

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

KATEDRA TECHNOLOGIÍ A MĚŘENÍ

BAKALÁŘSKÁ PRÁCE

Lineární posuv fotoaparátu

Originál (kopie) zadání BP/DP

Abstrakt

Práce obsahuje popis konstrukce, postupy úprav a sestavení lineárního posunu. Porovnání s nejčastějšími konstrukcemi. Návrh a realizaci řídicí elektroniky, vhodný výběr pohonu k požadované činnosti. Napsání programu pro řízení motoru a spínání závěrky fotoaparátu.

Klíčová slova

Lineární posuv / posun, konstrukce, kolejnice, vozík, krokový motor, plošný spoj, ATmega 32, H-můstek L298, HD44780, řídicí program posuvu, fotoaparát, časosběr

Abstract

This thesis contains construction details, modification process and construction procedure for linear slider. Comparison of common constructions. Draft and realization of control electronics, suitable choice of propulsion for the required activity. The programming part for motor control and camera shutter.

Key words

Linear slider, rails, construction, cart, stepper motor, printed circuit, ATmega 32, H-bridge L298, HD44780, motor driver, camera, timelapse

Prohlášení

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

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

.....

podpis

V Plzni dne 4.6.2015

Vít Krejčí

Poděkování

Tímto bych rád poděkoval vedoucímu diplomové práce doc. Ing. Pavlu Drábkovi, Ph.D. za rady, konzultaci, připomínky a zařízení potřebných věcí k dokončení práce. Panu Ing. Jiřímu Čengerymu, Ph.D. za zhotovení desky plošného spoje. Kamarádovi Bc. Janu Pokornému za cenné praktické informace a diskuzí nad řešeným problémem návrhu schématu a programové části. Otcí Robertu Krejčímu za asistenci při výrobě konstrukce pojezdových kolejí.

Obsah

OBSAH	7
SEZNAM SYMBOLŮ A ZKRATEK	9
ÚVOD	10
BODY PRÁCE	10
1 PŘEDSTAVENÍ LINEÁRNÍCH POSUVŮ VE FILMOVÉ TECHNICE	11
2 KONSTRUKCE A REALIZACE LINEÁRNÍHO POSUVU	13
2.1 MOŽNOSTI ŘEŠENÍ VODÍCÍHO VEDENÍ	13
2.1.1 <i>Vedení pomocí štěrbin</i>	13
2.1.2 <i>Tyčové vedení</i>	14
2.2 REALIZACE A POPIS VLASTNÍ KONSTRUKCE	16
2.2.1 <i>Zhodnocení dostupných a hotových prvků</i>	16
2.2.2 <i>Popis výroby – koleje, vozík</i>	16
2.2.3 <i>Upevnění motoru, chod řemene</i>	18
2.2.4 <i>Volnoběžná řemenice</i>	18
2.2.5 <i>Instalace koncových spínačů</i>	19
2.2.6 <i>Řídící vodiče k fotoaparátu</i>	19
2.2.7 <i>Podpěrné příčnický</i>	19
2.2.8 <i>Krabička</i>	20
3 POHON PRO POSUV VOZÍKU	21
3.1 POŽADAVKY	21
3.1.1 <i>Výpočet zatížení</i>	21
3.2 STEJNOSMĚRNÝ MOTOR	22
3.2.1 <i>Teorie</i>	22
3.2.2 <i>Servomotor</i>	22
3.3 KROKOVÝ MOTOR	23
3.3.1 <i>Teorie</i>	23
3.3.2 <i>Hybridní krokový motor</i>	24
3.3.3 <i>Elektrické možnosti zapojení a řízení hybridních krokových motorů</i>	25
4 NÁVRH ŘÍDÍCÍ ELEKTRONIKY A REALIZACE	28
4.1 POPIS FUNKCE ELEKTRONIKY	28
4.1.1 <i>Napájení</i>	28
4.1.2 <i>Stabilizátor</i>	28
4.1.3 <i>Výkonové spínání motoru</i>	30
4.1.4 <i>Spínání spouště fotoaparátu</i>	31
4.1.5 <i>Zobrazovací jednotka</i>	32
4.1.6 <i>Řídící mikroprocesor a podřadné periférie</i>	33
4.2 NÁVRH PLOŠNÉHO SPOJE	34
4.2.1 <i>Rozmístění součástek na DPS</i>	35
4.2.2 <i>Výroba</i>	35
5 PROGRAMOVÁNÍ A OŽIVENÍ OBVODU	36
5.1 PROGRAMOVÁNÍ	36
5.1.1 <i>Nastavení mikroprocesoru</i>	36
5.1.2 <i>Roztočení motoru</i>	38
5.1.3 <i>Zobrazovací jednotka</i>	41
5.1.4 <i>Režimy pojezdu</i>	42

ZÁVĚR	44
SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ	46
PŘÍLOHY	1

Seznam symbolů a zkratek

m	Hmotnost [kg]
g	Gravitační zrychlení [m/s ²]
F	Síla [N]
r	Poloměr řemenice motoru [cm]
ω	Úhlová rychlost [rad/s]
M	Moment motoru [Nm]
P	Výkon [W]
α	Úhel kroku [°]
m	Počet fází
N_r	Počet zubů
U_{CE}	Úbytek napětí na tranzistoru, kolektor; emitor [V]
P_Z	Ztrátový výkon [W]
U_L	Napětí na induktoru [V]
L	Indukčnost [H]
di_L	Derivace proudu induktorem
dt	Derivace času
t_{rr}	Doba odezvy sepnutí diody po přivedení napětí na vývody [ns]

Úvod

Cílem práce je navrhnout a sestavit konstrukci lineárního posuvu. Vytvořit pevné pojezdové koleje s fixovaným vozítkem. Porovnat nejčastěji užívané typy posunů a zhodnotit výhody. Vybrat vhodnou pohonnou jednotku, pro požadovaný provoz snímání časosběru. Představit i jiné alternativy motorů. Zhotovit řídicí elektroniku posunu. Zvolit řídicí mikroprocesor s dostatečným počtem pinů a programovatelnými funkcemi bran. Obvod spínající výkonové požadavky pohonu. Popsat jejich úlohu, posoudit správnost zapojení a funkčnost v praxi. Napsat program pro obsluhu periférií elektroniky, řídicí synchronizovaně chod motoru se spouštěním fotoaparátu. Zobrazit stav krokování motoru na displeji s možností měnit jeho pracovní hodnoty pohybu.

Body práce

První kapitola je věnována představení posuvů ve snímání (filmové) technice s orientací na poloprofesionální a amatérské použití. Druhá kapitola popisuje detailní sestavení konstrukce lineárního posuvu, složené z běžně dostupných hliníkových profilů, doplněná vizuální foto dokumentací jednotlivých prvků. Výběrem pohonné části posuvu se zabývá třetí kapitola. Zde jsou uvedeny detaily o krokových, stejnosměrných i servo motorech. Větší zmínka patří krokovým motorům a jejich možnostem zapojení vinutí statoru. Čtvrtá část řeší návrh schématu, rozmístění součástek na plošném spoji a vysvětlení jednotlivých celků zapojení. Pátá pojednává o programové výbavě, nadefinování funkcí procesoru, řízení spínání cívek krokového motoru, zobrazení znaků na displeji, synchronizace pohybové platformy vozíku a fotoaparátu. V závěru jsou zhodnoceny dosažené výsledky, práce s posuvem v terénu, shrnuty možnosti zlepšení.

1 Představení lineárních posuvů ve filmové technice

Obecně se lineární posuvy ve filmovém průmyslu využívají už dlouhou řadu let. Koleje podobné těm železničním, jen menších rozměrů, obvykle se pokládají na zem. Na nich je umístěn pojezd, vozík s hydraulickou stoličkou (regulace výšky posedu) a uchycenou kamerou. Posuv je zpravidla tažen asistentem. Tento systém se využívá stále, ale s modernější a lehčí technikou nevyžadující robustní konstrukci pojezdu, se mohl zmenšit do skladných a přenosných rozměrů, ovládaný pouze jednou osobou. Kolejnice lze umístit libovolně dle situace: na zem, mezi dvě překážky nebo využít stativ. Vozík se stativovou hlavou nesoucí kameru umožňuje dynamicky sledovat scénu, nebo měnit úhel pohledu na objekt. Evoluce snímací elektroniky je obrovská, z toho důvodu jsou též lineární posuvy žádané, především pro méně rozpočtové projekty či reklamní spoty.



Obr. 1 – Profesionální vybavení filmařů (převzato z [1])

Kamerové lineární posuvy se vyrábí různých délek a konstrukčních provedení. V tomto článku budeme hovořit o poloprofesionálním a amatérském využití, nebudeme se bavit o profesionální technice. Konstrukční možnosti často souvisí s provozovaným zařízením (kamera, fotoaparát) z tohoto hlediska nás zajímá hmotnost techniky a poté samotné využití neboli práce s posunem, zda budeme chtít zachytit dynamické scény nebo nám postačí méně náročný pomalý posun bez prudkých akcelerací. V amatérských respektive v poloprofesionálních využitích se setkáváme s délkami kolejí obvykle do dvou metrů. Jelikož pak nastává problém s jejich manipulovatelností, tuhostí konstrukce a obsluhou.

U video scény se vždy snažíme o co nejplynulejší přejezd kamery, žádné zadrhávání není žádoucí. Zvláště pokud je nasazen objektiv s dlouhým ohniskem, kde rozeznáme sebemenší pohyb, projevující se rozhoupáním nebo vlněním obrazu. Pohyb nemusí být vždy konstantní, ale obvykle je kamera ručně posouvána a rychlost se přizpůsobuje pohybu snímaného objektu. Konstrukce musí být optimalizována na těžší techniku i proměnlivé zatížení, podle práce kameramana, jak se snímací technikou nakládá (kontrola děje, ostrost objektů).

Foto scéna se v tomto liší. Pohyb už není v tomto hledisku spojitý, ale naopak diskrétní. Fotoaparát se vždy pohne o určitou vzdálenost, zastaví, zachytí snímek a opět se posune. Proto bývají obvykle tyto posuny vybaveny pohonnou jednotkou, která zabezpečuje stále stejný krok posunu. Jednotlivé snímky se poté dají v počítači ještě dostatečně upravit. Snímáme-li do formátu RAW, možnost úprav je opravdu veliká. Nemluvě o značně větším rozlišení fotek, které bývají oříznuty do formátu 16:9 (formát obrazu pro video), ale i tak mají téměř o $\frac{1}{2}$ větší rozlišení než standart Full HD. Z pořízených snímků lze následně složit sekvenci a vytvořit plynulou projekci (díky konstantnímu posunu). Tato metoda je zaměřená především na statické či pomalu se pohybující objekty. Proto se tato metoda označuje jako časosběr. Rychlé objekty jsou poté vyobrazeny rozmazaně a pomalé objekty najednou můžeme vnímat velice dynamicky. Takovýto efekt podtrhuje samotný pohyb snímacího zařízení, tudíž i statické věci se jistým způsobem mění, jelikož se fotoaparát posouval po kolejích. To diváka samozřejmě velice zaujme.

Pohyb posunu se využívá jak v horizontální rovině, kdy váha je rozložena přes vozík na koleje, tak i téměř ve vertikálním směru, kdy veškerou váhu vozíku a fototechniky drží pohonná jednotka. I s tímto využitím je nutné v návrhu počítat a zamyslet se nad průhybem konstrukce bude-li zapřena pouze na koncových bodech kolejí, nebo nad použitím dostatečně výkonného motoru s dostatečným záběrným momentem.

2 Konstrukce a realizace lineárního posuvu

Předně požadujeme sestavit stabilní, pevnou a zároveň lehkou konstrukci pro pojezd kamerového systému. Je nežádoucí, aby se konstrukce měla pohybem rozvíbrovat, případně rezonovat. Nemělo by docházet k pohupování při jemném zavádění o fotoaparát nebo opětovném rozjetí a zastavení vozíku, eliminovat co nejvíce setrvačnost dokmitávání (rozmazání obrazu při dlouhých expozicích, světelné čáry). Konstrukci je vhodné vyrobit z lehkého materiálu, kvůli časté manipulaci, týkající se donesení na místo obvykle vzdálené nebo nepřístupné. Proto je nutné vybírat z různých slitin materiálů a vhodně volit i konstrukční profil. Snažit se docílit uvedených parametrů, ale zároveň respektovat hmotnost.

2.1 Možnosti řešení vodícího vedení

Je celá spousta možností řešení vodícího vedení, vyberu hlavní zástupce skupin a zmíním ty nejvíce rozšířené.

2.1.1 Vedení pomocí štěrbin

Koleje jsou zde řešeny různými profily obvykle tvaru U, H apod.. Ve štěrbině jsou umístěny kolečka vozíku. Tento systém je velice jednoduchý, nicméně vozík není žádným způsobem fixován na kolejnice. Vedený pohyb není tak přesný. Avšak tento pohyb je ve výsledku dobře plynulý, tudíž malé odchylky tolik nevadí. Vozík se dá obvykle použít i samostatně na hladké podlaze, potřebujeme-li natáčet z vyšší úrovně, použijeme vodící koleje, které umístíme na stativ. Tento systém je využíván jen pro pořizování video záznamu zpravidla amatérského.



Obr. 2 – Vozík se silikonovými kolečky určen na hladký povrch, nebo využijeme kolejnic (převzato z [2, 3])

2.1.2 Tyčové vedení

Toto vedení je nejrozšířenější, v obdobných variantách se s ním můžeme setkat i v profesionálním vybavení filmařů. Kolejnice je prezentována tyčovým¹, trubkovým² tvarem. Různé modifikace se poté liší systémem použití posuvných prvků a upnutí vozíku na kolejnice.

- **Třecí posuv**

Vozík může být upnut jen částečně, použitím vhodných protikusů, které doléhají podélně na kolejnice a obepínají je alespoň do jedné poloviny, nebo pevně, vozík je na vedení nasunut. Kluzný posuv s rostoucí hmotností lze hůře ovládat, nemá dojezdovou setrvačnost, může se zadržávat (u filcových vložek). Plastové vložky jsou nejlepší variantou pro kluzný posuv. Vyžadují jen minimální údržbu, v důsledku eliminaci ložisek je systém imunní vůči vodě.



Obr. 3 – Vozík s třecími vložkami (převzato z [4, 5])

- **Ložisková kolečka**

Vhodnou soustavou ložiskových koleček vozíku obepneme vodící trubičku. Kolejnice můžeme obepnout dvěma identickými vozíky přišroubovanými k sobě a mezi jejich bočními hranami bude probíhat vedení sevřené kolečky. V druhém způsobu jsou kolečka různě úhlově posunuta (místo trubičky se používá i čtyřhranný profil, kolečka jsou posunuta o 45°) a mezi kolejemi mohou být i rozepřena. Třetí možností je využití váhy vozíku. Na přední a zadní straně je čtveřice koleček (2 kolečka pod kolejnicí, 2 kolečka nad) po zatížení se pákovým způsobem rozepřou o koleje.

¹ Tyč – plný profil

² Trubka – dutý profil

³ Prokluz – motor se točí, ale řemen stojí a klouže v řemenici, obvykle klouzne jen o několik zubů a opět se chytne. Bez zpětné vazby určujeme polohu vozíku s chybou.

⁴ První číslo pólové dvojice značí směr tekoucího proudu

² Trubka – dutý profil



Obr. 4 – Sestava různých typů vozíků a jejich vedení (převzato z [6 - 9])

- **Lineární ložiska**

Nejpřesnější metodou je přichycení vozíku pomocí lineárních ložisek využívající se obvykle v průmyslu například u CNC fréz, 3D tiskáren a ostatních přesných posuvů využívající systém souřadnic X, Y. Na kolejnici je nasunuto ložisko zajišťující přesný pohyb, s minimální vůlí, rozkládá váhu v celé délce ložiskového pouzdra, po montáži musíme pouzdro dodatečně vystředit posunutím vnitřní vložky (nutné u podepřených tyčí). Ložisko může být celé uzavřené ať už oválné nebo čtvercové s montážní plochou, nebo ložisko s otevřeným domkem (průřezem) pro podepřené tyče (koleje). Nasunutí ložiska též zajistíme fixaci vozíku a umožníme pohyb jen v jedné ose. Tímto způsobem docílíme jak posuvu v horizontálním směru, tak i ve směru vertikálním. Vozík se z kolejí nevychýlí. Vedení je náročnější na údržbu, obvykle je natřeno tenkou vrstvou vazelíny, na kterou lehce přilnou nečistoty, nejvíce vadí zrnka písku vnikající mezi dráhu ložisek způsobující zadrhávání pojezdu.



