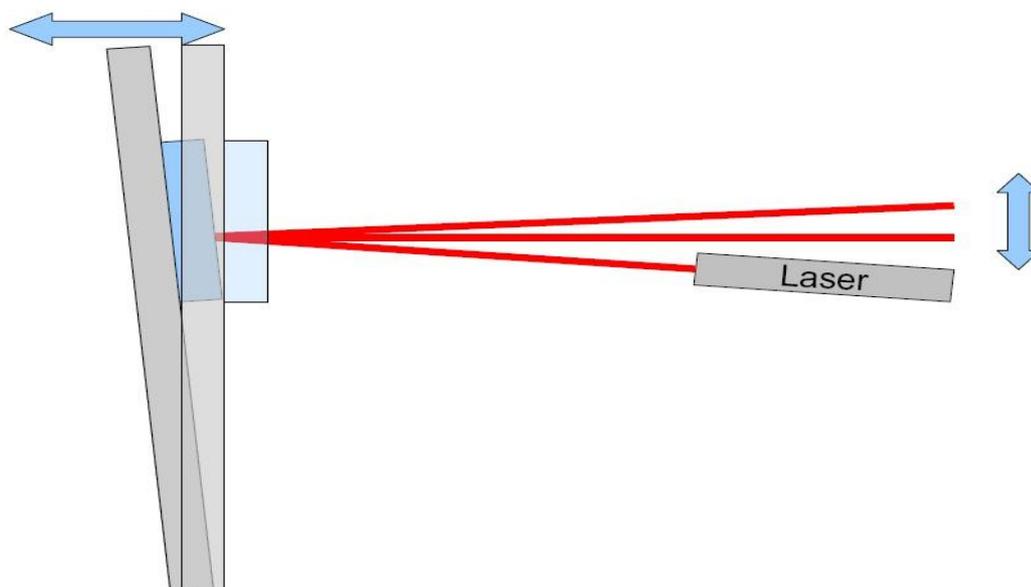
obr. 1: náhled rotující mechaniky

Laserový paprsek svítí na rotující kotouč stále ze stejného směru. Rotací válce se zrcadly je poté dosaženo rozmítání paprsku do matice. Při jedné otáčce válce dojde k vykreslení celého obrazu právě jednou. Různé úhly zrcátek vychylují paprsek do řádků. Počet řádků je tedy přímo dán počtem zrcátek. Nastavení úhlu je provedeno pomocí šroubů a během funkce zařízení se nemění. Při rotaci válce paprsek pomocí každého zrcátka přejde přes jeden řádek, modulací jasu během toho pohybu dojde k rozlišení jednotlivých bodů. Čas pro vykreslení jednoho bodu je možné měnit softwarově, rozlišení v horizontálním směru je tedy možné měnit.

Pro stálost obrazu je nutné jej obnovovat s frekvencí alespoň 25 Hz, protože dojde k vykreslení matice bodů za jednu otáčku válce, musí tento válec rotovat alespoň 1500 otáček za minutu.

Na následujících obrázcích je podrobně popsán princip vychylování.

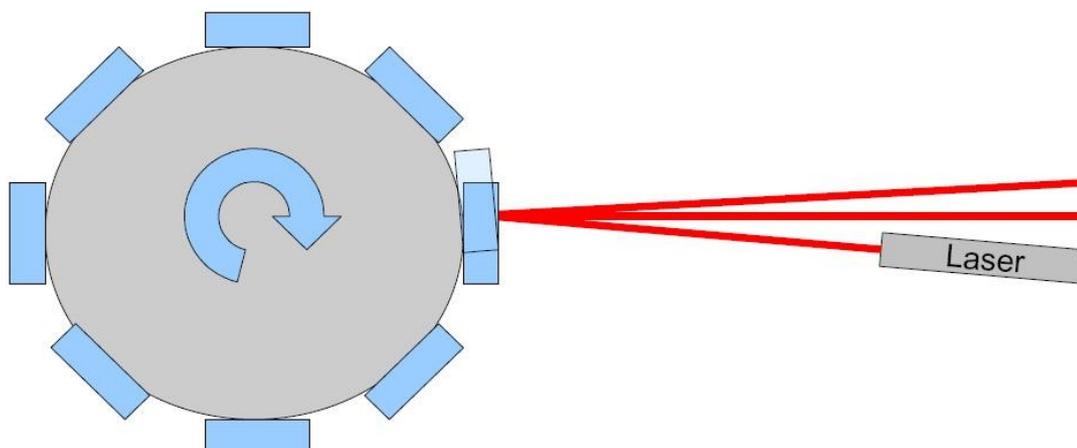
Pohled z boku na zrcátko při vykreslování



obr. 2: vertikální vychylování paprsku

Každé zrcátko je vychýleno o jiný úhel a tím se laserový paprsek rozmítá do řádků promítané matice.

Pohled shora na rotující mechaniku se zrcátky.



obr. 3: horizontální vychylování paprsku

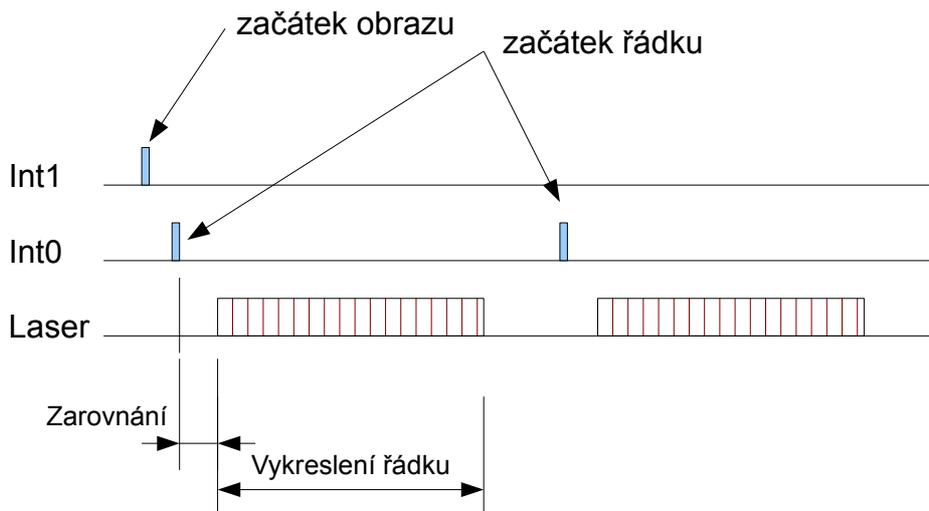
Při rotaci kotouče je jas laseru modulován procesorem, tím vzniká rozklad paprsku na jednotlivé body řádku. Počet bodů v řádku závisí na rychlosti spínání laseru a je omezen frekvencí spínače a možnostmi kalibrace zařízení.

2.2 Diagram časování

Pro správnou funkci zařízení je nutné aby byl mikroprocesor synchronizován s rotující částí. Při předpokladu přesných a stabilních otáček kotouče se zrcátka teoreticky stačí jedna událost za jednu otáčku kotouče. Druhá možnost je zaznamenávat událost kdy má dojít k začátku nového řádku promítané matice a událost pro první řádek matice. To vede na n událostí za jednu otočku v závislosti na počtu promítaných řádků, plus jedna událost indikující první řádek.

Tento postup byl zvolen při konstrukci zařízení, projektor je opatřen osmi zrcátky, synchronizace se tedy provádí osmi značkami pro začátek řádku a jednou pro indikaci prvního řádku.

Na následujícím obrázku je zobrazena časová souslednost událostí při projekci matice bodů.



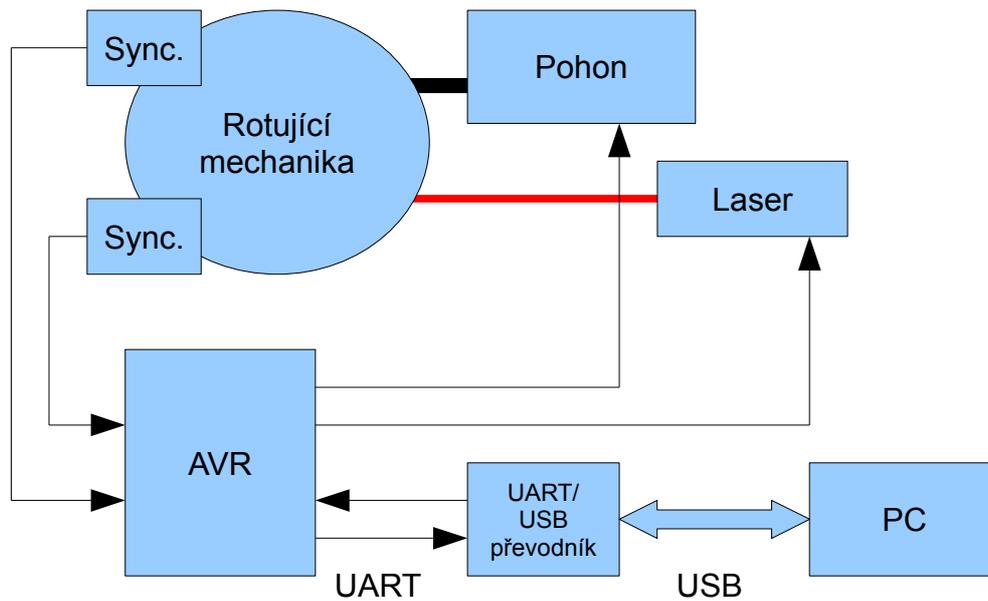
obr. 4: Diagram časování

Po každé značce začátku řádku následuje pauza pro zarovnání všech řádků pod sebe. Při dokonale provedené mechanice by tato pauza byla zbytečná, v praxi se bez ní zařízení neobejde.

Při změně otáček dochází k protažení nebo zkrácení obrazu v horizontálním směru, ne však k jeho rozpadu.

Vlastní mechanismus snímání příslušných okamžiků je popsán v kapitole realizace.

2.3 Blokové schéma



obr. 5: Blokové schéma

Základem projektoru je rotující mechanika se zrcátky, její otáčky a poloha je snímána synchronizačními elementy v několika bodech. Řídící jednotkou zařízení je procesor řady AVR. Podle synchronizačních signálů řídí pohon mechaniky a moduluje laserový paprsek. Zařízení komunikuje s PC pomocí sériové linky a převodníku na USB.

2.4 Spínání laseru

Nároky na spínací element projektoru závisí na požadovaných výkonech. Při použití malého výkonu laseru, pokud je zdrojem laserového záření modul s laserovou diodou, stačí ke spínání šestice invertorů technologie TTL. Proud takovou diodou je řádově 30mA. Při požadavku vyššího výkonu je nutné použít speciální budič. Proud a také světelný tok laserové diody je velmi závislý na teplotě, proto se do jednoho pouzdra s laserovou diodou umísťuje PIN dioda jako detektor světelného toku. Tato PIN dioda slouží poté jako zpětná vazba pro regulaci proudu laserovou diodou. Při konstrukci budiče je nutné tuto zpětnou vazbu zapojit, nebo použít speciální budič určený pro laserové diody.

Nároky z hlediska spínací frekvence závisí na počtu zobrazených bodů, u malého obrazu s 100 body na řádek je frekvence řádově 100kHz. Tento fakt klade na spínací element omezení, většina tranzistorů však vyhovuje.

U skutečného provedení předváděcího modelu byla použita šestice invertorů technologie ATC.

2.5 Pohon

Pro pohon rotující části zařízení je nejvýhodnějším řešením synchronní motor. U takového pohonu není nutné řešit stabilitu otáček a jejich regulaci. U synchronních motorů je však problém s jejich rozběhem a dostupností malých motorů pracujících na požadovaných otáčkách.

Asynchronní motor by vyhovoval bezproblémovostí rozběhu i stabilitou otáček. Motor vhodných rozměrů se mi však nepodařilo nalézt.

Použití stejnosměrného motorku s komutátorem není sice nejvhodnější z hlediska stability otáček, ale výhodou jeho použití je jeho snadná regulace. K spínání stačí jeden unipolární tranzistor řízený pomocí PWM přímo generovanou mikroprocesorem.

U skutečného provedení projektoru popsaného níže byl použit stejnosměrný motor právě pro svoji snadnou dostupnost a jednoduché ovládání.

U stejnosměrného motoru je vhodné použít převod, umožňující provoz motoru blízko jeho jmenovitých otáček, které obvykle bývají vyšší než požadované otáčky mechaniky se zrcátky. Přímé mechanické spojení by bylo vhodné při použití synchronního nebo asynchronního motoru. U přenosu hnacího momentu z motoru na mechaniku projektoru bylo experimentováno s převodem pomocí ozubených kol. Toto řešení se ukázalo jako funkční, ale při daných otáčkách bylo nepříjemně hlučné a vytvářelo nežádoucí vibrace. U konečného provedení projektoru byl použit převod pomocí řemenice a gumičky. Toto řešení nedovoluje sice okamžitý rozběh, je však minimálně hlučné a tlumí přenos vibrací z motoru na mechaniku projektoru.