Obr. 5 – Vlastní konstrukce vozíku, ukázka ložiska (převzato z [10])

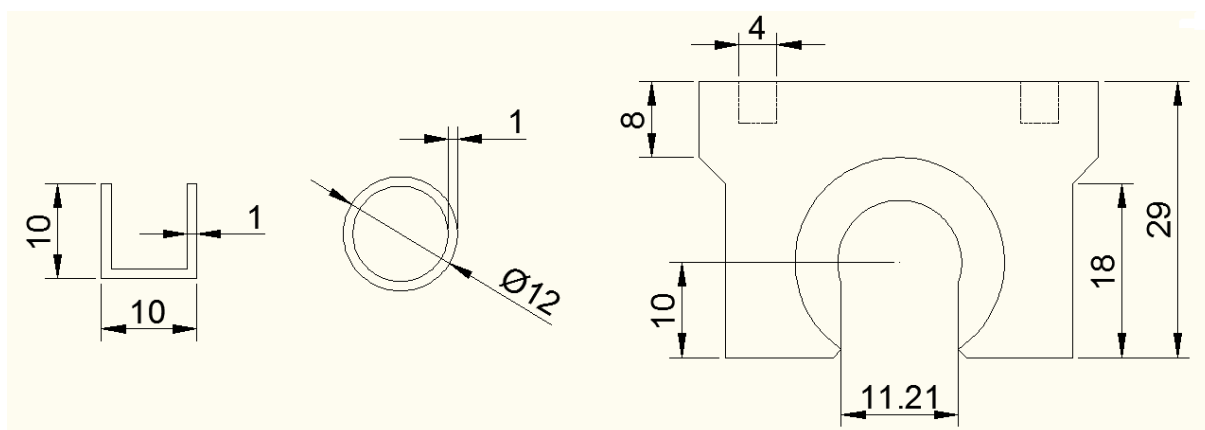
2.2 Realizace a popis vlastní konstrukce

2.2.1 Zhodnocení dostupných a hotových prvků

V prvotních přípravách byly zhodnoceny veškeré průmyslové pojezdové profily, sériově vyráběny v dostupných cenách. Pojezdové vybavení se týkalo převážně dveřních systémů. Nicméně hledané řešení v sortimentu nebylo. Z tohoto důvodu bylo rozhodnuto o vlastní konstrukci.

2.2.2 Popis výroby – koleje, vozík

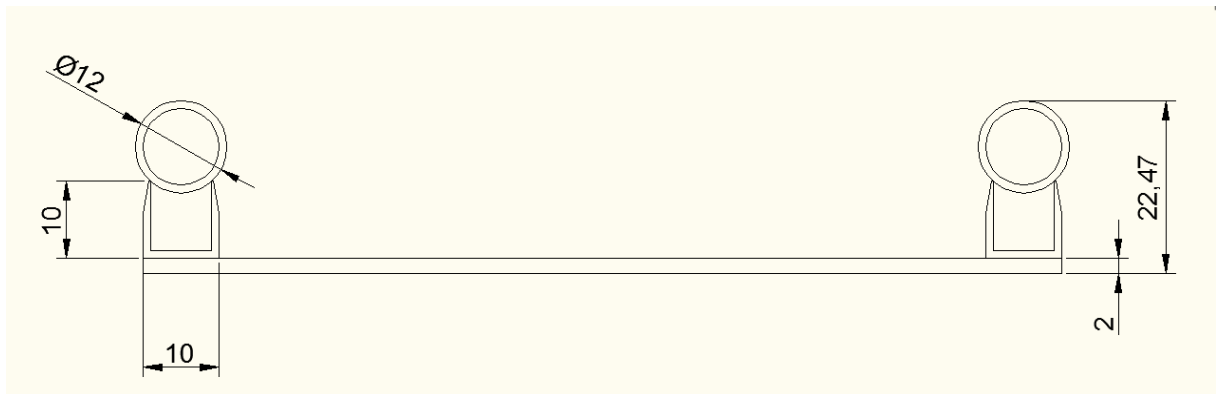
K sestavě vlastních kolejí, byly preferované dostupné profily v běžných obchodech. Nalezeny byly hliníkové profily tvaru U a trubičky. Tvrdost a pevnost hliníku není tak vyhovující, bohužel jiný materiál v cenově dostupné relaci není. V obchodě pro RepRap tiskárny (3d tisk) byly nalezeny vyhovující ložiska s otevřenou spodní částí, tak aby pod nimi mohl být umístěn U profil. Rozměry pouzdra ložiska odpovídají průměru trubičky. Po nalezení vhodných dílů započala sestava konstrukce.



Obr. 6 – Vybrané profily: U, trubička, ložisko.

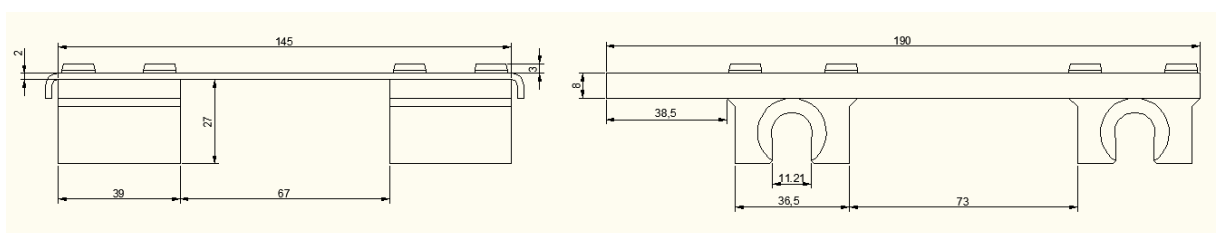
Nejprve bylo nutné u tvaru U zbrousit horní vnější hrany, aby nedocházelo k otěru ložiska ve spodní otevřené části a ložisko mohlo lehce dosednout. Po kontrole dostatečné

průchodnosti byla na U profil přišroubována trubička vruty. Po sešroubování vznikl pevný profil využíván jako kolejnice.



Obr. 7 – Sestava kolejnic

Po smontování i druhé kolejnice byly obě části spojeny částmi hliníkového plechu. Během spojování kolejnic bylo velmi důležité dodržet konstantní rozestup. Na konci jedné strany kolejí byl namontován plech s čelem, zakončující dráhu vedení a uzavírající jednu stranu. Poté na koleje byly nasunuty ložiska, opět zhodnocena průchodnost. Následovalo rozmístění ložisek, posazení hliníkového plechu na ložiska a rýsování os montážních otvorů. Provedlo se odvrtání, kontrola os otvorů, dobroušení, přišroubování a nasazení na kolejnice. Řešení problému s třením o hrany U profilu, úprava plastové vložky ložiska, její pootočení a eliminaci tření. Vyzkoušení a zajištění plastové vložky zápustným šroubem (červíkem). Do vozíku byl vyvrtán otvor pro instalaci stativového závitu pro uchycení fotoaparátu. Po téměř bezproblémovém posunu, byla ložiska promazána olejem a kolejnice potřeny jemnou vrstvou vazelíny.



Obr. 8 – Vozík

Přesahující plocha vozíku byla původně zamýšlena pro posuv z vnější strany, pomocí závitové tyče. Na vyčnívající části by musela být uchycena unášející matka. Od tohoto druhu řešení se upustilo, kvůli značnému navýšení hmotnosti a nemožnosti pohybovat vozíkem manuálně.

Jedna strana kolejí zůstala tedy otevřena pro nasunutí vozíku a pozdější návrh upevnění motoru, pohonu vozíku.

2.2.3 Upevnění motoru, chod řemene

Po vhodném výběru pohonné jednotky začal návrh umístění řemene a upevnění motoru. Pohon byl vyňat z 8“ disketové mechaniky, odkud bylo převzato i upevnění motoru. Následovalo přešroubování rámu a umístění jej na plošku plechu mezi kolejemi. Byl přiřazen ke kraji vedení po hloubku vyčnívajícího oblouku, otvoru pro uložení motoru. Zde odvrtny otvory a zespuďu přišroubovány (rám měl na několika místech otvory se závity). Ochranné rezistory vynutí, byly přeuspořáďány a vloženy dovnitř kostry. Přitaženy na její stěny kvůli chlazení. V přední části směrem do dráhy kolejí, byla v místě počátku osy motoru vyříznuta podélná spára pro chod řemene. Otvor pro kabeláž motoru byl převrtán z levé strany na pravou, kvůli vedení řemene v těchto místech. Motor je tedy fixován v rámu kolmo na kolejnice.

Uložení motoru je rozhodující pro uspořáďání řemene. V tomto případě, řemen běhá v jedné linii nad sebou. Tento způsob je preferován, je-li délka řemene již značná a dochází k průhybu. Řemen může být mírně prověšen, ale řemenice ho bez problémů navine. V opačném případě, využíváme-li horizontální oběh, může dojít k vyjetí řemene mimo zuby řemenice, vlivem prověšením jedné strany a ztrátu pnutí, uvolnění vozíku. Řešení by nebylo ideální, nebo by muselo obsahovat pomocné kladky.

2.2.4 Volnoběžná řemenice

Vyměřením osy hřídele motoru, jak hluboko zasahuje do konstrukce a přenesení této vzdálenosti na druhý konec kolejí nám vznikne bod osy řemene. V těchto místech se bude nacházet volnoběžná řemenice. Domeček pro řemeničku je vyroben z hliníkového plechu, naohýbán do tvaru C. V otevřené části jsou nalisovány ložiska s osou. Na ose je nasazena řemenice zafixovaná závitovým červíkem. Domeček je přišroubován do zadního čela kolejí tak, aby se s ním dalo posouvat a mohlo regulovat pnutí řemene. Po nastavení vhodného tahu je domeček zajištěn pomocnými šrouby ze spodu vedení. Správné napnutí řemene je důležité, k zamezení prokluzu³, naopak není-li zařízení dlouho používáno, je vhodné řemen povolit. Řemen byl pořízen pouze v potřebné metrácii a je napojen k vozíku z obou stran.

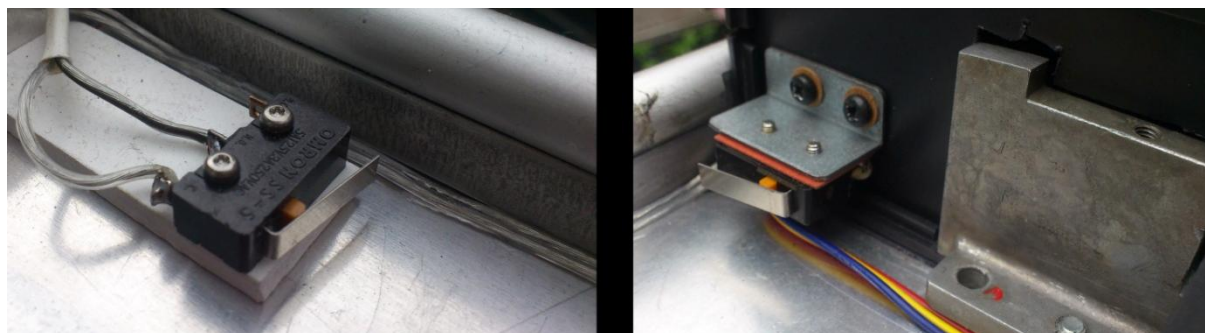
³ Prokluz – motor se točí, ale řemen stojí a klouže v řemenici, obvykle klouzne jen o několik zubů a opět se chytne. Bez zpětné vazby určujeme polohu vozíku s chybou.



Obr. 9 – Volnoběžná řemenice usazená v ložiskách, s možností regulací prnutí řemene

2.2.5 Instalace koncových spínačů

Spínače jsou umístěny na obou koncích kolejí, na jedné straně je spínač podložen izolační destičkou a na druhé je přichycen na čele krabičky. Instalační výška spínacích jazýčků zvolena v úrovni ložisek, dosedající přesně na vnitřní hranu. Po sepnutí, je informace následně vyhodnocena v procesoru a zastaven motor. Drátové vedení je přilepeno na dolní hraně kolejí.



Obr. 10 – Koncové spínače, jazýček dosedá na spodní hranu lineárního ložiska

2.2.6 Řídící vodiče k fotoaparátu

Po jedné straně kolejí je vedena dvojlinka směrem k dorazovému koncovému spínači, po druhé straně je veden svazek tří vodičů určen k ovládání fotoaparátu. Je vyveden zhruba uprostřed pojezdu. Zde jsou vodiče zakončeny konektorem (3,5 mm jack) a dále napojen kabelem určeným k ovládání fotoaparátu. Toto řešení bylo zvoleno kvůli pohybu vozíku a vlečení kratší délky kabelu.

2.2.7 Podpěrné příčníky

V počátku výroby bylo plánováno, že koleje budou moci ležet na zemi v rovině. Proto byly zvoleny i podepřené trubičky. Přesto po namontování držáku motoru v těchto místech

vyčníval 17 mm oblouk. Kvůli náklonu byly vyrobeny nasazovací nohy s možností regulace výšky. Příčnick je nasazen zobákem na spojovací plech. Jednoduché nasazování bylo vybráno, kvůli častému přesouvání na jiné pozice, ale i snadné demontáži pokud upínáme koleje na stativy.



Obr. 11 – Příčnick se stavitelnými nohami

2.2.8 Krabička

Řídící elektroniku je vhodné schovat do krabičky tak, aby byla chráněna proti mechanickému poškození či nepříznivým podmínkám počasí. Víčka krabičky bylo nutné upravit kvůli nosné konstrukci motoru, obkreslit hrany rámu a zaříznout. V horním víčku je zabudován kontrolní displej s ovládací klávesnicí. Celek je zakryt novou maskou kopírující přesněji vyčnívající periferie. Tlačítka bylo nutné upevnit jako plošný blok, aby po zmáčknutí nezapadla. Uříznutím cuprextitu s vnitřními rozměry krabičky, provrtanými krajními rohy, nasadíme na montážní otvory a rozeprème do krabičky. Zkontrolujeme, zda deska dostatečně drží. Přes vytisknutý rastr rozmístíme tlačítka na plošném spoji, označíme a vyvrtáme díry. Lepenkou zalepíme odvrtné plošky a necháme odleptat. Zapájíme tlačítka spolu s přívodními konektory od procesoru. Displej je přichycen vruty na body od tavného lepidla (tafix).



Obr. 12 – Krabička se zabudovanou zobrazovací jednotkou a klávesnicí

3 Pohon pro posuv vozíku.

Odlišujeme posuvy pro tvorbu časosběrů (s motorem) a posuvy pro zachycení scény ve spojitém časovém úseku (v reálném čase, obvykle bez motoru).

3.1 Požadavky

Vyžadujeme rychlou odezvu motoru, velký počáteční moment. Přesný chod motoru, pootočení vždy o stejnou úhlovou vzdálenost, minimální setrvačnost. Velký brzdňý moment, ideálně i po odpojení napájení. Pracovní napětí 12 V stejnosměrných. Vysoké otáčky nejsou žádoucí.

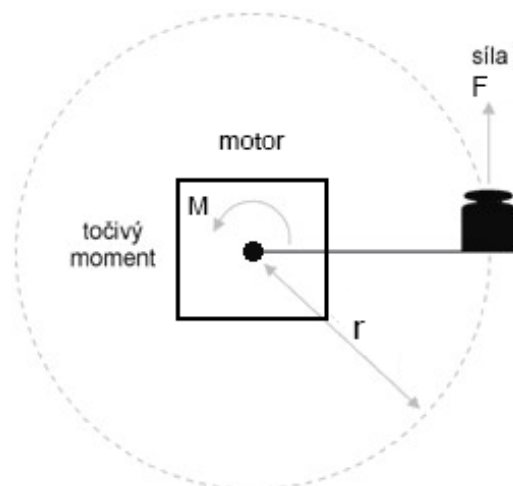
3.1.1 Výpočet zatížení

Celková hmotnost fotovýbavy (tělo, objektiv, stativová hlava) a vozíku je cca 1,5 kg. Budeme počítat s rezervou a zaokrouhlíme hmotnost na 2 kg. Dle vztahu (1) nám vyplývá zátěžný moment 10 Ncm. Toto zatížení je počítáno pro vertikální postavení kolejí („nejhorší stav“), kdy veškerá váha je zavěšena na motoru.

$$M = F \cdot r \quad (1)$$

F – síla působící na motor (hmotnost * gravitační zrychlení; $2 \cdot 9,81$)

r – poloměr páky na hřídel motoru (průměr řemenice 1 cm; poloměr 0,5 cm)

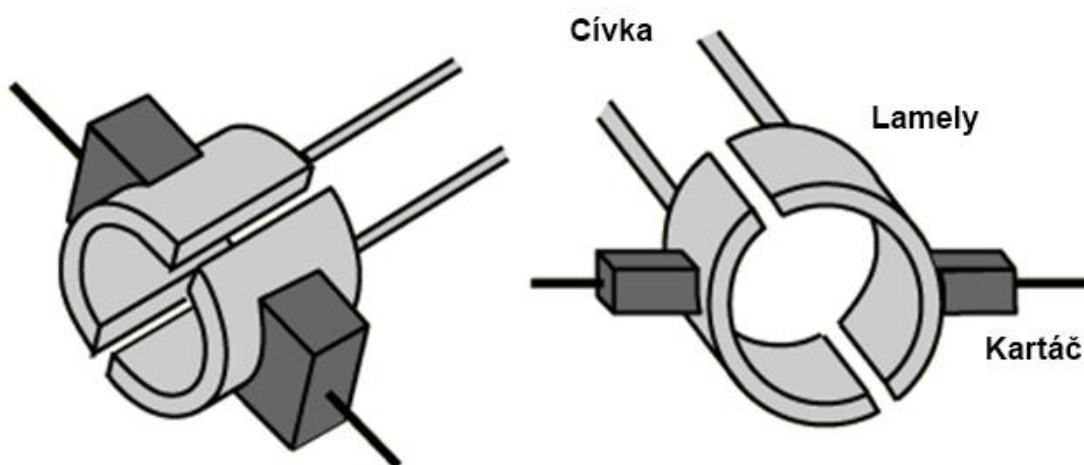


Obr. 13 – Váha břemene působící na hřídel, řemenici (zdroj: [11])