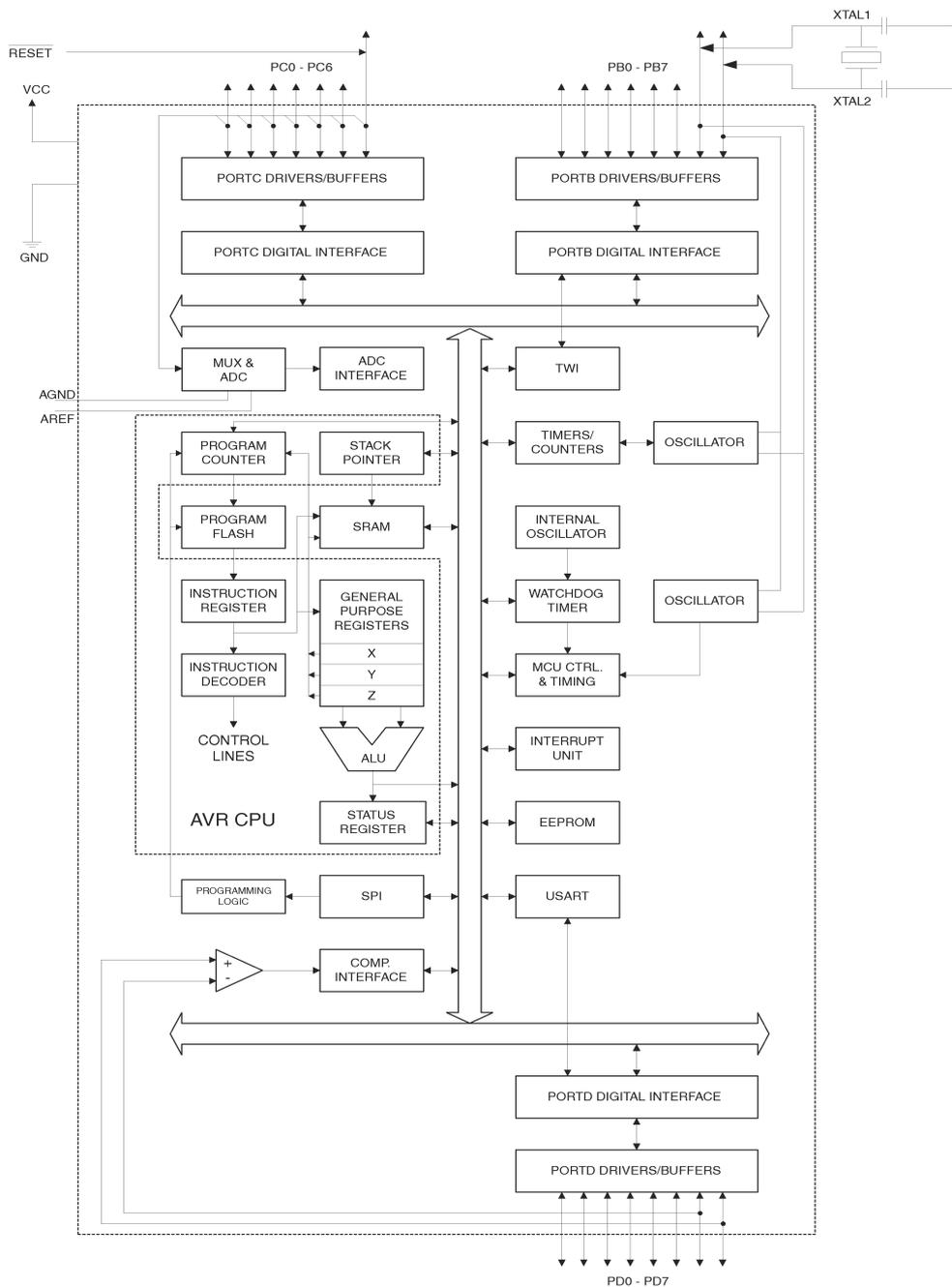
2.6 Synchronizace

Při použití synchronního motoru jako pohonu by bylo možné použít jeho řídicí signály k synchronizaci, při použití stejnosměrného motoru je nutné procesor s mechanikou synchronizovat jinak. Uvažovány byly magnetické snímače pro svoji spolehlivost, takové řešení by sestávalo z malého magnetu na rotující mechanice a na halově sondě sloužící k detekci průchodu. U skutečného provedení byla zvolena varianta optických bran a kotouče s otvory. Zvolena byla pro jednoduchost a snadné nastavení, blíže je popsána v kapitole realizace HW.

2.7 Mikrokontrolér AVR

Procesory AVR jsou moderní osmibitové mikrokontroléry od firmy Atmel. Jedná se o procesor s RISC architekturou. Jejich velká výhoda je rychlost, většina instrukcí je vykonána za jeden nebo dva takty oscilátoru. Pro realizaci modelu projektoru byl zvolen typ ATMEGA8

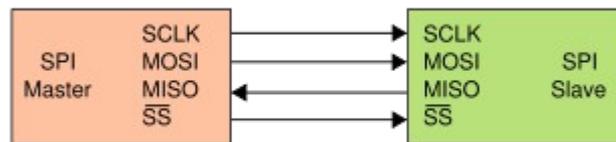
Na následujícím obrázku je blokové schéma procesoru ATMEGA8



obr. 6: Architektura AVR

Aplikace využívá na čipu integrovanou sériovou linku, 3 nezávislé časovače a 2 vstupy pro vnější přerušení. Jako zdroj hodinového signálu je použit vnitřní oscilátor. K uchování zobrazovaných dat slouží vnitřní paměť EEPROM.

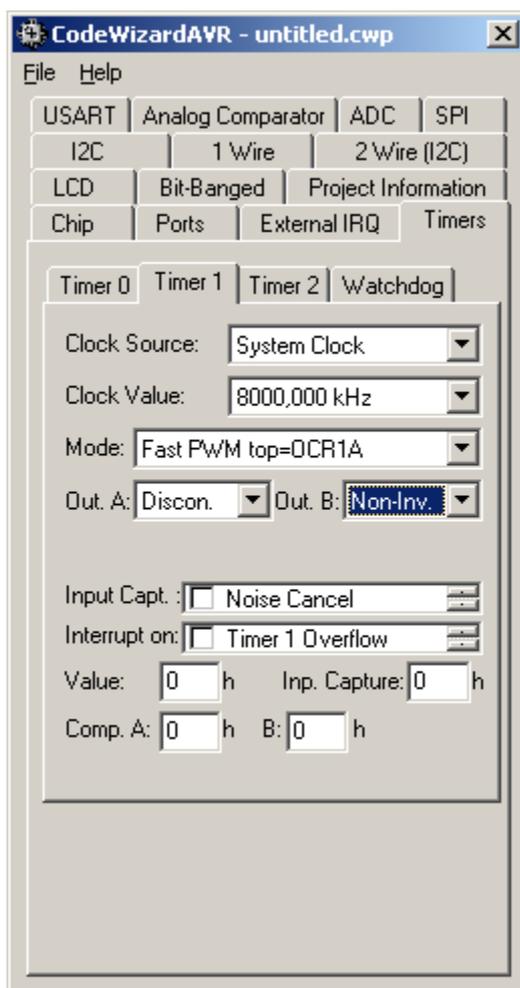
K nahrávání programu do procesoru slouží SPI rozhraní. Toto rozhraní umožňuje nahrávat program do procesoru bez nutnosti jeho vyjmutí z aplikace.



obr. 7: Komunikace standartu SPI

2.8 Vývojové prostředí CodeVision

Pro vývoj software mikrokontroléru bylo použito vývojové prostředí CodeVision AVR [6]. Program mikrokontroléru je v tomto prostředí možno psát v jazyce C a vkládat části kódu v assembleru. Vývojové prostředí dále nabízí možnost automatického generování kódu jako je kostra programu a inicializace komponent mikroprocesoru.



obr. 8 : Generátor kostry programu a inicializace

Tyto nástroje výrazně usnadňují tvorbu programu. Další součástí tohoto prostředí je přímá podpora mnoha typů SPI programátorů a sériový terminál. Kompilátor programu CodeVision převádí program v jazyce C do assembleru a poté do hex souboru. Program umožňuje prohlížet kód v assembleru.

2.9 Komunikace

Pro komunikaci s nadřazeným počítačem slouží sériová linka mikrokontroléru. Protože dnešní PC již nebývají zpravidla osazovány klasickým sériovým portem s rozhraním RS232. Byl použit převodník firmy FTDI. Tento převodník s minimem dalších součástí zajišťuje most mezi sériovou linkou mikroprocesoru a USB portem PC prostřednictvím virtuálního COM portu. Tento sériový port je poté v PC dostupný pro programy jako běžný COM port. Sériová linka převodníku je napěťovými úrovněmi kompatibilní se sériovou linkou mikroprocesoru a není nutné mezi ně vkládat další součástky.

3 Realizace HW

3.1 Mechanické provedení

Kostrou zařízení je deska ke které jsou připevněny ostatní elektronické a mechanické komponenty. Elektromotor je umístěn pod deskou, mikroprocesor a ostatní obvody jsou umístěny nad deskou. Deska mimo nosné funkce funguje také jako stínění mezi motorem a logickými obvody. Deska je vyztužena tak, aby tlumila vibrace vznikající rotací válce se zrcátky.

Rotující část zařízení sestává z válce ke kterému jsou připevněny lamely nesoucí jednotlivá zrcátka. Každou lamelu je možné pomocí šroubu ohnout a tím nastavit příslušné zrcátko do požadovaného úhlu. Dále je k válci připevněn kotouč ve kterém jsou pravidelně rozmístěné otvory pro synchronizaci optickými bránami. Výkres této části mechaniky je přiložen v příloze.

Tento válec je poháněn stejnosměrným elektromotorem, hnací moment je přiveden přes reduktor o převodovém poměru přibližně 1:4. Tento převod umožňuje aby se motor otáčel rychlostí při které se snadněji reguluje a dovoluje použití motoru s malým točivým momentem.

3.2 Elektronika

Elektronika projektoru je provedena na jedné desce plošného spoje. Mimo tuto desku jsou umístěny optické brány a převodník sériové linky na USB port. Ke tvorbě DPS byl použit SW Eagle [7].

Zdroj a spínání motoru

Zdroj sestává z dvoucestného usměrňovače, filtračního kondenzátoru a monolitického stabilizátoru 7805. Ke spínání motoru slouží unipolární tranzistor IRF 540. Spínací tranzistor musí splňovat podmínku dostatečného otevření při napětí na vstupu gate 5V. Proto byl použit tento tranzistor, přestože jeho katalogový proud je mnohem vyšší než je požadavek v této aplikaci. Paralelně s motorem je dále zapojena dioda v závěrném směru pro potlačení napěťových špic vznikajících při odpojení motoru od napájecího

proudu. Tyto napěťové špice jsou obecně problémem vznikajícím u spínání indukčních zátěží.

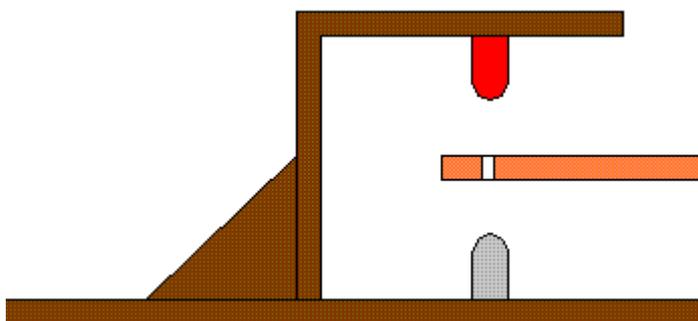
Řídící elektronika

Řídící elektronika sestává z vlastního mikroprocesoru, obvodu 7414 obsahující shmitovy invertory, budiče laseru a konektorů pro připojení ostatní elektroniky. Obvod 7414 slouží pro tvarování signálu z optických bran a odstraňuje případné zákmity. Protože vnější přerušení mikroprocesoru je citlivé na hranu, je tvarování těchto signálů důležité pro správnou funkci.

Jako budič laseru byla použita šestice inverterů technologie ATC v obvodu 74ATC04. Tyto invertory se vyznačují větší zatížitelností výstupu. Laser je na výstup inverterů připojen přes odpor o hodnotě 10 ohmů.

Optické brány

Pro snímání otáček rotující mechaniky a pro synchronizaci procesoru slouží dvě stejné optické brány. Každá optická brána sestává z LED diody a fototranzistoru umístěných proti sobě. Mezi nimi je jako clona umístěn kotouč s vyvrtanými otvory.



obr. 9: konstrukce optické brány

USB převodník

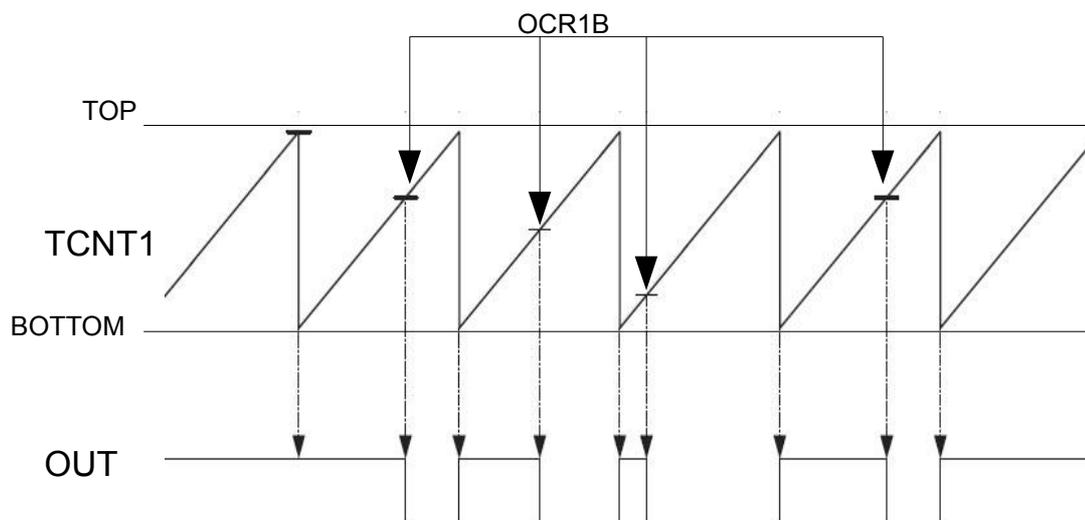
USB převodník je umístěn na samostatné DPS. Jedná se o katalogové zapojení obvodu FT232RL. Převodník je proveden jako samostatný modul a jeho použití je univerzální. Na jedné straně je opatřen přímo konektorem USB který dovoluje jeho přímé připojení do počítače bez nutnosti kabelu.

4 Implementace SW mikrokontroléru

Tato kapitola popisuje nároky kladené na procesor a postup jak je procesor řeší. Při implementaci programu pro mikrokontrolér byl použit manuál k vývojovému prostředí [1] a k mikrokontroléru ATMEGA8 [3].

4.1 PWM

PWM je generována pro řízení otáček motoru. Pro generování PWM je použit časovač 1 v módu FastPWM. V tomto módu je PWM generována zcela automaticky bez nutnosti obsluhovat časovač programem. Pro nastavení parametrů generované PWM slouží několik registrů, registr OCR1A nastavuje maximální hodnotu časovače a registr OCR1B hodnotu při které se aktivuje výstup. Výstup se nuluje při dosažení maximální hodnoty (OCR1A) kde se také nuluje vlastní časovač. Tyto registry jsou šestnáctibitové.



obr. 10: Generování PWM

4.2 Regulace otáček

K regulaci otáček slouží v procesoru implementovaný PI regulátor. Pro volbu PI regulátoru rozhodla skutečnost že dosahuje nulové ustálené odchylky a není pro implementaci příliš složitý. Bylo by možné implementovat složitější

regulátor, ovšem za cenu vyššího zatížení procesoru a výsledný regulační děj by nebyl o mnoho lepší.

K měření rychlosti slouží osmibitový časovač softwarově rozšířený na 16 bitů. Regulátor je dále omezen pouze na kladnou akční veličinu, a to z důvodu nemožnosti realizace záporné akční veličiny při spínání motoru pouze v jednom kvadrantu.

4.3 Modulace laseru

K modulaci laserového paprsku slouží rutina obsluhy přerušení časovače 2. Po časové značce začátku řádku se do registru čítače nahraje kalibrační hodnota pro příslušný řádek a časovač se spustí. Při prvním zavolání obslužné rutiny přerušení od časovače 2 je paprsek na místě, kde je možné začít s vykreslením prvního bodu řádku. Funkce rozhodne zdali bod vysvítit nebo ne, nastaví časovač pro délku bodu a skončí.

Při dalším zavolání této obslužné rutiny dojde k vykreslení druhého bodu, tak se děje periodicky pro všechny body daného řádku. Po vykreslení všech řádků se čítač zastaví a čeká až na spuštění od události synchronizace pro další řádek.

4.4 Struktura dat

Data k vykreslení jsou uložena jako pole `byte[234]` umístěné v paměti eeprom. Každý byte reprezentuje jeden sloupec výsledné matice. Byty 0-77 představují první řádek textu, byty 78-155 druhý řádek textu, byty 156-234 třetí řádek textu. Řádky je možné zobrazovat jednotlivě nebo postupně jako běžící text. Sloupce jsou číslovány zleva doprava. Každý bit jednotlivého byte potom představuje jeden pixel, bit s nejnižší vahou reprezentuje pixel umístěný nahoře.

Kalibrační data jsou umístěna také v paměti eeprom a to jako pole `byte[8]` kde každý byte představuje čas nutný k vyrovnání matice. Číslo takto uložené vynásobené čtyřmi představuje počet us.

4.5 Komunikace s PC

Při komunikaci s PC procesor přijímá data z PC a zapisuje je do paměti dat eeprom, veškerá komunikace se děje v obslužné rutině sériové linky.