3.2 Stejnosměrný motor

3.2.1 Teorie

Zaměříme se na motor s permanentními magnety, který je preferován především pro svou jednoduchost. Počet permanentních magnetů určuje kolika je to pólový motor. Připojením napájecího napětí na cívky rotoru se cívka natočí k opačnému magnetickému pólu statoru. Tím dojde k pootočení hřídele. Jak se cívka blíží do rovnovážné polohy s magnetickým pólem, dojde k prohození polarity napájené cívky a ta se znovu přitáhne k opačnému pólu statoru, pokračuje ve stejném směru otáčení. Změnami polarity cívky vznikne kruhový točivý pohyb. Prohození polarity v cívkách rotoru umožňuje komutátor. Komutátor je složen z kartáčů připojených na vnější napájení (nepohybují se) a doléhají na lamely, které jsou umístěny kolem hřídele. Z lamel jsou napájeny jednotlivé cívky, kdy točivým pohybem se mění směr protékajícího proudu a vzniká střídavé napětí. Otáčky motoru lze regulovat velikostí přiloženého napájecího napětí, jeho charakteristika je lineární závislostí otáček motoru na buzeném napětí. Využití motorů nalezneme v malé spotřební elektronice, ale i u výkonnějších strojů, často do výkonu 1 kW.[12]



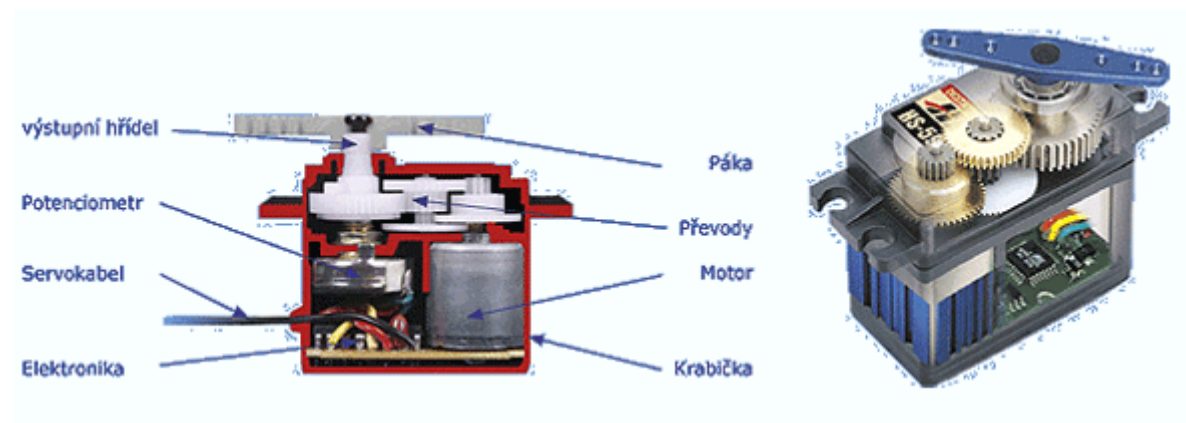
Obr. 14 – Složení komutátoru (převzato z [16])

3.2.2 Servomotor

Jedná se o modelářské servo, osazeno klasickým stejnosměrným motorem s permanentními magnety, umístěného v pouzdře s řídicí elektronikou a několika stupňovou převodovkou. Elektronika zpracovává řídicí signál PWM a podle požadavků natočí hřídel o několik stupňů. Správné natočení je kontrolováno a řízeno zpětnou vazbou. Otáčky jsou snímány potenciometrem s vyvedenou osou na převodové kolečko. S tímto systémem není servo schopno otáčení o celých 360°. Musí následovat úprava, kdy je zbroušen doraz na

převodovém kolečku a odstraněn zpětnovazební potenciometr. Nahradíme ho vhodným odporovým děličem. Takto upravené servo lze řídit v celém kruhovém rozsahu. Ovládáme směr otáčení i rychlost, podle nastavení odchylky od středové hodnoty. Požadujeme-li znát přesné natočení serva, je nutné zařadit novou zpětnou vazbu na hřídeli např.: dírkovaný kotouč s optickým čidlem.[13-15, 17]

Převodovka motoru je dostatečně dimenzována. Po neplánovaném odpojení napájení nehrozí nekontrolované sesunutí vozíku (jsou-li koleje postaveny vertikálně), hmotnost vozíku nepřekoná převodový stupeň převodovky a vozík zůstane stát na místě.



Obr. 15 – Servomotor, řez (převzato z [18])

3.3 Krokový motor

3.3.1 Teorie

Nástup těchto strojů začal teprve v 60. letech minulého století s rozvojem dostatečně rychlých spínacích prvků. Využíval se hlavně v počítačové technice a přídatných periférií, později u obráběcích strojů. Krokový motor je speciální typ stroje s nespojitým otáčením hřídele. Princip je založen na proměnlivém magnetickém odporu mezi vinutím statoru a permanentními magnety rotoru. Vybudíme-li cívku statoru, vytvoříme moment, který natočí rotor do nejbližší rovnovážné polohy, v této poloze je aktuálně nejmenší magnetický odpor. Přepínáním budících cívek se rotor vždy otočí do polohy s minimální reluktancí. Toto natočení nazýváme úhel kroku.[19]

$$\alpha = \frac{360}{m \cdot N_r} [^\circ] \quad (2)$$

Kroky motoru jsou definovány konstrukcí stroje ve výrobě a odpovídají jeho nasazení v praxi. Např.: Pro jemné řízení se využívají motory s úhlem kroku 1,8 stupňů. Chceme-li provést otočku o 360° musíme jednotlivé cívky sepnout 200 krát. Rychlost otáčení motoru je

závislá na rychlosti spínání cívek. Obsluhu spínání vykonává mikrokontrolér spínající obvykle výkonový H-můstek.[20]

Přednosti krokového motoru jsou:

- *Jednoduše říditelný z digitálních systémů*
- *Běžně pracuje bez zpětné vazby*
- *Chyba polohy není kumulativní*
- *Bezkartáčový*
- *Rozběhový moment*
- *Velmi vysoký točivý moment*

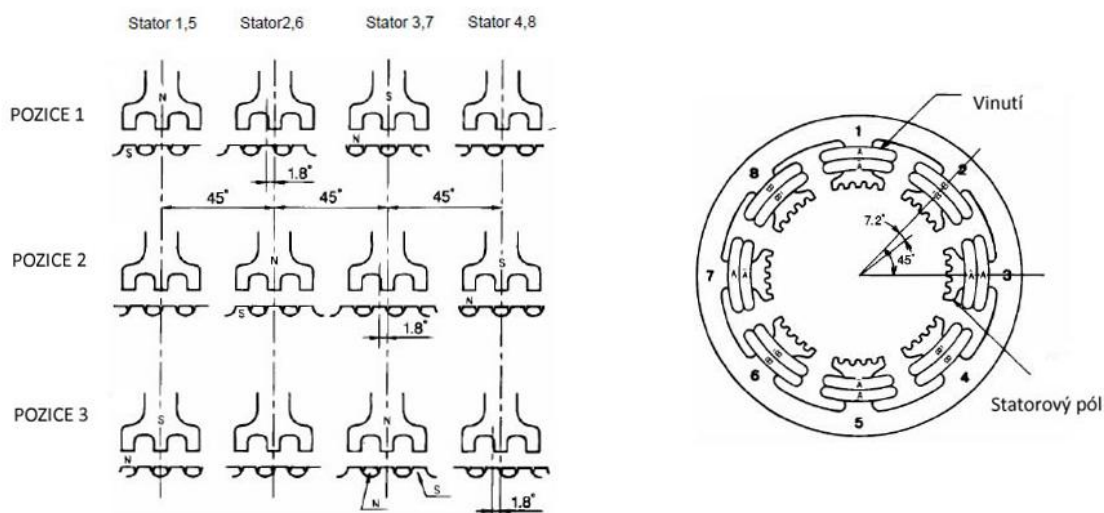
Krokové motory dělíme na:

- *Motory s proměnlivou reluktancí*
- *Motory s permanentními magnety*
- *Hybridní motory*

Zaměříme se na hybridní krokové motory, určené pro malé výkony s malým krokem, orientovanými především pro přesné řízení polohy.

3.3.2 Hybridní krokový motor

Kombinuje výhody obou předchozích motorů, vysoké otáčky, velký moment a dobré tlumení (malý rozestup zubů). Rotor je válcový magnet (neodym) podélně zmagnetizován. Na něj jsou z obou stran nasunuty rotorové pólové nástavce (zubaté příruby), vzájemně proti sobě pootočené tak, aby zuby z jedné strany odpovídaly vybrání na straně druhé. Nástavec „vpředu“ prezentuje severní pól magnetu a nástavec „vzadu“ jižní pól magnetu. Ozubený adaptér vsunutý na neodym je složen z křemíkových plechů. Stator, nejčastěji dvoufázový, tvořen osmi póly. Na každém pólu je umístěn pólový nástavec s vyniklými zuby a dvě vnutí. V závislosti na směru proudu mění pól orientaci pole, S nebo J. [20]



Obr. 16 – Spínání fází a princip pohybu rotoru (převzato z [20])

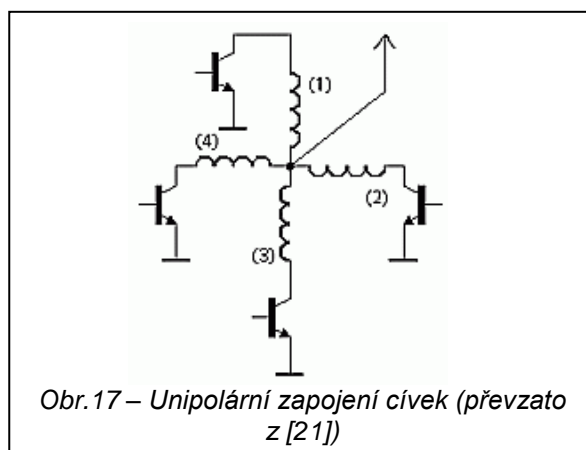
Po připojení napětí na fáze ⁴pólů 1;5 a 3;7 se rotor natočí obrácenou orientací na zuby pólového nástavce. Zmagnetizujeme-li póly 2;6 a 4;8, rotor se posune o $\frac{1}{2}$ zubu. Odpojíme-li napájení pólů 1;5 a 3;7 rotor se opět posune o $\frac{1}{2}$ zubu. Poté opět připojíme napětí, ale v opačné polaritě 5;1 a 7;3. Takto se cyklus opakuje.[20]

Současné hybridní krokové motory jsou vyráběny v hranatých šasi, staré motory mají kulaté provedení.

3.3.3 Elektrické možnosti zapojení a řízení hybridních krokových motorů

3.3.3.1 Zapojení cívek motoru

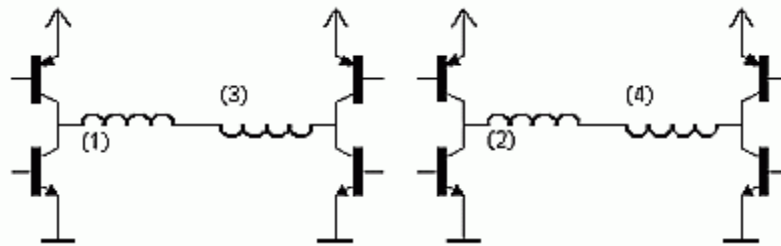
- **Unipolární**
 - *Cívky se spínají jednotlivě*
 - *Menší odběr motoru*
 - *Menší kroutící moment*



Obr. 17 – Unipolární zapojení cívek (převzato z [21])

⁴ První číslo pólové dvojice značí směr tekoucího proudu

- **Bipolární**
 - *Cívky zapojeny antisériově, mají navzájem opačně orientované magnetické pole*
 - *Proud protéká oběma cívkami*
 - *Větší krouticí moment*
 - *Vyšší spotřeba*
 - *Složitější řídicí jednotka a výkonový spínací obvod (dva H-můstky)*



Obr. 18 – Bipolární zapojení cívek, H-můstek bez ochranných diod (převzato z [21])

3.3.3.2 Spínání fází motoru

- **Jednofázové**

Magnetické pole vytváří pouze jedna cívka (u bipolárního zapojení tedy antisériově zapojení dvou cívek).[21]

- **Dvoufázové**

Magnetické pole generují shodně dvě sousední cívky, dosáhneme většího krouticího momentu, ale i spotřeby, která je dvojnásobná oproti jednofázovému zapojení.[21]

3.3.3.3 Chod kroku motoru

- **S plným krokem**

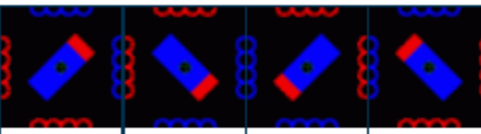
Na jednu celou otáčku je potřeba tolik kroků, kolik má motor zubů.[21]

- **S polovičním krokem**

Příklad jsme si již uvedly v popisu principu otáčení rotoru. Na otočení motoru je potřeba dvojnásobný počet kroků. Dosáhneme 2x většího rozlišení kroku hřídele.[21]

3.3.3.4 Aplikovaná konfigurace

Použit byl hybridní krokový motor s osmi vývody, každá cívka má zvlášť vyvedeny přívody napájení. Můžeme se tedy rozhodnout pro volbu zapojení cívek. Zvoleno bylo bipolární zapojení. Upřednostňujeme větší krouticí moment motoru. Do série zapojených cívek je zařazen $15\ \Omega$ odpor pro ochranu vinutí. Řízení spínání cívek je s plným krokem, přesnost je dostatečná a zachováme krouticí moment.



	-	+	+	-
Cívka 1	-	+	+	-
Cívka 2	-	-	+	+
Cívka 3	+	-	-	+
Cívka 4	+	+	-	-

Obr. 19 – Dvoufázové bipolární spínání cívek, s plným krokem (převzato z [21])

4 Návrh řídicí elektroniky a realizace

Zařízení je mobilní a využíváno převážně venku, mimo zdroj síťového napětí. Velikost napájecího napětí závisí na typu akumulátoru. Od velikosti vstupního napětí vybereme vhodný motor a funkci logických obvodů zajistíme stabilizátorem napětí.

4.1 Popis funkce elektroniky

Vysvětlíme si jednotlivé funkční bloky a konkrétní zapojení součástek.

4.1.1 Napájení

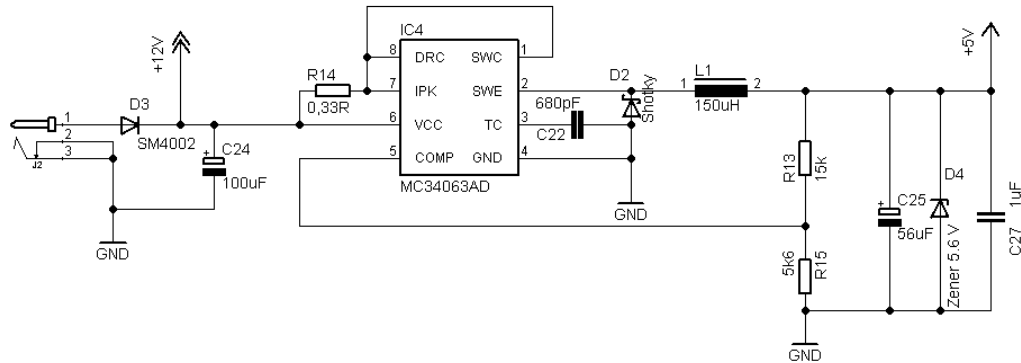
Zdrojem napětí je 12 V olověný akumulátor s kapacitou 7,5 Ah. Výdrž akumulátoru je na běžné použití předimenzována (využito kvůli jeho vlastnictví). Baterie je umístěna mimo posuv, odložena na zemi. Hmotnost baterie je značná 2,2 Kg, pro přenos na větší vzdálenosti doporučuji jiný typ akumulátoru.

Akumulátor napájí vstupní elektroniku, stabilizátor a větev do H-můstku pro výkonové spínání motoru. Elektroniku je možno napájet i síťovým adaptérem, je nutné pamatovat na odběrové proudové špičky krokového motoru. Stabilizátor zajistí doporučenou úroveň napětí pro logické obvody (i díky velké napájecí toleranci procesoru), může však dojít k omezení proudu pro motor a poklesu momentu. Rozhodnete-li se pro napájení ze sítě, doporučuji adaptér s tvrdší proudovou charakteristikou.