Většina příkazů sloužících ke komunikaci mezi projektorem a PC jsou jednotlivé znaky. Tyto příkazy jsou vypsány v následující tabulce.

příkaz	význam
'n'	stop – projektor se zastaví, sériová linka funguje
'l'	rozsvítí laser – vhodné při kalibraci zrcátek
'm'	start – projektor se spustí
horizontální kalibrace	
'w'	přesun o řádek nahoru
's'	přesun o řádek dolů
'a'	posun řádku doleva
'd'	posun řádku doprava
módy zobrazení	
'0'	zobrazení prvního řádku
'1'	běžící text, všechny řádky
'2'	zobrazení druhého řádku
'3'	zobrazení třetího řádku

Pro odesílání dat k vykreslení se používá speciální formát přenosu. Nejprve se odešle znak, který informuje procesor o začátku datového paketu a místa v paměti pro jeho umístění. Poté se odešle znak jehož dekadická hodnota udává počet bytů k přenesení. Poté se již odesílají jednotlivé byty. Po přijetí příslušného počtu bytů přejde procesor zpátky do režimu příjmu příkazů.

5 Obslužný SW pro PC

5.1 Úvod

Obslužný software pro PC slouží ke komunikaci zařízení s uživatelem. Zařízení samo není vybaveno žádnými ovládacími prvky. Veškeré ovládání tedy zařizuje software na PC. Pro vývoj aplikace bylo použito prostředí Microsoft Visual studio [5] verze 2005. Aplikace je napsána v jazyce C#, hlavní výhodou tohoto programovacího jazyka je snadná a rychlá tvorba aplikace.

5.2 Požadavky

Požadavky na obslužný software jsou z principu dva. Za prvé je to možnost kalibrovat nepřesnosti mechaniky a tím srovnat výsledný obraz. Za druhé potom možnost nahrát data ke zobrazení. Je vhodné aby data ke zobrazení bylo možné zadat jak graficky, tak textem.

5.3 Řešení

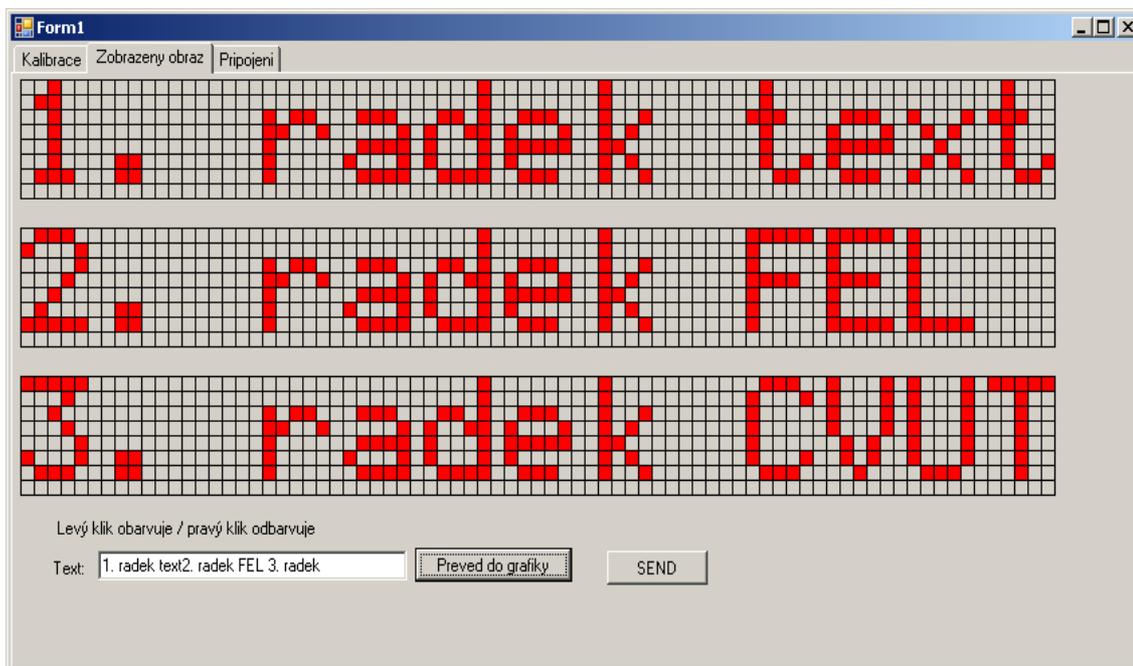
Program sestává z jednoho okna, ve kterém jsou formou záložek umístěny jednotlivé ovládací prvky.

V panelu připojení je možné zvolit jméno portu pro komunikaci a jeho rychlost. Při kliknutí na tlačítko OPEN se program pokusí otevřít daný port a pokud uspěje, zobrazí se tato informace v příslušném labelu. Pokud se port nepodaří otevřít, zobrazí se opačná informace. Tlačítko CLOSE funguje obdobně, slouží k uzavření portu a tím k jeho uvolnění. Kliknutím na tlačítko test program vyšle po sériové lince znak a čeká na odpověď. Pokud v příslušném čase dorazí odpovídající odpověď, je uživatel informován o připojení zařízení.

Na panelu kalibrace se nacházejí 4 tlačítka, která slouží k nastavení jednotlivých kalibračních konstant. Při kliknutí na některé tlačítko program vyšle znak po sériové lince.

Panel Obraz umožňuje zadat obraz určený ke zobrazení projektorem. Obraz je možné zadat buď klikáním bod po bodu nebo je možné zadat text do textového pole. Při zadání textu program převede text na matici bodů kterou je možné dále graficky editovat. Převod probíhá na základě fontu pro

maticové LCD displeje. Jeden znak je reprezentován maticí 5x8 bodů uloženou jako 5 bytů velké pole. Mezi znaky se vkládá jeden bod mezera.



obr. 11: náhled přípravy dat k vykreslení

5.4 Odeslání dat

Odesílání dat do mikrokontroléru probíhá přes vybraný port. Nejprve se pošle znak uvozující přenos dat, poté počet odesílaných byte a nakonec vlastní data. Mezi jednotlivými datovými byty se čeká na příjem potvrzovacího znaku. Pokud potvrzovací znak nedorazí včas, dochází k timeoutu a sekvence se ukončí. Vlastní odesílání běží jako vlastní vlákno a jeho čekání na případný timeout nezpůsobí zaseknutí aplikace.

6 Závěr

Výsledkem mé práce je funkční model laserového projektoru schopný promítat matici 8x78 bodů. Model je realizován včetně technické dokumentace. Projektor komunikuje s nadřazeným počítačem pomocí sériové linky. Základem projektoru je mikroprocesor ATMEGA 8 s programem o délce přibližně 2k slov.

Dále je realizován program v jazyce C# pro PC. Tento program slouží k ovládání projektoru a demonstraci jeho funkce. Umožňuje zadávat obrázky i text ke zobrazení.

Tento systém projekce může být dále rozvíjen a při konstrukci mechaniky s větším počtem zrcátek je schopen snadno promítat matici o větším počtu řádků. Počet řádků je však omezen možností umístění konečného počtu zrcátek na kotouč. Při použití procesoru s větší pamětí dat eeprom nebo s externí pamětí je možné promítat mnohem více obrazů, případně zvýšit horizontální rozlišení. Zvyšování horizontálního rozlišení je limitováno spínací frekvencí laseru a možným vyzařovacím úhlem projektoru. Zvětšování horizontálního rozlišení klade také větší nároky na kalibraci.