4.1.2 Stabilizátor

Použit je spínaný stabilizátor s obvodem MC34063. Obvod je zapojen jako snižující napěťový stabilizátor. Na vstupu je zařazena obyčejná usměrňující dioda, tvořící ochranný prvek proti přepólování. Vstupní filtrační kondenzátor C24. Rezistor R14 omezuje maximální výstupní proud, obvod vyhodnocuje úbytek napětí na rezistoru. Kondenzátor C22 nastavuje pracovní frekvenci oscilátoru. Shottkyho dioda D2, musí mít rychlou dobu sepnutí (t_{rr}), kvůli pulzovému vybíjení cívky v době rozepnutí stabilizátoru, kdy se indukčnost vybíjí a pokrývá odběr zátěže spolu s koncovým kondenzátorem. Dělič napětí R13 a R15, určuje hodnotu (1,25 V) pro komparátor. Na konci obvodu jsou zařazeny opět filtrační kondenzátory. Zenerova dioda vykonává funkci přepětíové ochrany, v případě průrazu nedojde k poškození nízkonapěťových prvků, ale pouze zdrojové části. Pro ulehčení výběru základních součástek

zajišťujících funkčnost obvodu, existují softwarové kalkulátory. V nich si nadefinujeme pouze vstupní a výstupní napětí, proud odebíraný zátěží, případně frekvenci oscilátoru. Z těchto zadaných požadavků získáme doporučené hodnoty jednotlivých součástek.[22]



Obr. 20 – Stabilizátor, schéma zapojení

Porovnání lineárního a spínaného stabilizátoru

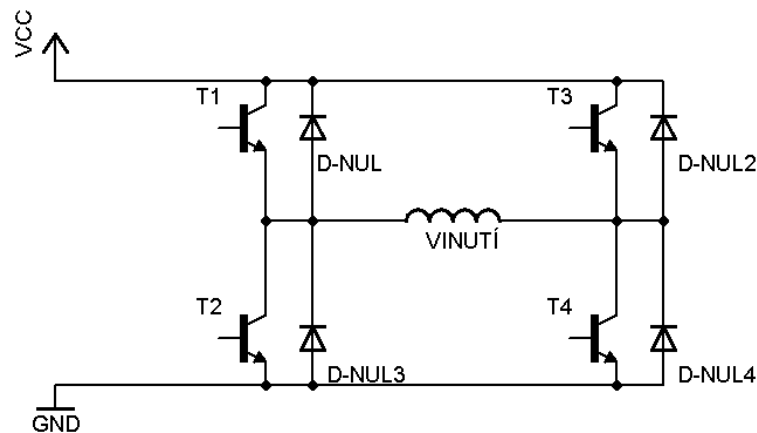
V tomto srovnání bych se zaměřil na účinnost a vyzdvihl rozdíl ve vzniku tepelných ztrát. Spínaný stabilizátor vhodným otevíráním a zavíráním tranzistoru řídí velikost přiváděného napětí na koncový kondenzátor, respektive zátěž. V sepnuté periodě je výkon dodáván do zátěže stabilizátorem, po uzavření tranzistoru se stává zdrojem kondenzátor, který je dobíjen energií z cívky (LC filtr) a vykryvá ztráty odebírané zátěží. Tímto řízením chodu dosáhneme větší účinnosti, tranzistor buď vede, nebo ne. I tady se však neobejdeme bez ztrát. Výkonová ztráta vzniká úměrně s frekvencí spínání tranzistoru (spínací ztráty), při sepnutém stavu ztráta závisí na odebíraném proudu zátěží a úbytku napětí na tranzistoru ($P_z = I * U_{ce}$). Vhodným výběrem součástek dosáhneme na výstupu rovný nezvlněný průběh napětí. Se vzrůstající a především časově proměnlivou zátěží zvlnění vzrůstá. Přesto pro malé zátěže a omezený zdroj energie (akumulátor) je spínaný stabilizátor výhodný.[23]

Lineární stabilizátor je nenáročný na zapojení, vyžaduje méně obslužných součástek. Regulace výkonu je zde plynulá, rychle reaguje na proudové změny zátěže. Rozdíl napětí (vstupní - výstupní) je úbytek napětí nacházející se na stabilizátoru. Vynásobíme-li toto napětí proudem zátěže, dostaneme ztrátový tepelný výkon. Tepelné ztráty musí být pomocí chladiče odvedeny ze součástky. Regulujeme-li napětí velikostně značně odlišné (maximálně o tři desítky voltů) úměrně roste i tepelná ztráta. Stabilizátor se v konečném důsledku chová jako proměnný rezistor v sérii se zátěží, kontrolující jen výstupní napětí a zbytek napětí s procházejícím proudem mění v tepelné ztráty.[23]

$$P_z = (U_{vst} - U_{výst}) * I \quad (3)$$

4.1.3 Výkonové spínání motoru

Výkonovou kontrolu nad motorem provádí integrovaný obvod L298. Obvod obsahuje dva H-můstky. Můstek umožňuje vhodným sepnutím dvojic tranzistorů měnit polaritu výstupního napětí. Změnu polarity napětí využijeme např. při reverzaci chodu stejnosměrného motoru, nebo při řízení krokového motoru, kdy měníme polaritu napětí na cívkách motoru k docílení pohybu rotoru. Činnost sledu spínání je řízena procesorem.[24]



Obr. 21 – H-můstek, obecné zapojení

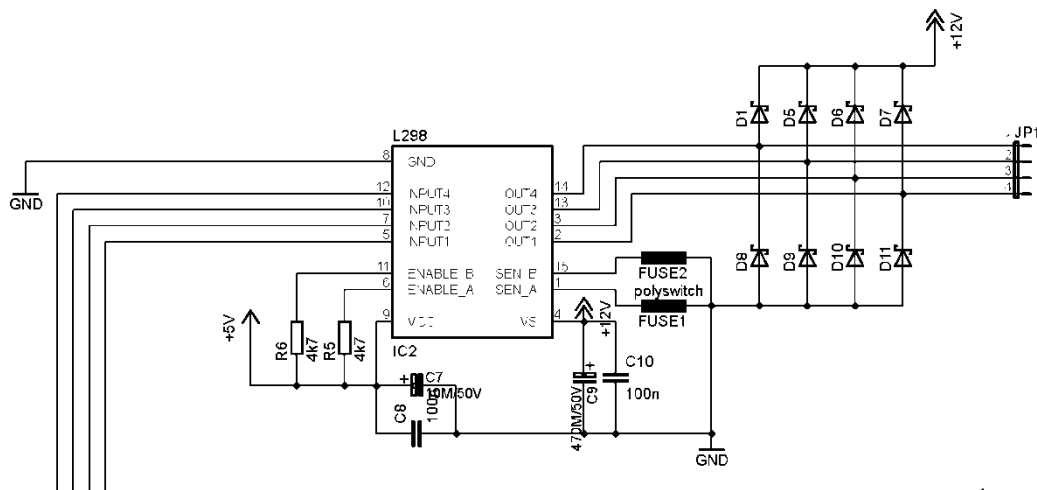
Obvod má tedy čtyři logické vstupy, předávající informaci o sledu spínání na výkonový výstup. Na výstupu jsou připojeny nulovací diody, zabráňující vzniku napěťových špiček. Ty vznikají po odpojení indukční zátěže. Magnetické pole vyzářené kolem cívky se naindukuje zpět do indukčnosti (s opačnou polaritou) a přemění se na energii elektrického pole. Podle vztahu (4) se v počátku rozeznutí blíží napětí k nekonečnu. Někde v obvodu, musí nastat průraz (elektrický oblouk), aby se energie uzavřela. Tímto jevem by došlo k poškození výkonových tranzistorů v H-můstku. Proto jsou zde zařazeny nulovací diody s rychlou dobou sepnutí ($t_{tr} = 50 \text{ ns}$), neboli doba za kterou dioda dokáže vést proud po přiložení napětí.[25]

$$U_L = L * \frac{di_L}{dt} \quad (4)$$

Vstupy enable A, B zpřístupňují ovládání H-můstků. Jsou připojeny přes pull-up rezistor na 5 voltů, high level. Logická 1 zajistí průchod informací přes hradlo, neboli povolí řízení můstku.[24]

Obvod L298 je napájen 5 V pro informační povely hradel a 12 V přiváděny k výkonovým tranzistorům, čekající na sepnutí od hradel. Co nejbližše vstupům jsou umístěny filtrační kondenzátory. Keramický (100n) pro filtrování vysokofrekvenčního rušení a elektrolytický pro vykrytí úbytků napětí při výkonovém buzení vinutí motoru. Na logické větvi stačí 10uF a na výkonové je doporučeno 470uF .[25]

Vývod 1 a 15 jsou uzemněny přes proudové pojistky, takzvaně polyswitch, neboli vratná pojistka. Slouží k ochraně motoru, abychom nespálili vinutí, ale i k ochraně H-můstku (při nevhodném sepnutí tranzistoru). Výstupní proud z H-můstku je uzavírán právě přes polyswitch na zem. Do cesty lze umístit i rezistory s malým odporem pro snímání protékajícího proudu.[25]

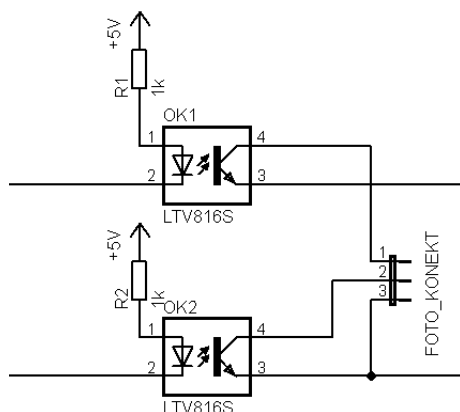


Obr. 22 – Obvod L298, zapojení

4.1.4 Spínání spouště fotoaparátu

Fotoaparát má vyvedeny tři piny pro snímání pomocí drátové spouště. Je zde vyvedena zem (GND), signál pro pořízení snímku, signál pro fixaci expozičních hodnot a zároveň nastavující správné ohnisko (ostření). Po spojení některého signálu se zemí se provede uvedená funkce.

Spínání je tedy velice jednoduché. Avšak aby nemohlo dojít k nepředpokládanému chování fotoaparátu nebo rozdílnému potenciálu úrovní země, jsou signály spínány pomocí optočlenu. Ten zajistí dostatečnou izolaci od řídicí elektroniky a fotoaparátu.



Obr. 23 – Zapojení optočlenů, dálkové ovládání spouště fotoaparátu

4.1.5 Zobrazovací jednotka

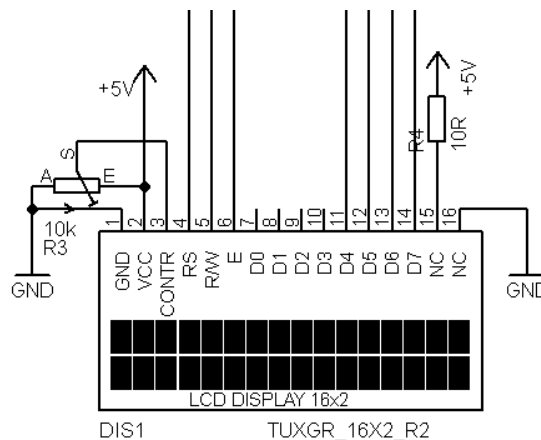
Jedná se o alfanumerický maticový display s dvaceti sloupci a čtyřmi řádky, neboli zobrazením 20 x 4 znaků. Display je osazen na vlastním plošném spoji s pomocnými obvody, takzvaným řadičem. To zjednodušuje ovládání zvenčí. V řadiči je nahraná např.: sada znaků, z které už pouze vybereme požadovaný symbol a určíme pozici kurzoru. Vyvedeny jsou piny pro napájení obvodu, řízení intenzity posvícení, kontrolu kontrastu symbolů, obsluhu řízení a datový přenos. Pořadové uspořádání je vypsáno v tabulce 1. Toto sestavení pinů je klasické pro znakové displeje s řadičem HD44780.[26]

Tab. 1 – Pořadové označení pinů displeje s řadičem HD44780 (převzato z [15])

ČÍSLO PINU	SYMBOL	POPIS
1	V_{SS}, GND	GND napájení displeje
2	V_{DD}, V_{CC}	+5V napájení displeje
3	V_0	Pin pro nastavení kontrastu LCD
4 - 6	$RS, R/W, E$	Řízení řadiče
7 - 14	$DB0 - DB7$	Datové piny
15	$LED+$	Anoda podsvícení displeje
16	$LED-$	Katoda displeje

Kontrast symbolů regulujeme potenciometrem o hodnotě 10 k Ω , zapojeným mezi napájení displeje (pin 1 a 2). Podsvícení displeje je připojeno přes rezistor (možno i přes diodu, získáme úbytek napětí 0,6 V, závisí na velikosti napájení) snižující velikost hladiny napětí na doporučenou úroveň 4,2 – 3,8 V. Odběr podsvícení je 180mA, při 5 V napájení je vyhovující 22 Ω rezistor. Piny 4 – 10 jsou připojeny k procesoru pro následnou komunikaci a

řízení. Zbylé datové piny 11 – 14 nejsou využity, používáme pouze polovinu datových pinů, komunikace probíhá po 4 bitech dvoucyklově. (Samozřejmě můžeme použít i komunikaci po celých 8 bitech, ale zabereme více pinů procesoru). [26, 27]



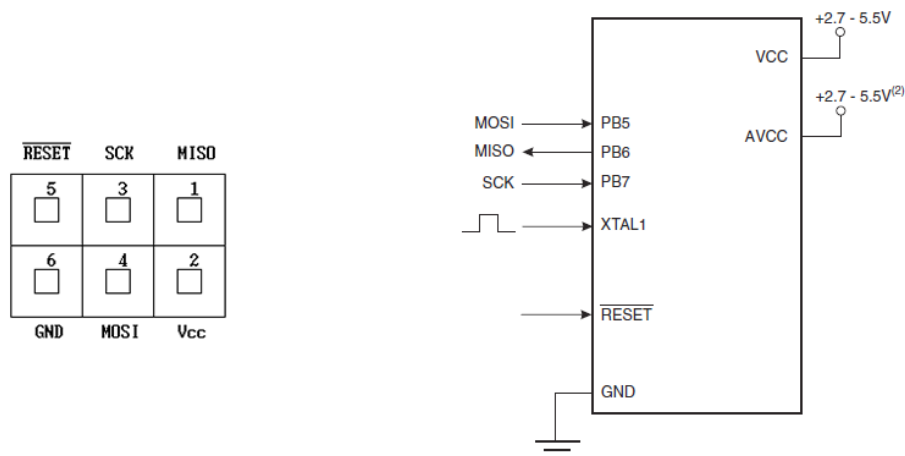
Obr. 24 – Zapojení displeje s řadičem HD44780, (verze s menším počtem maticových okének)

4.1.6 Řídící mikroprocesor a podřadné periférie

Mikroprocesor byl vybrán od firmy Atmel, pocházející z rodinné řady ATmega32. Jedná se o 8-mi bitový procesor s 32kB flash pamětí. Disponuje A/D převodníkem, 16-bitovým časovačem a mnoha dalšími funkcemi viz datasheet [28]. Má čtyři vstupní/výstupní brány. Využity jsou bezmála všechny. Takt udává externí oscilátor, krystal 8 MHz. Všude u napájecích vstupů jsou umístěny blokovací kondenzátory (100nF), filtrující možné vysokofrekvenční rušení.

K mikrokontroléru jsou připojeny všechny logické jednotky vyžadující řízení. Byly zmíněny před uvedením mikroprocesoru. Obvod L298 byl celý připojen k jedné bráně, kvůli snadnější možnosti inkrementace a cyklickému měnění polarity cívek motoru. Koncové spínače je vhodné připojit na výstupy přerušení INTERRUPT, které po sepnutí ihned odskočí z běhu programu a zastaví motor. Pro funkci výstupu Bluetooth je nutné připojit piny na TxD RxD výstupy. Ostatní prvky lze připojit libovolně (krom programovacího konektoru, zmíněného níže), tak jak vám budou vyhovovat na desce plošného spoje. Tlačítka slouží k ovládní pojezdu a nastavování parametrů.

Procesor je programován pomocí ISP konektoru. Během komunikace je přijímán synchronizační signál SCK. Vstupní data jsou přiváděny na pin MOSI. Poté zpět vyčítána z procesoru a kontrolována shodnost zápisu pomocí pinu MISO. Po naprogramování je procesor restartován.



Obr. 25 – Zleva uspořádání zapojení konektoru ISP, naznačen tok dat (převzato z [28])

4.2 Návrh plošného spoje

Návrh plošného spoje byl realizován až po výběru motoru a způsobu upevnění ke kolejm. Zbýlý prostor měl být využit k uložení části elektroniky. Plošný spoj je tedy vyříznut ve tvaru písmena L, abychom využili co nejlépe prostor, i místo nacházející se mezi kolejemi a konstrukcí pro upevnění motoru. Takto jsme získali nejmenší možnou celistvou plochu, která se následně dala zakrýt do vhodné krabičky. Samotný konec úzkého pásu je využit pro konektory vedoucí do kolejiště. Spodní strana pruhu se používá pro obvody vyžadující chlazení. Blízký profil kotvící motor je dobře tepelně vodivý a lze využít jako chladič pro integrovaný obvod L298 (H-můstek).