7 Použité zdroje a software

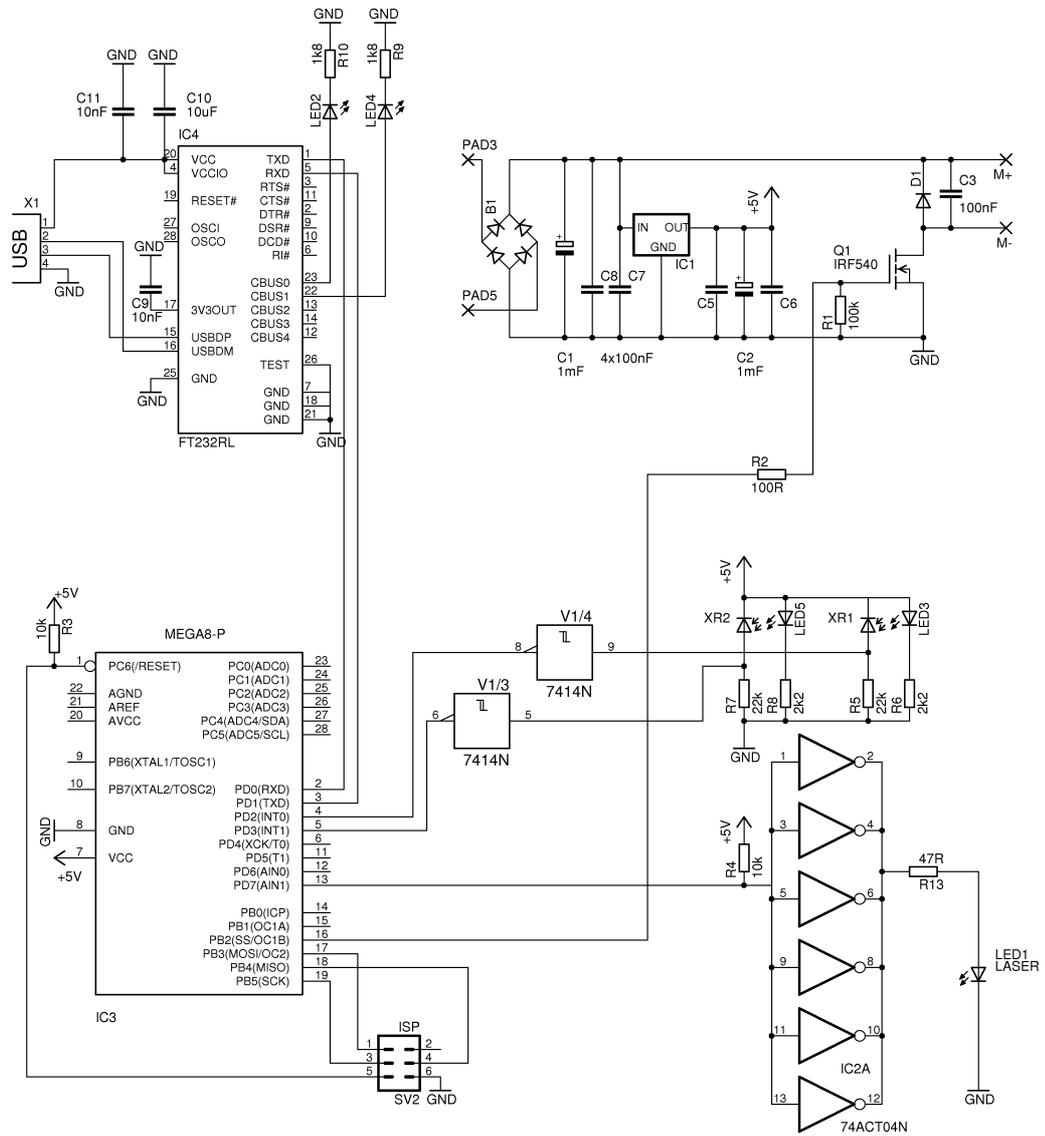
- [1] Manuál k vývojovému prostředí AVR - Code Vision AVR manual (eng.)
<http://www.hpinfotech.ro/cvavrman.zip>
- [2] Přednášky na PJR - Ing. Richard Šusta Ph.D. – C# tutorial
<http://dce.felk.cvut.cz/pjr/>
- [3] Atmega8 manuál – Atmel (eng.)
http://www.atmel.com/dyn/resources/prod_documents/doc2486.pdf
- [4] Font pro tvorbu znaků v PC programu
<http://www.sxlist.com/TECHREF/datafile/charset/8x6.htm>
- [5] Prostředí pro tvorbu aplikace pro PC Microsoft Visual studio 2005
<http://www.microsoft.com/cze/msdn/produkty/vstudio/default.mspx>
- [6] Prostředí pro tvorbu SW mikrokontroléru Code Vision AVR
<http://www.hpinfotech.com/>
- [7] Program pro kreslení schémat a DPS Eagle Layout Editor 4.16r2
<http://www.cadsoft.de>

8 Přílohy

8.1 Rozpiska součástek

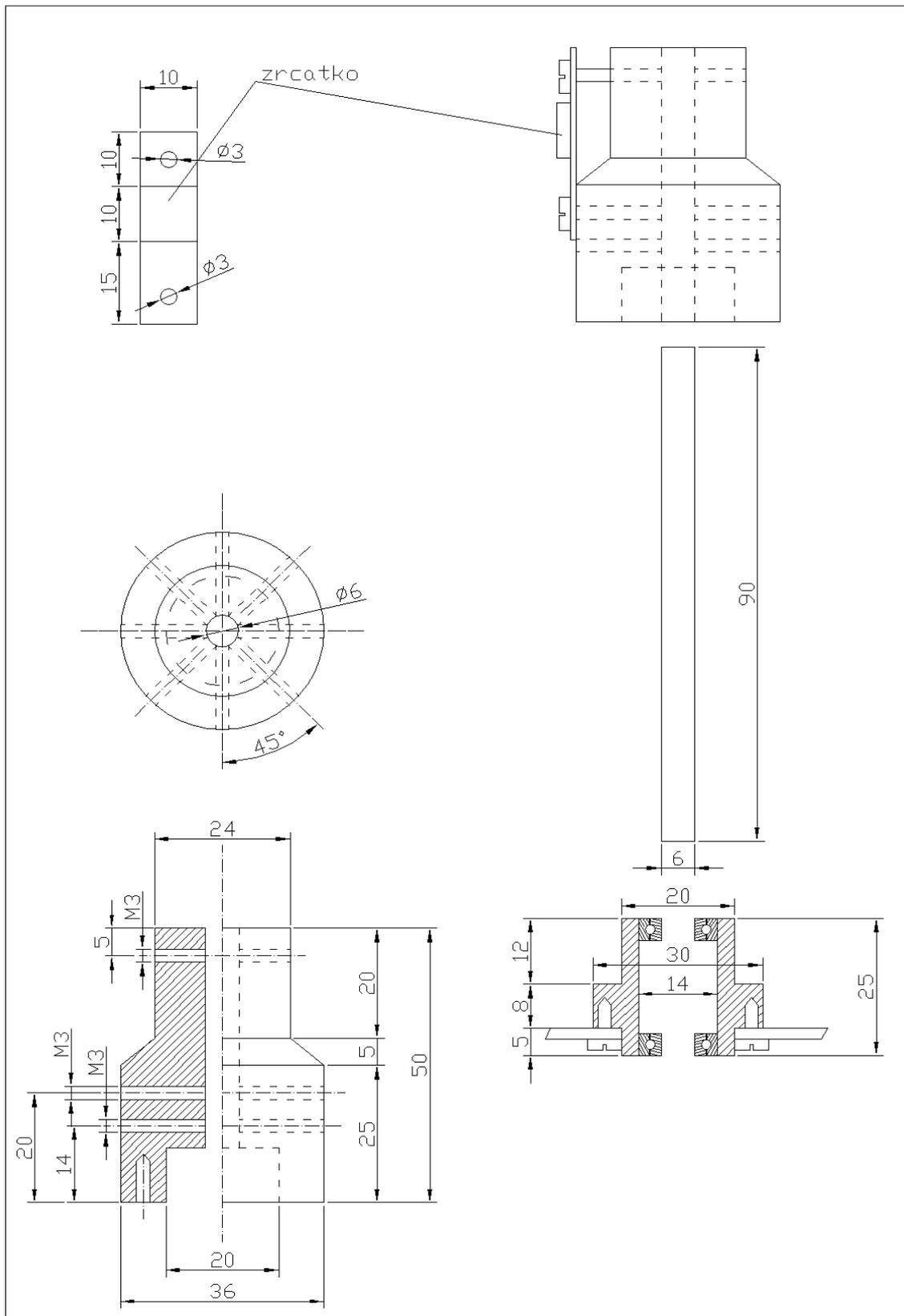
ID	Hodnota	Typ	pouzdro
B1		usměrňovací můstek	
C1	220uF	elyt.	E5-8
C2	220uF	elyt.	E5-8
C3	100nF	keramika	C050-025X075
C5	100nF	keramika	C050-025X075
C6	100nF	keramika	C050-025X075
C7	100nF	keramika	C050-025X075
C8	100nF	keramika	C050-025X075
C9	10nF	keramika	C0805
C10	10uF	keramika	C1206
C11	10nF	keramika	C0805
D1		1N4007	DO41
IC1		7805L	TO220
IC2		74ACT04N	DIL14
IC3		MEGA8-P	DIL28-3
IC4		FT232RL	SSOP28
LED1	LASER	5mW	modul
LED2		LED SMD	1206
LED3		LED 3MM	LED3MM
LED4		LED SMD	1206
LED5		LED 3MM	LED3MM
Q1		IRF540	TO220
R1	100k		0207
R2	100R		0207
R3	10k		0207
R4	10k		0207
R5	22k		0207
R6	2k2		0207
R7	22k		0207
R8	2k2		0207
R9	1k8		0207
R10	1k8		0207
R12	100R		0207
R13	10R		0207
V1		7414N	DIL14
XR1		fototranzistor	3mm
XR2		fototranzistor	3mm

8.2 Schéma zapojení



obr. 12: Schéma zapojení

8.4 Výkres mechaniky



obr. 16: Výkres mechaniky

8.5 Výpis programu mikrokontroléru

```
#include <mega8.h>
#include <stdio.h>

//deklarace globalnich promenych
bit pomaluflag;
bit firstlineflag;
bit datatransmission;
bit datacountflag;
char datacount;
//char start;

char radek=0;
char sloupec=0;
char aktualnimaska=0;
char aktualnidata=0;
char pointer=0;
char datapointer=0;
//char datapage=0;
char count;
eeprom char mode=0;
char start;
unsigned int rychlost;
int akcni=0;
signed int odchylka=0;
unsigned int I=0;
static unsigned int zadana=40000;
eeprom char kalibrace[8]={0x39,0xff,0x34,0x44,0x34,0x41,0x21,0x11};
eeprom char datapole[234];
flash char maska[8]={ 0b00000001,
                      0b00000010,
                      0b00000100,
                      0b00001000,
                      0b00010000,
                      0b00100000,
                      0b01000000,
                      0b10000000} ;

// hlavni program
void main
(void) //-----
{
    // inicializace procesoru
    TIMSK = 0b01000001;//[OC2;TO2;TI1;OCA1;OCB1;TO1;-;TO0 ; povoli
preruseni
    //nastaveni casovace 1 jako FAST PWM
    TCCR1A=0xA3;
    TCCR1B=0x19;
    OCR1A=4095; //maximum pwm

    // NASTAVENI CASOVACE_0
    TCCR0 =0b00000001;// ;

    //;nastaveni seriove linky
    UBRRL = 25 ;// ;nastaveni rychlosti na 19200 (pri 8 MHz clock)
    //UBRRL = 51 ;// ;nastaveni rychlosti na 9600 (pri 8 MHz clock)
```

```

//UBRR1 = 207 ;// ;nastaveni rychlosti na 2400 (pri 8 MHz clock)
UCSRB = 0b10011000 ; // ;povoleni preruseni

//;nastaveni externiho preruseni
MCUCR = 0b00001010;//10; na hranu
GICR = 0b11000000;//192; //povol preruseni

DDRB=0x06; // smer portu B
DDRC=0xFF; // smer portu C
PORTD=0b00001100; // pullup
DDRD= 0b11010010; // smer portu D

#asm("sei") // Global enable interrupts
while (1)
{
    //loop
}
} // ----- konec funkce
main();-----

void regulace (void)
{
    odchylka = (signed int) rychlost - zadana;
    if (rychlost > zadana) {I++;}
    else {I--;}
    if (I==0)I=1;
    if (I==0x0fff)I=0x0FFE;
    akcni = odchylka>>1;
    akcni += I;
    if (akcni < 0) akcni = 0;
    if (akcni > 0x0fff) akcni = 0x0fff;
    OCR1B = akcni;//)&0x0FFF);
}

//----- mereni rychlosti -----
interrupt [TIM0_OVF] void timer2_overflow_isr(void)
{
    count++; //sw rozsireni casovace na 16 bit
    if (count == 0) { // pretečení, motor se asi netoci
        rychlost = 65500;
        pomaluflag=1;
        regulace();
    }
}

interrupt [EXT_INT0] void ext0(void)
{
    GICR = 0b10000000;
    //pokud se motor drive netocil
    if (pomaluflag==1){pomaluflag=0;}
    else{ //odecet rychlosti
        rychlost = count;
        rychlost = rychlost <<8;
        rychlost = rychlost + TCNT0;
    }
    TCNT0=1;//nulovani casovace
    count=0;
    regulace();
    radek++;//prechod na dalsi radek
    if (radek ==9){radek=0;}//radku je 8
}

```

```

if (firstlineflag==1)
  { //nastaveni prvnio radku
    radek=0;
    firstlineflag=0;
    if (mode==1) start+=1;
    if (mode==0) start=0;
    if (mode==2) start=78;
    if (mode==3) start=156;
  }
sloupec=start; // zacatek radku

//horizontalni kalibrace

TCNT2=250-kalibrace[radek];
//maska pro vyber bitu z dat k vykresleni
aktualnimaska=maska[radek];
SFIOR=2;
TIMSK = 0b01000001;
TCCR2=0b00000011;
}
//-----

interrupt [EXT_INT1] void ext1(void)
{ // vnejsi preruseni pro 1. radek
  firstlineflag=1;
}

interrupt [TIM2_OVF] void timer2_over(void)
{

  SFIOR=2; //nuluji prescaler

  TCNT2=251; //delka pixelu (255 je nejmensi delka, 250 uz dost dlouha)
  TCCR2=0b00000011;
  aktualnidata = datapole[sloupec]; //vyber aktualniho sloupce
  #asm("SBI PORTD,7") //zhasnu laser
  //a pokud je v pameti pixel tak rozsvitim
  if (sloupec<234){
if ((aktualnidata&aktualnimaska)!=0){ #asm("CBI PORTD,7")} }
  // po vykresleni celeho radku konec
  if (sloupec ==start+77) {
    TCCR2=0b00000000; //vypnu casovac
    #asm("SBI PORTD,7") //zhasnu laser
    GICR = 0b11000000;
  }
  sloupec++; //prechod na dalsi pixel
}

interrupt [USART_RXC] void prijem(void)
{ //komunikace s PC
  if (datatransmission) { //pokud ocekavam data
    //tak je zapisu
    datapole[datapointer]=UDR;
    //a presunu kurzor
    datapointer++;
    //konec datove relace
    if (datapointer==datacount) datatransmission=0;
  }
  else { //nebo cekam na pocet dat
    if (datacountflag) {