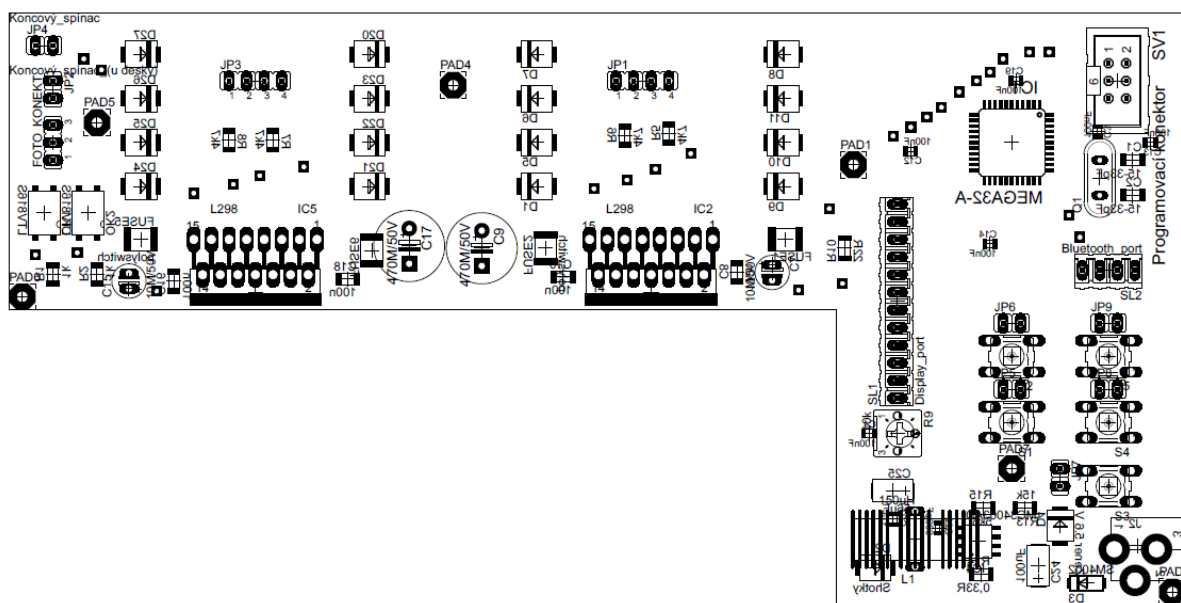
Návrh byl realizován v programu Eagle dle schématu zapojení, které jsem mohl sestavit ještě před výběrem motoru. Z důvodů dodržení rozměrů k možnosti zakrytí elektroniky krabičkou, musel být plošný spoj několikrát zmenšován. Každá změna si vyžádala malou úpravu součástek a především cest. V konečném řešení máme kolem celého obvodu cestu potenciálu GND rozlitou pomocí polygonu po celé desce s izolační mezerou 0,3 mm. Za vstupním konektorem máme 12 V výkonovou větev napájející motor a stabilizátor. Napětí 5 V rameno pro logiku se dělí na dvě strany, stoupající k procesoru, levá strana je propojena na vrchní vrstvu, aby přepažila spodní trasy H-můstků a dostala se do vzdáleného křídla. Řídící cesty od procesoru jsou taženy 0,4 mm cestou, opět v levé části vyvedeny na druhou stranu plošného spoje a svedeny k H-můstkům, část pokračuje dále ke konektorům. Nejsložitější bylo zapojení obvodu L298, jenž se musel vejít do zúžené části kvůli chlazení. Z této příčiny pravý obvod nemá ideálně řešenou výkonovou napájecí větev, která je v místě mezi nožičky obvodu vedena čarou tloušťky 0,3 mm. Jinak ostatní cesty jsou dostatečné šířky.

4.2.1 Rozmístění součástek na DPS

V důsledku minimalizace je plošný spoj oboustranný, vrchní (Top) a spodní (Bottom) vrstva. Vrchní vrstva rozvádí především signálové cesty a pomocné propojky. Všechny součástky a konektory jsou připájeny na spodní vrstvě. Blok spínaného stabilizátoru je umístěn v dolní části plošného spoje. Z tohoto celku je vývodová pouze cívka. Napájecí konektor je zabudován v čele krabičky. Nad stabilizátorem je sestava konektorů pro připojení tlačítek, displeje, programátoru a Bluetooth modulu. Mezi konektory k přípojným perifériím nalezneme mikroprocesor ATmega 32. Na všech napájecích větvích je zařazen blokovací kondenzátor 100 nF umístěn co nejbližší procesoru, taktéž i kmitací krystal. Před obvodem L298 je rozestavěna dvouřadá linie rychlých diod, pojistky a vývodové elektrolytické kondenzátory. Druhý obvod nebyl zatím osazen, jeho nasazení by mohlo být využito pro řízení motoru s polovičním krokem, nebo pro motor umístěný na vozíku (otáčení fotoaparátu kolem své osy 360°). Poslední jsou optočleny (ovládání fotoaparátu) spolu se 4 piny pro připojení koncových spínačů.

4.2.2 Výroba

Plošný spoj byl nechán dělán fotocestou, přiložením masky z obou stran. Všechny linky se vyleptaly správně, nedošlo k podleptání, cesty byly celistvé nenarušené, spojovací body vrstev se stranově shodovaly. Po vyvrtání otvorů byl plošný spoj připraven pro osazení.



Obr. 26 – Rozmístění součástek plošného spoje

5 Programování a oživení obvodu

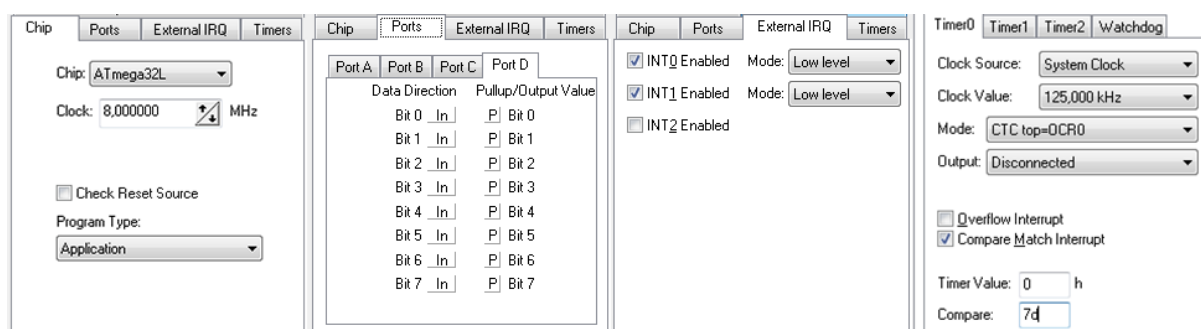
Kontrolování správné funkce obvodu probíhalo už během letování součástek v tomto sledu. Po spájení vstupního stabilizátoru a pomocných součástek byly ověřeny jeho parametry stabilizace, tedy velikost napětí. Připájení mikroprocesoru, ověření jeho komunikace s programátorem. Poté byly připájeny veškeré zbylé součástky. Některé hodnoty rezistorů bylo nutné v zapojení pozměnit v důsledku pracovního napětí v úrovni 4,86 V. Konkrétně se jednalo o rezistory před „LED diodami“, podsvícení displeje a vstup optočlenu.

5.1 Programování

Kód byl psán ve vývojářském studiu CodeVisionAVR určen pro procesory Atmel. Programovacím jazykem je C. Program podporuje několik programátorů, mezi nimiž je i používaný AVR 910. Komunikuje s PC přes rozhraní RS232 (sériový port).

5.1.1 Nastavení mikroprocesoru

Po založení projektu se otevře nové okno, kde si vybereme z nabídky možností. Nabídka je přehledná, nastavování probíhá převážně graficky. V první podsložce zvolíme typ mikrokontroléru ATmega32L (L = Low Power). Pracovní kmitočet 8 MHz, v tomto případě externího oscilátoru. Má vliv na práci čítače/časovače.



Obr. 27 – Nastavení procesoru v CodevisionAVR

5.1.1.1 Nastavení portů

V podsložce portů nalezneme veškeré brány A až D. Podle schématu přiřadíme jednotlivě pinům příznak vstup/výstup. Tlačítka přepneme na vstupy (IN) a zapneme podporu pull-up rezistorů (P). Brána A je celá výstupní (OUT) s počáteční logickou úrovní 0. Optočlenu jsou připojeny na PB3 a PB4, příznak výstupu s počáteční logickou úrovní 1 (obvod je připojen na

log. 1 a uzavírá se přes procesor na zem). Definování výstupní brány připojenou k displeji najdeme pod ALPHANUMERIC LCD (více u zobrazovací jednotky).

5.1.1.2 Vnější přerušení

Zaškrtneme podporu přerušení, INT0 i INT1, mode low level. Reaguje pouze na nulovou úroveň (vozík drží tlačítko stisklé).

5.1.1.3 Časovač

Zapneme Timer0, zdroj taktu necháme systémový (8MHz), děličkou kmitočtu získáme a zvolíme 125,000 KHz. Mód CTC top, po načítání koncové hodnoty zásobníku se smaže, čítá znovu od nuly. Zaškrtneme-li políčko COMPARE MATCH INTERRUPT a do okýnka compare zapíšeme hodnotu 7D. Čítač počítá impulzy od oscilátoru, počet načítaných pulzů porovnává se zadanou hodnotou, dosáhne-li shody, dojde k přerušení programu, čítač se vynuluje a čítá znovu. Takto zajistíme pravidelné přerušování běhu programu, nutné pro přesné časové spínání cívek motoru. Přepínání cívek vinutí musí odpovídat jednotným intervalům, abychom zachovali plynulý točivý pohyb. Hodnotu 7D jsme získali následovně: Spočteme periodu kmitů 8 μ s, vynásobíme ji 256 (maximální hodnota osmi bitového čítače).

$$T = \frac{1}{f} = \frac{1}{125000} = 8\mu\text{s} \quad (5)$$

$$8b = 2^8\text{dec} = 256\text{dec} \quad (6)$$

Od časovače dostaneme přerušení každé 2 ms. Našemu použití však nevyhovuje. Požadujeme poloviční dobu. Hodnota 125 již splňuje požadavek. Každou milisekundu se odskočí od cyklu programu a proběhne příkaz ve funkci časovače. Hexadecimální číslo 7D získáme převodem ze 125.

$$8\mu\text{s} * 256 = 2\text{ms} \quad (7)$$

$$8\mu\text{s} * 125 = 1\text{ms} \quad (8)$$

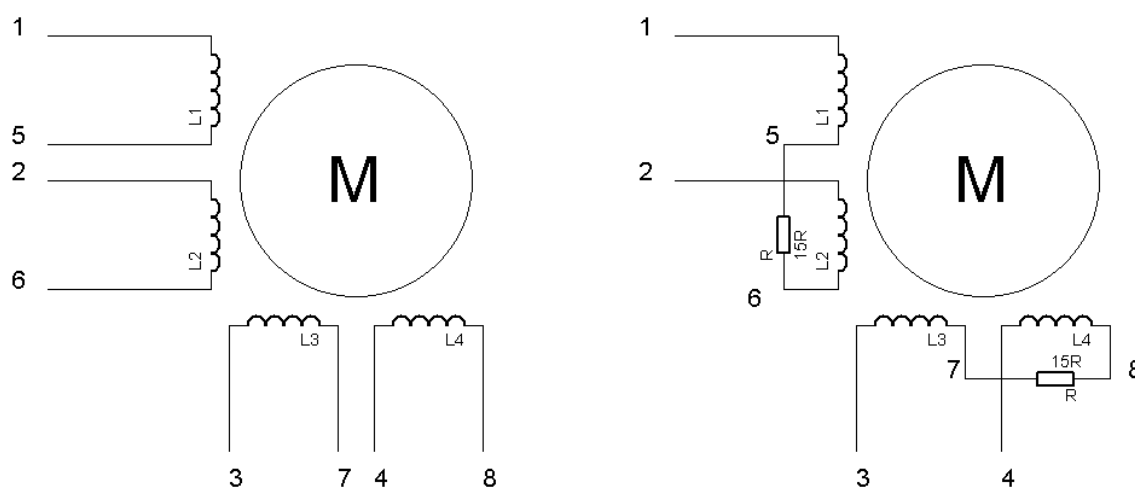
Navolené nastavení v grafickém editoru si poté vygenerujeme do kódu, s kterým dále pracujeme v textové formě.

5.1.1.4 Programování fuse bitů

Nastavení pojistek mikroprocesoru probíhá při prvním programování. Konkrétní popisy jednotlivých pojistek vyhledáme v datasheetu. Uvedené funkce čteme pečlivě, po špatné volbě by se mohl procesor uzamknout, nebo přestat komunikovat. Nás zajímá především přepnutí z interního RC oscilátoru na vnější krystalový oscilátor s menší odchylkou kmitání. Registr CKOPT nastavíme do 1, CKSEL3..1 všechny přepnout do 1. V menu nastavení si však dáme pozor. Zaškrtnutím⁵ políčka uvádíme registr do 0. Toto může být trochu matoucí.

5.1.2 Roztočení motoru

První cíl bylo roztočení krokového motoru, jeho funkce byla zkoušena pouze s manuálním přikládáním napětí na piny. Po zhlédnutí schéma zapojení sledu cívek motoru jsem je připojil k obvodu L298. Jednu sériovou kombinaci cívek na první H-můstek a druhou na druhý H-můstek.



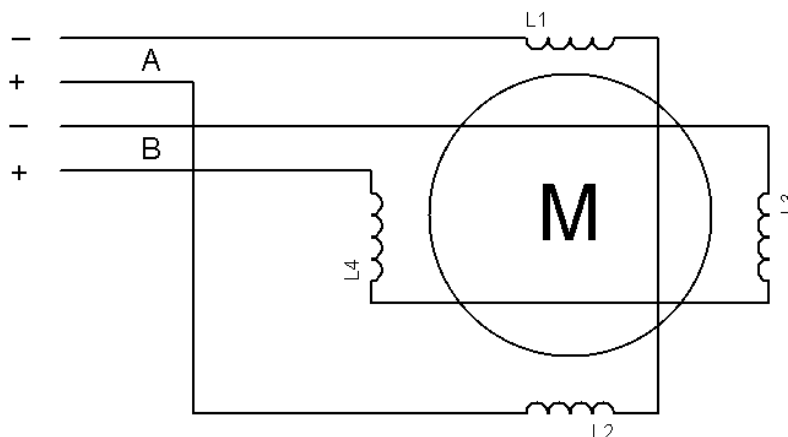
Obr. 28 – Vlevo, drátové vývody jednotlivých cívek. Vpravo zapojení cívek (antisériové) s ochranným rezistorem 15Ω omezující proud pod konstrukční 1 A.

Piny procesoru do L298 jsou nastaveny jako výstupní. Střídáním spínání pinů roztočíme rotor. Polaritu spínání měníme ve sledu viz. tabulka 2. Rychlostí spínání cívek měníme rychlost rotoru i velikost protékaného proudu, který s rychlostí klesá (klesá i moment).

⁵ V mém případě jsem odškrtnul všechna pole nastavení. Stačí však nastavit registry pro výběr zdroje kmitů a procesor bude fungovat správně.

Tab. 2 – Posloupnost spínání motoru

Cívka 1	-	+	+	-
Cívka 2	+	-	-	+
Cívka 3	-	-	+	+
Cívka 4	+	+	-	-



Obr. 29 – Rozmístění vinutí po obvodu motoru, magnetické působení cívek musí být jednotné. V tomto případě proti směru hodinových ručiček.

Nadefinovali jsme si 4 hodnoty (PHASE0 až PHASE3) každé z nich jsme nastavili posloupnost polarit (z tabulky 2), kladná úroveň znamená logickou 1, záporná logickou 0. Prvky PHASE vložíme do pole. Inkrementací (postupné navyšování) prvky jednotlivě přepínáme a měníme směr proudu tekoucí přes cívky. Po dosažení konce pole PHASE3 se opět vracíme na začátek PHASE0.

```
//Definování výstupní brány
#define PHASE0      0xA0 // 10 10: - + - +
#define PHASE1      0x90 // 10 01: + - - +
#define PHASE2      0x50 // 01 01: + - - -
#define PHASE3      0x60 // 01 10: - + + -

//Vytvoření pole
unsigned char _motor_phase[] = {PHASE0, PHASE1, PHASE2, PHASE3};
```

Proud tekoucí přes cívky je nutné nechat procházet určitou dobu, aby vytvořily dostatečně velké magnetické pole a přitáhly rotor do nové pozice. Limitující rychlost⁶ motoru nastala při intervalovém spínání 3 ms, poté se magnetická vazba mezi státorem a rotorem roztrhla. Rychlost otáčení rotoru je převrácenou hodnotou spínání jednotlivých cívek motoru

⁶V tuto chvíli nastavení čítače nebylo ještě využito, jednalo se pouze o testování chodu motoru. Z limitující hodnoty 3 ms vycházelo nastavení čítače. Struktura programu v této fázi nebyla některak složitá a chod motoru byl plynulý.

Koncové spínače jsou připojeny na výstupy přerušení INTO a INT1 reagující na trvalou nulovou úroveň, vstupní piny jsou nastaveny s pull-up rezistorem. Dojde-li ke stisknutí, sepnou se do logické 0. Motor se ihned zastaví, zůstane zabrzděn a přeruší se inkrementace v poli.

```
#define LIMIT_SWITCH_START    (PIND.3 == 0)    // koncový spínač 1 (u motoru)
#define LIMIT_SWITCH_END      (PIND.2 == 0)    // koncový spínač 2

#define CHOICE_BOTTOM_LEFT    (PINC.0 == 0)    // kurzor (volba) doprava
#define CHOICE_BOTTOM_RIGHT   (PIND.5 == 0)    // kurzor (volba) nahoru

void main()
{
    //Krok doprava, podmíněn koncovým spínačem

    if(CHOICE_BOTTOM_RIGHT && !LIMIT_SWITCH_START)
    {
        PORTA = _motor_phase[_phase];

        _phase++;
        if(_phase > 3) _phase = 0;
        delay_ms (3);
    }

    //Krok do leva
    if(CHOICE_BOTTOM_LEFT && !LIMIT_SWITCH_END)
    {
        PORTA = _motor_phase[_phase];

        _phase--;
        if(_phase < 0) _phase = 3;
        delay_ms (3);
    }
}
```

Nyní můžeme začít posouvat pole jen v opačném směru, dekrementace. Po rozepnutí koncového spínače, můžeme směr posuvu opět měnit. Inkrementaci a dekrementaci měníme klávesami.