```

```

        //prijmu pocet dat
        datacount=UDR;
        datacountflag=0;
        //a cekam na data
        datatransmission=1;
    }
else    { //jinak cekam prikaz
    if (UDR==116)    { //zacatek datapaketu na 1.obraz
        datacountflag=1;
        datapointer=0;
    }
    if (UDR == 109) { //start
        GICR = 0b11000000;
        TIMSK = 0b01000001;
    }
    if (UDR == 110) { //stop
        GICR = 0b00000000;
        TIMSK = 0b00000000;
        OCR1B =0;
    }

    //kalibrace
    if (UDR == 115)pointer++;
    if (UDR == 119)pointer--;
    if (pointer==8)pointer=7;
    if (pointer==255) pointer=0;
    if (UDR==100)kalibrace[pointer]++ ;
    if (UDR==97)kalibrace[pointer]--;
    if (UDR==108)#asm("CBI PORTD,7")
    if (UDR==48)mode=0;
    if (UDR==49)mode=1;
    if (UDR==50)mode=2;
    if (UDR==51)mode=3;
    }
    }
    putchar(65); //po prijeti kazdeho znaku poslu potvrzeni
}

```

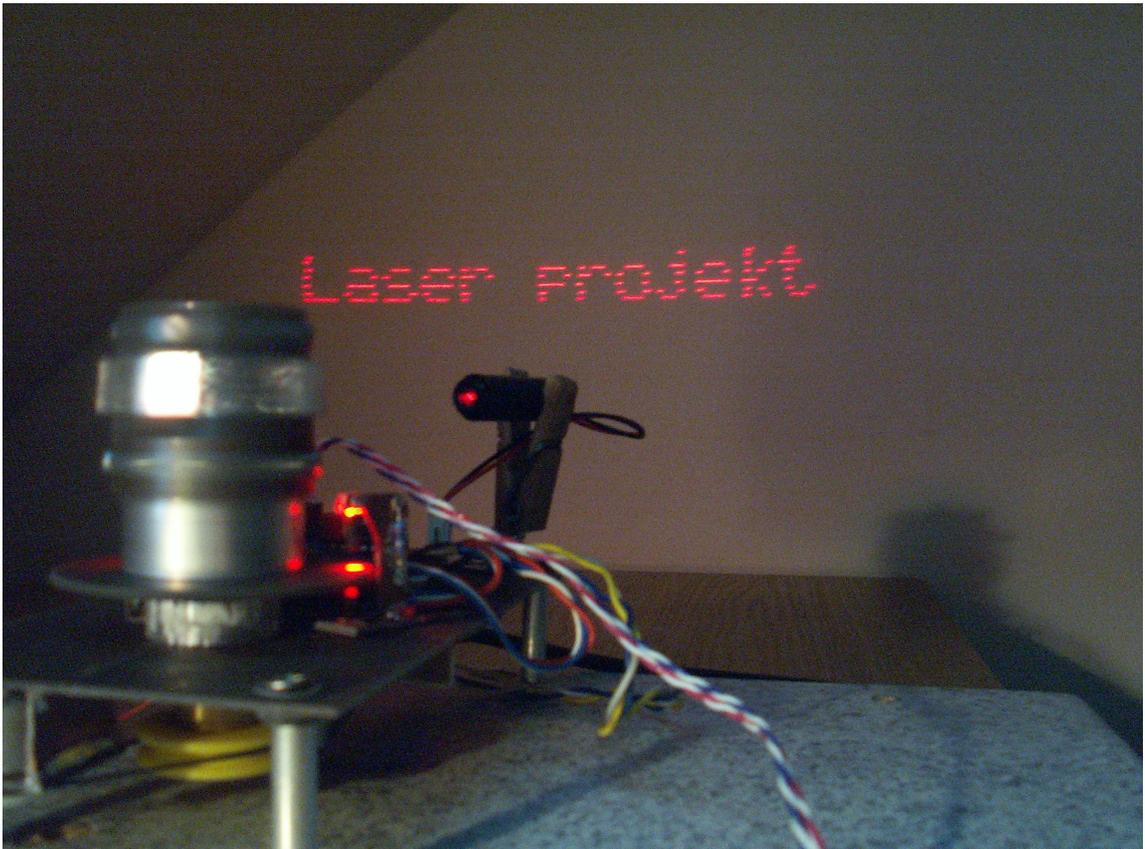
8.6 Obrazová dokumentace



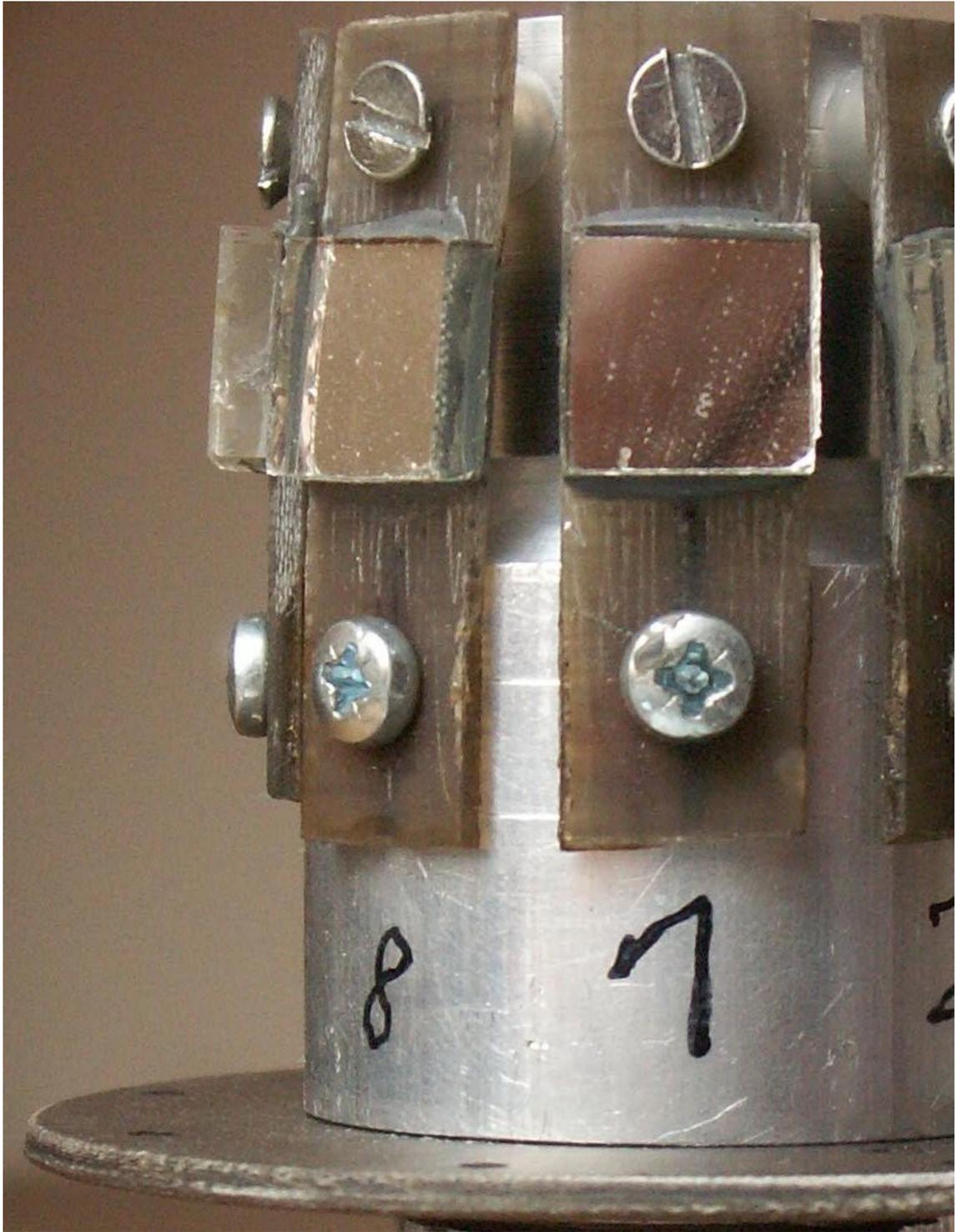
obr. 17: Hello world



obr. 18: Funkce



obr. 19: Fungující prototyp



obr. 20: Detail mechaniky