Dále požadujeme zjistit počet kroků k přejetí dráhy vozíku. Z tohoto údaje budeme schopni určit přibližnou polohu a využijeme ji i v dalších aplikacích. Platformu přesuneme k okraji, do funkce kroku vložíme novou proměnou, kterou necháme s krokem motoru přičítat a zobrazit ji na displeji. Výsledek byl pokaždé jiný, lišil se v řádu jednotek kroků v průměru o 4. Jako referenční hodnotu jsme zvolili 8200 kroků.

5.1.3 Zobrazovací jednotka

Zelenožluté podsvícení displeje bylo příliš oslňující, musíme snížit vstupní napětí, regulovatelným zdrojem byla vybrána vhodná intenzita a vypočítán odpor 34Ω (předtím byla vložena pouze dioda). Nastavení kontrastu znamenalo přetočit trimr na plnou výchylku, otočný rezistor lze tedy odstranit a pin trvale uzemnit.

LCD jednotka je osazena driverem HD44780. Tento řadič od firmy Hitachi se stal standardem pro ovládání řádkových alfanumerických displejů. K procesoru je připojena čtveřice datových vodičů posílající informaci po čtyřech bitech a trojici řídicích signálů. E hodinový vstup, RS volba instrukce/data, RD výběr módu čtení/zápis.[29]

V průvodci CodeVisionAVR máme možnost přímého nastavení znakového displeje, vybereme typ řadiče HD44780, počet znaků na řádek 20 a přiřadíme jednotlivé výstupy dle schématu zapojení. Studio nám vygeneruje zdrojový kód s řídicími funkcemi a přidá hlavičkový soubor s jejich definicí. V generovaném kódu si všimneme, že inicializace displeje je už provedena.

```
/ Alphanumeric LCD initialization
// Connections specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 1
// RD - PORTC Bit 2
// EN - PORTC Bit 3
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 20
  lcd_init(20);
  delay_ms(10);
```

Přizpůsobuje odesílání po 4 bitech dvoucyclově. Odešlou se horní 4 bity inicializačního slova a pulz na hodinový vstup E. Celé slovo se poté pošle znovu, ale už 4 bitovým způsobem. Dvoucyclové odesílání probíhá takto: Po obdržení hodinového signálu E načteme horní polovinu bytu a v druhém pulzu E, načteme spodní polovinu bytu odeslané informace.[29] Chceme-li zobrazit nové symboly, LCD vždy smažeme a poté znaky zapíšeme. K výběru konkrétní polohy kurzoru používáme systém souřadnic x, y.

```
lcd_clear();
delay_ms(3);
lcd_gotoxy(1, 0);
lcd_puts("LINEARNI POSUN");
```

Textový řetězec určen k zobrazení je psán do uvozovek, každý znak je převeden do číselného kódu dle ACII tabulky, jemu odpovídající hodnota v paměti řadiče je zastoupena určitým symbolem (soupis nalezneme v datasheetu), jenž je zobrazen na displeji. První znak textového řetězce leží na pozici kurzoru. Jednotka umožňuje nadefinování 8 vlastních znaků.

b7- b3 b4 - b0		0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
0000	CG RAM (1)		0	a	P	`	P		-	9	3	α	p	
0001	(2)	!	1	A	Q	a	a	。	ア	チ	4	ä	q	
0010	(3)	"	2	B	R	b	r	「	イ	ツ	×	ß	ø	
0011	(4)	#	3	C	S	c	s	」	ウ	テ	E	ε	ø	
0100	(5)	\$	4	D	T	d	t	、	イ	ト	ト	μ	Ω	
0101	(6)	%	5	E	U	e	u	=	オ	ナ	1	ε	Ü	
0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ	
0111	CG RAM (8)	'	7	G	W	g	w	フ	キ	ヌ	ラ	g	π	
1000	CG RAM (1)	<	8	H	X	h	x	ィ	ウ	ホ	リ	フ	×	
1001	(2)	>	9	I	Y	i	y	ッ	ケ	ル	」	」	y	
1010	(3)	*	:	J	Z	j	z	エ	コ	ン	レ	j	¥	
1011	(4)	+	:	K	L	k	l	ク	オ	サ	ヒ	ロ	*	斤
1100	(5)	,	<	L	¥	1	1	ハ	シ	フ	フ	φ	円	
1101	(6)	-	=	M	I	m	i	ユ	ズ	ン	ン	ト	÷	
1110	(7)	.	>	N	^	n	→	ヨ	セ	ホ	ン	ン		
1111	CG RAM (8)	/	?	O	_	o	←	ッ	ソ	マ	°	ö	■	

Obr. 30 – Tabulka znaku pro řadič HD44780 (převzato z [30])

5.1.4 Režimy pojezdu

Po zasunutí napájecího konektoru probíhá inicializace polohy. Motor navijí vozík do startovní polohy (strana s řídicí elektronikou). Teprve po dojezdu na koncový spínač se vozík zastaví. Získáme počáteční polohu 0. Po rozjetí přičítáme pulzy od spínání cívek. Přejetí dráhy na druhý konec odpovídá cca 8200 kroků. Po dojetí na koncový spínač je počet kroků vždy aktualizován a dojde ke kalibraci. Z počtu kroků je zobrazena procentuálně poloha vozíku. Dokončí-li se inicializace je zobrazena nabídka menu. V levé části jsou vypsány módy, do kterých lze vstoupit a definovat operační hodnoty. Vpravo máme naznačeny symboly, či prvním písmenem slova význam tlačítek, uspořádány do kříže. Rozpoložení odpovídá tlačítkům na krabičce. Znaky k jednotlivým spínačům jsou uváděny z důvodu proměnlivé funkce. Horní levé tlačítko slouží k rotaci v nabídce menu. Kurzor rotace (šipka) je zobrazen před vybranou možností. Výběr potvrzujeme pravým horním tlačítkem. Dolními spínači provádíme změny hodnot, přičítáme nebo odečítáme jedničku k výsledku.

5.1.4.1 Auto (mód 1)

Tento mód byl vytvořen pro tvorbu časosběrů. V podmenu pracujeme se třemi možnostmi.

- *čas přejezdu dráhy; minuty, hodiny*
- *počet chystaných snímků*
- *délka expozice*

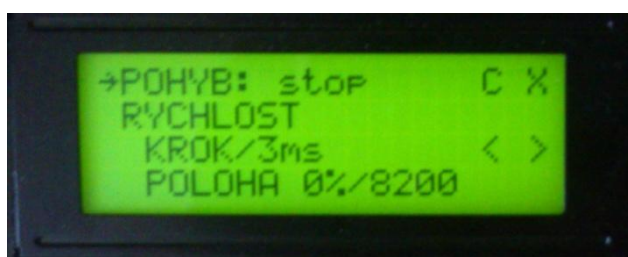
Víme-li kolik kroků je přibližně nutno k přejetí posuvu, snímky rozložíme s jednotným krokováním. Vrátime se zpět do hlavního menu a odstartujeme posuv. V tomto módu připojujeme k fotoaparátu kabel k řízení otevírání závěrky. Snímání fotoaparátu musí být synchronizováno s krokováním motoru. V době expozice je vozík zastaven, probíhá focení. Jedná-li se o noční focení s dlouhou expozicí, pauzu před samotným snímáním je nutné protáhnout z důvodů dokmitávání vozíku. Doba přejezdu dráhy, je konečný čas přejetí celých kolejí. Algoritmus obsahuje korekce intervalu snímání, zastavení a rozjetí vozíku.



Obr. 31 – Menu Auto

5.1.4.2 Manuál (mód 2)

Pohyb je manuálně ovládán klávesami. Používáme jej pro video záznam. Tlačítka vybíráme směr posuvu, po přepnutí volby měníme tlačítka rychlost. Natáčení video sekvence zapínáme na fotoaparátu. Na displeji vidíme aktuální polohu vozíku vypsanou v procentech a rychlost udanou v časovém intervalu spínání cívek motoru.



Obr. 32 – Menu manuál

Závěr

Posuvy pro kamery či fotoaparáty se na trhu vyskytují již delší období, nicméně není uveden žádný standard a model od modelu se obvykle liší. Konstrukční materiály se mohou taktéž odlišovat. Jedinou možností může být vyzkoušení konkrétního posuvu (obvykle nelze), věřit uživatelským recenzím, nebo shlédnout prezentující video (nejlépe od 3 strany). Z těchto příčin jsem se rozhodl pojezd sestavit sám. Nemalou roli v tom hrálo i moje amatérské užití a konečná cena posuvu. Hotové sety pojezdů byly pro mě cenově nedostupné. Ceny veškerého fotopříslušenství jsou celkově vysoké.

Postavením vlastního pojezdu jsem se snažil vyhnout konstrukčním vadám pojezdů vyskytujících se na trhu. Věděl jsem, kde mohu čekat slabá i silná místa a vše připravit pro moje potřeby. Konečná délka dráhy posuvu se zastavila na 1,45 m. Pohybová možnost vozíku je o jeho délku kratší 1,30 m. S vlastním provedením konstrukce, lze vždy něco zlepšit. Protože se jedná o běžně dostupné profily z lehkých slitin, úprava či vyztužení dílů je možná. Do budoucna plánuji v první fázi odstranit spojovací plechy mezi kolejemi a nahradit je pouze jedním celistvým plechem po celé délce. V druhém kroku, podepřít kolejnice dalším L profilem. Rozhodl jsem se, že upřednostním pevnost celku před hmotností. Pracujeme-li v terénu, není obvykle čas kolejje perfektně připravit a vypodložit, bylo by to i časově náročné při neustále se měnící scéně a přesouvání se na jinou pozici. Postavíme-li kolejje na dva stativy, konec s motorem je výrazně prohnut, spojovací plech potřebuje přemístit do těžiště kolejí v blízkosti motoru. Hmotnost motoru v porovnání se zbývající konstrukcí je značně rozdílná, proto po vyztužení sestavy se pocitová váha nezmění.

Pohon funguje překvapivě dobře (vyndán z 8“ disketové mechaniky, 70. léta). Horizontální posun zvládá bez obtíží, avšak při určitých rychlostech dochází k lokálnímu rozvibrování vozíku, nebo celé konstrukce. Posuv při naklonění cca 45° zdolá převýšení pouze pro pomalé rychlosti, kdy dosahuje velkého momentu. Potenciální výkon motoru není však plně využit, připojené ochranné 15 Ω rezistory omezují proud na necelých 400 mA (max. 1000mA). Rezistory je možné vyměnit za menší, nebo je překlenout a řídit průchod proudu pulzně šířkovou modulací (PWM). Jednalo by se především o regulaci proudu v zabrzděném stavu, motor stojí na místě, cívky jsou stále připojeny ke zdroji napětí, proud je omezován jen odporem vynutím a regulován PWM pod úrovní 1 A. Odpojením napájení

ztratíme brzdny moment, vozík se nekontrolovatelně rozjede, je-li posuv nakloněn. Konstrukce funguje i jako dobrý tepelný chladič. Vnější teplota motoru byla srovnatelná s teplotou rámu. Menší část tepla byla odvedena i kolejiemi.

Displej nelze softwarově otočit, fyzicky taktéž nelze provést kvůli výstupním pinům určeným k připojení konektoru. Po otočení by zasahovaly do hřídele motoru. Piny lze pouze odletovat a připájet vodiče, které povedeme těsně po povrchu displeje. Zatím se k zobrazovací jednotce přistupuje z opačné strany, než bylo plánováno. Pokud v praxi uvidím, že mi tento přístup vyhovuje, zachovám jej. Z důvodů úspor energie je vhodné vložit vypínač mezi přívod napájení podsvícení displeje.

Fotoaparát je napájen vlastním akumulátorem s výdrží 1650 mAh. Vyzkouším, jak zvládne 9 hodinové focení. Nebude-li kapacita dostatečná, fotoaparát bych dobíjel z hlavní 12 V baterie přes stabilizátor.

V programovém výběru módů chci dodělat režim s prioritou momentu neboli maximální rychlosti. Od této veličiny spolu s počtem snímků a délkou expozicí dopočítat konečný čas. Nasazení módu by se uplatnilo u vertikálního posunu. Vertikální zdvih v tomto zapojení motoru (s rezistory) není však proveditelný. Proto nebyl zatím zaveden.

Seznam literatury a informačních zdrojů

- [1] "AlamoFilming" by Sean Devine. Licensed under CC BY-SA 3.0 via Wikimedia Commons [online]. 2004 [cit. 2015-05-25]. Dostupné z: <http://commons.wikimedia.org/wiki/File:AlamoFilming.jpg#/media/File:AlamoFilming.jpg>
- [2] Cameraslider: use plastic pipes. Youtube.com [online]. 2011 [cit. 2015-05-25]. Dostupné z: <http://i.ytimg.com/vi/bnjif5KDjSI/hqdefault.jpg>
- [3] Camera Rail. DHgate [online]. 2015 [cit. 2015-05-24]. Dostupné z: <http://www.dhgate.com/product/free-ship-mini-desktop-camera-rail-car-table/160483948.html>
- [4] Sevenoak Camera Slider. Inter-store.cz [online]. 2015 [cit. 2015-05-24]. Dostupné z: <http://www.inter-store.cz/produkt/Sevenoak-Camera-Slider-Bracket-60cm-SK-GT01-p4779176.html>
- [5] Filmcity SL-2. The cinecity [online]. 2015 [cit. 2015-05-24]. Dostupné z: <http://www.thecinecity.com/eshop/Filmcity-SL-2-DSLR-Camera-Slider.html>
- [6] DSLR camera slider. Jiankou.net [online]. 2015 [cit. 2015-05-24]. Dostupné z: <http://www.jiankou.net/jiankoushop/goods.php?id=5962>
- [7] Zl1600 camera slider. Zolinger.com [online]. 2014 [cit. 2015-05-24]. Dostupné z: <http://zolinger.com/zl1600-camera-slider/>
- [8] MicroWheel Stage-3. Rigwheels.com [online]. 2014 [cit. 2015-05-24]. Dostupné z: <http://www.rigwheels.com/product/camera-slider/>
- [9] Glide Gear Video Camera Slider. Amazon [online]. 2015 [cit. 2015-05-24]. Dostupné z: <http://www.amazon.com/Glide-Gear-Video-Camera-Slider/dp/B00SNVA3EG>
- [10] SME otevřený linearset. Cncshop.cz [online]. 2010 [cit. 2015-05-24]. Dostupné z: <http://www.cncshop.cz/sme20-otevreny-linearset-pr-20mm>
- [11] Výkon a točivý moment. Hotrally.sweb.cz [online]. 2004 [cit. 2015-05-27]. Dostupné z: <http://hotrally.sweb.cz/tech/teorie.html>
- [12] Stejnoseměrné stroje (motory). 2004. Katedra elektroniky v Ostravě [online]. [cit. 2015-05-09]. Dostupné z: <http://fei1.vsb.cz/kat430/old/Studium/Materialy/VS/kap22.pdf>
- [13] Modelářské servo. 2015. Wikipedie [online]. [cit. 2015-05-09]. Dostupné z: http://cs.wikipedia.org/wiki/Model%C3%A1%C5%99sk%C3%A9_servo
- [14] Modelářská serva: Základní informace. 2011. POHOŘELÝ, Viktor. Pojezdy.eu [online]. [cit. 2015-05-09]. Dostupné z: <http://pojezdy.eu/view.php?nazevclanku=modelarska-serva-zakladni-informace&cisloclanku=2011070004>
- [15] Modelářská serva: úprava na 360° otáčení. 2011. POHOŘELÝ, Viktor. Pojezdy.eu [online]. [cit. 2015-05-09]. Dostupné z: <http://pojezdy.eu/view.php?nazevclanku=modelarska-serva-uprava-na-360%EF%BF%BD-otaceni&cisloclanku=2011070007>
- [16] Motor DC: Komutátor. 2013. D' AFAN, rohman. [Http://rohmandafan.blogspot.cz](http://rohmandafan.blogspot.cz) [online]. [cit. 2015-05-08]. Dostupné z: http://2.bp.blogspot.com/_BYILVBKey0c/TSDbTNPUK6I/AAAAAAAAAOA/02tnB3vMw6M/s1600/komutator.PNG
- [17] Servo motor: Princip. 2015. Pohonná technika [online]. [cit. 2015-05-09]. Dostupné z: <http://www.pohonnatechnika.cz/skola/motory/servo-motor>
- [18] Serva. 2012. Moraviaracing [online]. [cit. 2015-05-09]. Dostupné z: <http://www.moraviaracing.cz/clanky/serva/>

- [19] Hrabovcová, Valéria, Ladislav Janoušek, Pavol Rafajdus a Miroslav Ličko. Moderné elektrické stroje. Žilina: EDIS, 2001. ISBN 80-7100-809-5.
- [20] Krokový motor: Princip. 2015. Pohonná technika [online]. [cit. 2015-05-09]. Dostupné z: <http://www.pohonnatechnika.cz/skola/motory/krokovy-motor>
- [21] Krokové motory. 2002. ŘEZÁČ, Kamil. Robotika.cz [online]. [cit. 2015-05-09]. Dostupné z: <http://robotika.cz/articles/steppers/cs>
- [22] Když stabilizátor příliš topí... 2013. BEZSTAROSTI, Jiří. Robodoupe.cz [online]. [cit. 2015-05-14]. Dostupné z: <http://robodoupe.cz/2013/kdyz-stabilizator-prilis-topi/>
- [23] Spínané zdroje. BABČANÍK, Jan. Hw.cz [online]. 2007 [cit. 2015-05-16]. Dostupné z: <http://www.hw.cz/teorie-a-praxe/spinane-zdroje.html>
- [24] L298 - datasheet. STMicroelectronics [online]. 2000 [cit. 2015-05-17]. Dostupné z: <http://www.st.com/web/en/resource/technical/document/datasheet/CD00000240.pdf>
- [25] Aplikační poznámky k L298. , JRT. Robodoupě.cz: web o robotice [online]. 2011 [cit. 2015-05-17]. Dostupné z: <http://robodoupe.cz/2011/aplikacni-poznamky-k-l298-integrovanemu-h-mustku/>
- [26] Znakové LCD. HW Kitchen [online]. 2014 [cit. 2015-05-18]. Dostupné z: <http://www.hwkitchen.com/news/a13-arduino-a-displeje-ii-/>
- [27] LCD Display: 4x20 characters. HW Kitchen [online]. 2013 [cit. 2015-05-18]. Dostupné z: <http://www.hwkitchen.com/products/lcd-display-4x20-characters/>
- [28] ATmega32 - datasheet. Atmel.com [online]. 2011 [cit. 2015-05-19]. Dostupné z: <http://www.atmel.com/images/doc2503.pdf>
- [29] HD44780 Based LCD. FLEURY, Peter. Hispeed.ch [online]. 2003 [cit. 2015-05-28]. Dostupné z: <http://homepage.hispeed.ch/peterfleury/avr-lcd44780.html>
- [30] Ovládání znakových LCD s řadičem HD44780. Elektronika.kvalitne.cz [online]. 2007 [cit. 2015-05-29]. Dostupné z: <http://elektronika.kvalitne.cz/ATMEL/necoteorie/LCDmatice.html>

Přílohy

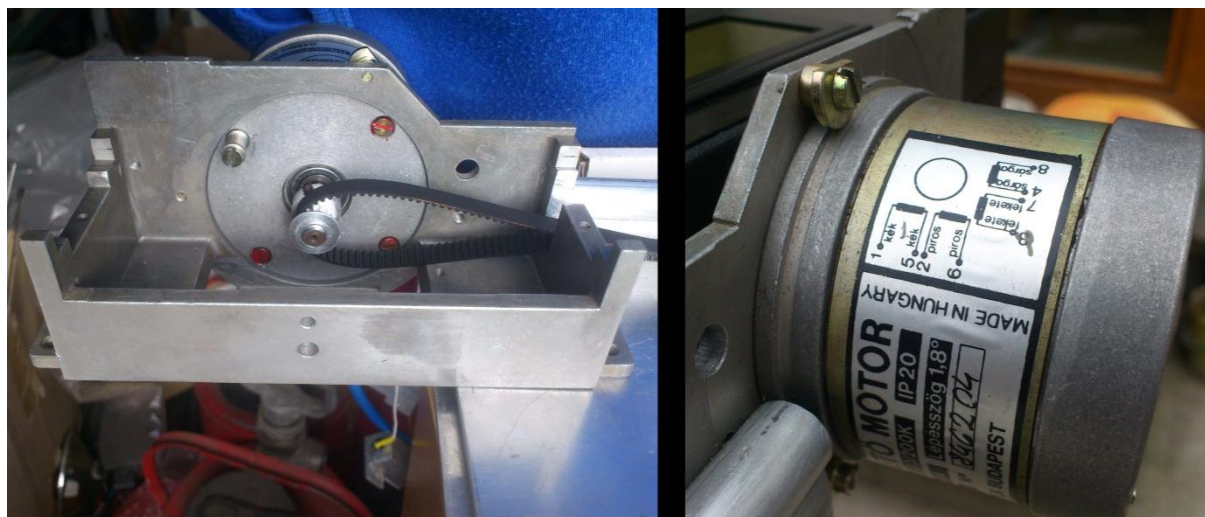
Přílohy na CD-ROM:

- Elektronická verze práce
- Dokumentace konstrukce
- Návrh plošného spoje – Eagle
- Řídící program – CodevisionAVR
- Ukázkové video
- Datasheet k hlavním periferiím

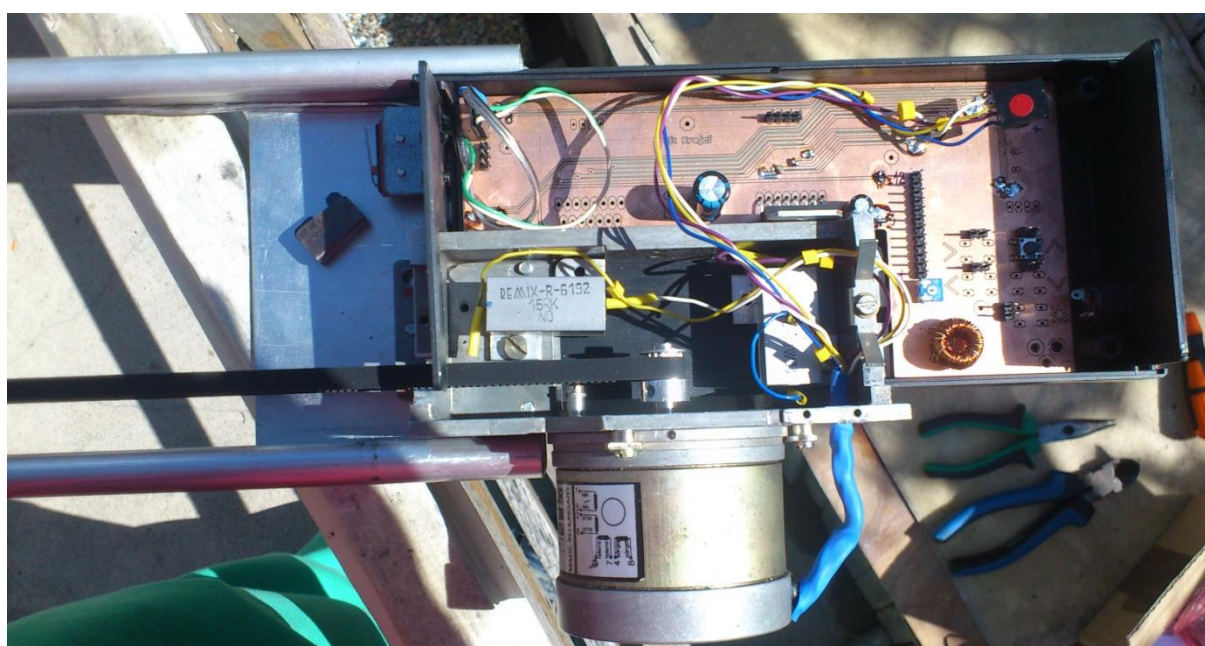
Příloha A – Fotodokumentace konstrukce výrobku



Obr. 1A – Zhotovené kolejiště s ložiskovým vozíkem



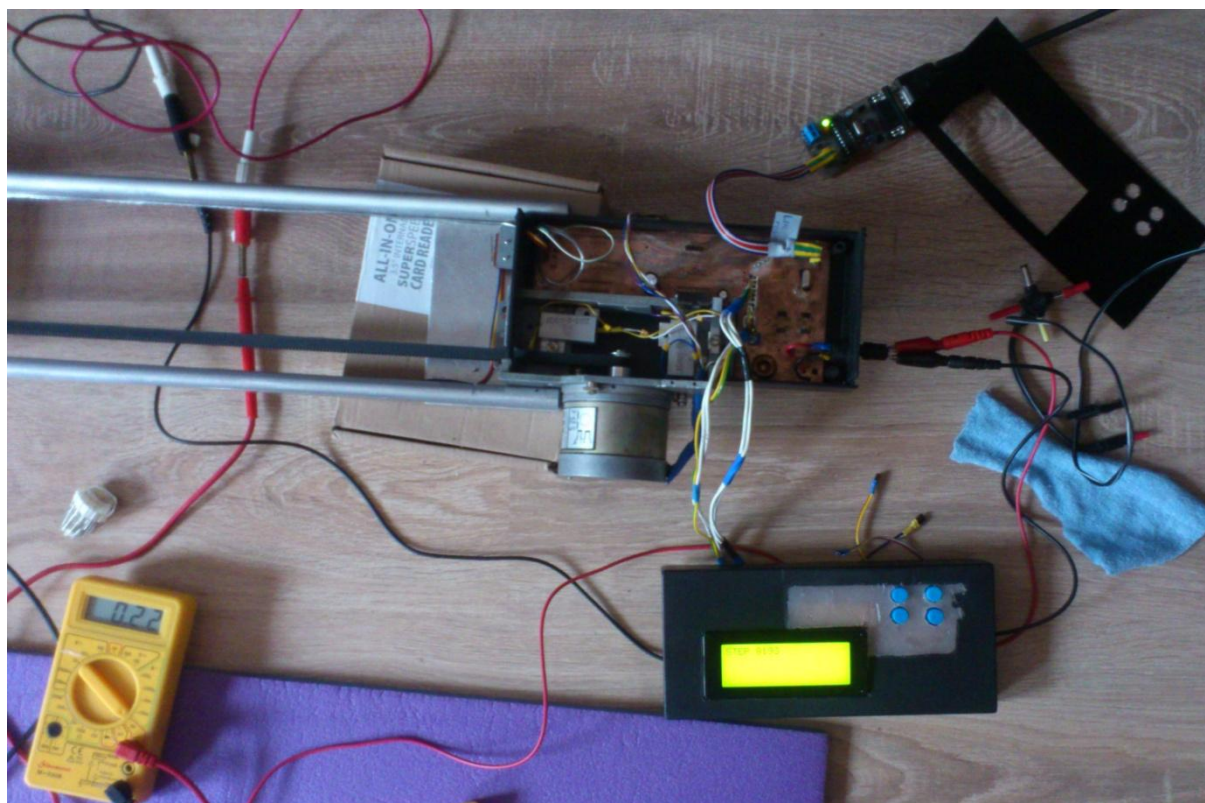
Obr. 2A – Uchycení, upevnění motoru



Obr. 3A – Uložení do krabčky



Obr. 4A – Stativová hlava

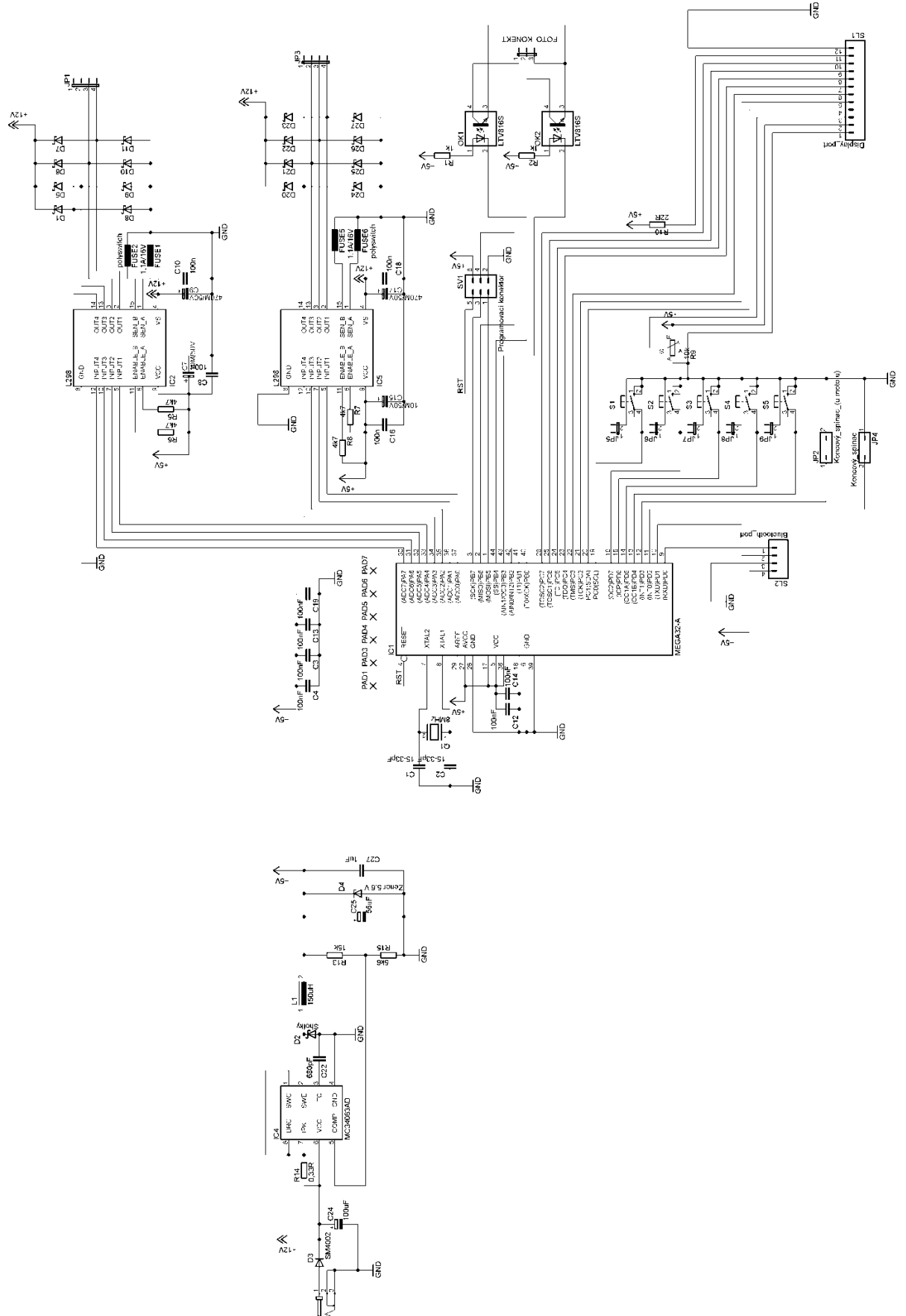


Obr. 5A – Průběh programování

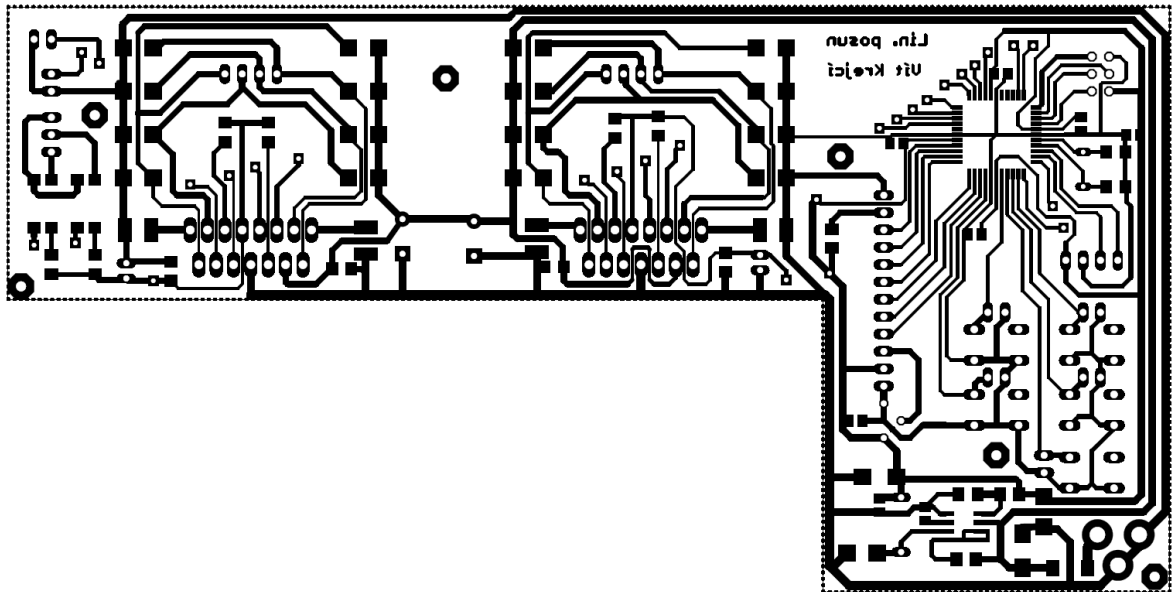


Obr. 6A – Rozložení na stativech

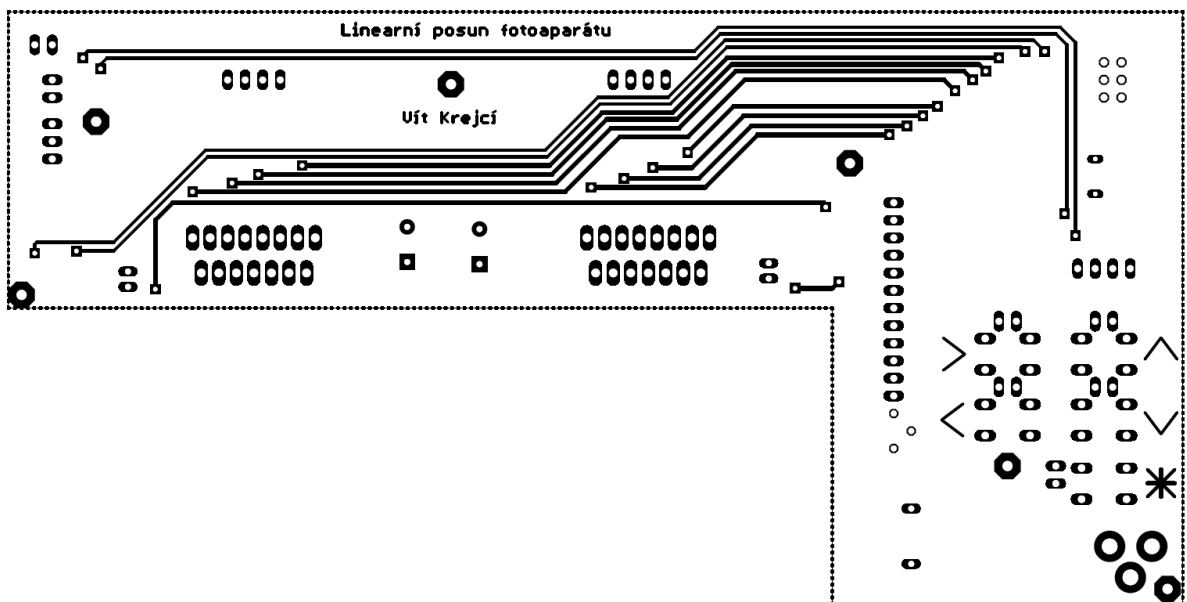
Příloha B – Návrh plošného spoje



Obr. 1B – Schéma



Obr. 2B – Plošný spoj, spodní vrstva



Obr. 3B – Plošný spoj, horní vrstva

Příloha C – Řídící program

```

/*****

```

```

This program was produced by the
CodeWizardAVR V2.05.0 Professional
Automatic Program Generator
© Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

```

```

Project : Lineární posun
Version : 1.0
Date   : 21.5.2015
Author : Vít Krejčí
Company :
Comments:

```

```

Chip type       : ATmega32L
Program type    : Application
AVR Core Clock frequency: 8,000000 MHz
Memory model    : Small
External RAM size : 0
Data Stack size : 512

```

```

*****/

```

```

#include <mega32a.h>

```

```

// Alphanumeric LCD Module functions
#include <alcd.h>

```

```

#include "slider.h"

```

```

// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void) //koncový spínač přerušení
{
    slider_inEndPosition();
    auto_stopAutomaticMode();
}

```

```

// External Interrupt 1 service routine //počáteční spínač přerušení
interrupt [EXT_INT1] void ext_int1_isr(void)
{
    slider_inStartPosition();
}

```

```

// Timer 0 output compare interrupt service routine
interrupt [TIM0_COMP] void timer0_comp_isr(void)
{
    static unsigned long int tick = 0;

    if(tick >= step_delay_ms)
    {
        slider_makeStep();
        tick = 0;

        auto_takeShot();
    }
    else

```

```
{
    tick++;
}

// CPU initialization
void init(void);

// Main function
void main(void)
{
    init();

    menu_showMenu();
    slider_initSlider();
    menu_setMenu();

    while(1)
    {
        menu_checkInputs();
        menu_showMenu();
        delay_ms(50);
    }
}

// CPU initialization
void init(void)
{
    // Input/Output Ports initialization
    // Port A initialization
    // Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
    // State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
    PORTA=0x00;
    DDRA=0xFF;

    // Port B initialization
    // Func7=In Func6=In Func5=In Func4=Out Func3=Out Func2=In Func1=In Func0=In
    // State7=T State6=T State5=T State4=1 State3=1 State2=P State1=P State0=P
    PORTB=0x1F;
    DDRB=0x18;

    // Port C initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=P
    PORTC=0x01;
    DDRC=0x00;

    // Port D initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
    // State7=P State6=P State5=P State4=P State3=P State2=P State1=P State0=P
    PORTD=0xFF;
    DDRD=0x00;

    // Timer/Counter 0 initialization
    // Clock source: System Clock
    // Clock value: 125,000 kHz
    // Mode: CTC top=OCR0
    // OC0 output: Disconnected
    TCCR0=0x0B;
    TCNT0=0x00;
```

```
OCR0=0x7D;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Falling Edge
// INT1: On
// INT1 Mode: Falling Edge
// INT2: Off
GICR|=0xC0;
MCUCR=0x0A;
MCUCSR=0x00;
GIFR=0xC0;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x02;

// USART initialization
// USART disabled
UCSRB=0x00;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
```

```
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

// Alphanumeric LCD initialization
// Connections specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 1
// RD - PORTC Bit 2
// EN - PORTC Bit 3
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 20
lcd_init(20);

// Global enable interrupts
#asm("sei")
}
```



```

/*****
Project : Lineární posun
Version : 1.0
Date   : 21.5.2015
Author  : Vít Krejčí
Company :
Comments:
*****/

#ifndef _SLIDER_DRIVER_INCLUDED_
#define _SLIDER_DRIVER_INCLUDED_

#include <delay.h>
#include "util.h"          // Správné zobrazení čísel

//kroky motoru
#define PHASE0          0xA0 // 10 10: + +
#define PHASE1          0x90 // 10 01: + -
#define PHASE2          0x50 // 01 01: - -
#define PHASE3          0x60 // 01 10: - +

//koncové spínače
#define LIMIT_SWITCH_START (PIND.3 == 0)    // koncový spínač 1 (u motoru)
#define LIMIT_SWITCH_END   (PIND.2 == 0)    // koncový spínač 2

unsigned char motor_phase[] = {PHASE0, PHASE1, PHASE2, PHASE3}; //pole s fázemi motoru
signed char phase = 0;    // fáze krokového motoru

unsigned long int step_delay_ms = 3;          //zpoždění mezi kroky v ms

volatile unsigned int max_position = 8200;    //maximální poloha v krocích
volatile unsigned int max_position_per = 82;  //faktor pro výpočet polohy v procentech

volatile unsigned int position = 0;          //poloha od počátku

signed char direction = 0;                  //směr jízdy
/* -1 : jízda do leva
 * 0: stop
 * +1: jízda do prava
 */

//textové buffery pro zobrazení na LCD
char step_delay_str[10] = {'3', '\0', '\0', '\0', '\0', '\0', '\0', '\0', '\0', '\0'};
volatile char position_str[6] = {'0', '\0', '\0', '\0', '\0', '\0'};
volatile char max_str[6] = {'8', '2', '0', '0', '\0', '\0'};

void slider_incSpeed()                      //Zvyšování rychlosti = snížit výčkávací dobu
{
    if(step_delay_ms > 3)
    {
        step_delay_ms--;

        longToStr(step_delay_ms, step_delay_str);    //Převodění aktualně nastaveného čísla do
bufferu
    }
}

void slider_decSpeed()                      //Snížení rychlosti
{
    step_delay_ms++;

```

```

        longToStr(step_delay_ms, step_delay_str);
    }

void slider_incDirection()
{
    if(direction < 1)
    {
        direction++;
    }
}

void slider_decDirection()                //Směr posuvu
{
    if(direction > -1)
    {
        direction--;
    }
}

//nastavení zpoždění = 1/rychlost mororu
char slider_setStepDelayInMs(unsigned long int step_delay)
{
    if(step_delay <= 3)
    {
        step_delay_ms = 3;
        step_delay_str[0] = '3';
        step_delay_str[1] = '\0';

        return 0;
    }
    else
    {
        step_delay_ms = step_delay;
        longToStr(step_delay_ms, step_delay_str);

        return 1;
    }
}

//vrácení polohy v procentech
unsigned char slider_getPositionInPercent()
{
    return (unsigned char)(position / max_position_per);
}

//!!! NUTNÉ ZAVOLAT V INTERRUPTU
void slider_inEndPosition()
{
    direction = 0;
    max_position = position;
    max_position_per = max_position / 100;

    intToStr(max_position, max_str);
}

//!!! NUTNÉ ZAVOLAT V INTERRUPTU
void slider_inStartPosition()
{
    direction = 0;

```

```
    position = 0;

    intToStr(position, position_str);
}

//krok do prava
void slider_stepIntoRight()
{
    if(!LIMIT_SWITCH_START)
    {
        PORTA = motor_phase[phase];

        phase++;
        if(phase > 3) phase = 0;

        position--;
        intToStr(position, position_str);
    }
}

//krok do leva
void slider_stepIntoLeft()
{
    if(!LIMIT_SWITCH_END)
    {
        PORTA = motor_phase[phase];

        phase--;
        if(phase < 0) phase = 3;

        position++;
        intToStr(position, position_str);
    }
}

//učíň krok
void slider_makeStep()
{
    if(direction < 0)
    {
        slider_stepIntoLeft();
    }
    else if(direction > 0)
    {
        slider_stepIntoRight();
    }
}

//inicializace slideru do výchozí polohy, po zapnutí
void slider_initSlider()
{
    #asm("cli")
    while(!LIMIT_SWITCH_START)
    {
        PORTA = motor_phase[phase];
        delay_ms(3);

        phase++;
        if(phase > 3) phase = 0;
    }
}
```

```
    #asm("sei")  
}  
#endif
```

```

/*****
Project : Lineární posun
Version : 1.0
Date   : 21.5.2015
Author  : Vít Krejčí
Company :
Comments:
*****/

#ifndef _AUTO_MODE_INCLUDED_
#define _AUTO_MODE_INCLUDED_

#include <delay.h>

#include "slider_driver.h"
#include "util.h"

#define SHOT      PORTB.4    // ovládání závěrky (vytvoření snímku)
#define FOCUS     PORTB.3    // ostření a fixace kompozice

unsigned long int hours = 0;
unsigned long int minutes = 1;

unsigned long int time_ms = 0;

unsigned int actual_spacing = 0;
unsigned int spacing = 0;

unsigned long int number_of_shots = 10;
unsigned long int exposition = 1;

unsigned char run = 0;

//textové buffery pro zobrazení na LCD
char hours_str[3] = {'0', '\0', '\0'};
char minutes_str[3] = {'1', '\0', '\0'};
char exposition_str[5] = {'1', '\0', '\0', '\0', '\0'};
char shots_str[6] = {'1', '0', '\0', '\0', '\0', '\0'};

void auto_takeShot()
{
    if((position >= actual_spacing) && (run == 2))
    {
        TCCR0=0x00;

        actual_spacing += spacing;

        delay_ms(100);

        //zaostříme
        FOCUS = 0;
        delay_ms(1000);
        //vyfotíme
        SHOT = 0;
        delay_ms(exposition * 1000);

        FOCUS = 1;
        SHOT = 1;

        delay_ms(100);
    }
}

```

```
        TCCR0=0x0B;
    }
}

void auto_startAutomaticMode()
{
    if(!LIMIT_SWITCH_END)
    {
        #asm("cli")

        if(run == 0)
        {
            slider_initSlider();
            actual_spacing = 0;

            delay_ms(1000);

            auto_takeShot();
            slider_decDirection();

            run = 2;
        }
        else if(run == 1)
        {
            slider_decDirection();
            run = 2;
        }
    }
    #asm("sei")
}

void auto_stopAutomaticMode()
{
    run = 1;

    delay_ms(100);
}

void auto_pauseAutomaticMode()
{
    if(run == 1)
    {
        slider_initSlider();
        run = 0;
    }
    else if(run == 2)
    {
        direction = 0;
        run = 1;
    }

    delay_ms(100);
}

void auto_updateTime()
{
    unsigned long int correction = number_of_shots * (1300 + (exposition * 1000));
    time_ms = (hours * 3600000) + (minutes * 60000);
}
```

```
        if(time_ms > correction)
        {
            time_ms -= correction;
            time_ms /= max_position;
        }
        else
        {
            time_ms = 3;
        }

        if(slider_setStepDelayInMs(time_ms) == 0)
        {
            //TODO: Dodělat opravu času pro zobrazení
        }
    }

    void auto_updateSpacing()
    {
        spacing = max_position / number_of_shots;
    }

    void auto_incExposition()
    {
        if(exposition < 1000)
        {
            exposition++;
            auto_updateTime();

            longToStr(exposition, exposition_str);
        }
    }

    void auto_decExposition()
    {
        if(exposition > 0)
        {
            exposition--;
            auto_updateTime();

            longToStr(exposition, exposition_str);
        }
    }

    void auto_incHour()
    {
        if(hours < 24)
        {
            hours++;
            auto_updateTime();

            longToStr(hours, hours_str);
        }
    }

    void auto_decHour()
    {
        if(hours > 0)
        {
            hours--;
```

```
        auto_updateTime();

        longToStr(hours, hours_str);
    }
}

void auto_incMinute()
{
    if(minutes < 59)
    {
        minutes++;
        auto_updateTime();

        longToStr(minutes, minutes_str);
    }
    else
    {
        minutes = 0;
        auto_incHour();

        minutes_str[0] = '0';
        minutes_str[1] = '\0';
    }
}

void auto_decMinute()
{
    if(minutes > 0)
    {
        minutes--;
        auto_updateTime();

        longToStr(minutes, minutes_str);
    }
    else
    {
        auto_decHour();
    }
}

void auto_incShots()
{
    if(number_of_shots < 10000)
    {
        number_of_shots++;
        auto_updateSpacing();
        auto_updateTime();

        longToStr(number_of_shots, shots_str);
    }
}

void auto_decShots()
{
    if(number_of_shots > 0)
    {
        number_of_shots--;
        auto_updateSpacing();
        auto_updateTime();
    }
}
```



```
        longToStr(number_of_shots, shots_str);  
    }  
}  
#endif
```

```

/*****
Project : Lineární posun
Version : 1.0
Date   : 21.5.2015
Author  : Vít Krejčí
Company :
Comments:
*****/

#ifndef _LCD_MENU_INCLUDED_
#define _LCD_MENU_INCLUDED_

#include "slider_driver.h"
#include "auto_mode.h"
#include "util.h"

//klávesnice
#define BUTTON_TOP_LEFT    (PIND.7 == 0) // kurzor (volba) doleva
#define BUTTON_TOP_RIGHT   (PIND.4 == 0) // kurzor (volba) dolů
#define BUTTON_BOTTOM_LEFT (PINC.0 == 0) // kurzor (volba) doprava
#define BUTTON_BOTTOM_RIGHT (PIND.5 == 0) // kurzor (volba) nahoru

#define INIT                -1
#define MENU0               0
#define AUTO                1
#define MANUAL              2

#define TIME                11

signed int menu = INIT;

unsigned char pointer = 0;

char position_per_str[4] = {'0', '\0', '\0', '\0'};

inline void menu_showHelpMarks(      char top_left,
                                   char top_right,
                                   char bottom_left,
                                   char bottom_right);

void menu_setMenu()
{
    menu = MENU0;
}

void menu_showMenu()                //Pokud nastane přerušení od čítače v tomto bodě, display není vykreslen
{
    lcd_clear();
    delay_ms(3);
    lcd_gotoxy(0, pointer);
    lcd_putchar('~');

    switch(menu)
    {
        case INIT:
            lcd_gotoxy(1, 0);
            lcd_puts("INICIALIZACE POLOHY");
            break;
    }
}

```

```
case MENU0:
    lcd_gotoxy(1, 0);
    lcd_puts("AUTO ");

    if(run == 2)
    {
        lcd_puts("running");
    }
    else if(run == 1)
    {
        lcd_puts("pause");
    }
    else
    {
        lcd_puts("stop");
    }

    lcd_gotoxy(1, 1);
    lcd_puts("MANUAL");

    lcd_gotoxy(2, 2);
    lcd_puts("STAV ");

    intToStr(slider_getPositionInPercent(), position_per_str);
    lcd_puts(position_per_str);
    lcd_puts("%/");
    lcd_puts(max_str);

    switch(pointer)
    {
        case 0:
            lcd_gotoxy(2, 3);
            lcd_puts(hours_str);
            lcd_puts(":");
            lcd_puts(minutes_str);
            lcd_puts("/");
            lcd_puts(shots_str);
            lcd_puts("/");
            lcd_puts(exposition_str);
            lcd_puts("s");

            menu_showHelpMarks('C', '>', 'O', 'I');
            break;

        case 1:
            lcd_gotoxy(2, 3);
            lcd_puts("POHYB ");

            if(direction > 0)
            {
                lcd_putchar('>');
            }
            else if(direction < 0)
            {
                lcd_putchar('<');
            }
            else
            {
                lcd_puts("stop");
            }
    }
}
```

```
    }

    menu_showHelpMarks('C', '>', '<', '>');
    break;
}

break;

case AUTO:
    lcd_gotoxy(1, 0);
    lcd_puts("CAS ");
    lcd_puts(hours_str);
    lcd_puts(":");
    lcd_puts(minutes_str);

    lcd_gotoxy(1, 1);
    lcd_puts("SNIMKY: ");
    lcd_puts(shots_str);

    lcd_gotoxy(1, 2);
    lcd_puts("EXPOZICE: ");
    lcd_puts(exposition_str);
    lcd_puts("s");

    menu_showHelpMarks('C', 'X', '<', '>');
break;

    case TIME:
    lcd_gotoxy(1, 0);
    lcd_puts("MINUTY: ");
    lcd_puts(minutes_str);

    lcd_gotoxy(1, 1);
    lcd_puts("HODINY: ");
    lcd_puts(hours_str);

    menu_showHelpMarks('C', 'X', '<', '>');
break;

case MANUAL:
    lcd_gotoxy(1, 0);
    lcd_puts("POHYB: ");

    if(direction > 0)
    {
        lcd_putchar('>');
    }
    else if(direction < 0)
    {
        lcd_putchar('<');
    }
    else
    {
        lcd_puts("stop");
    }

    lcd_gotoxy(1, 1);
    lcd_puts("RYCHLOST");

    lcd_gotoxy(2, 2);
```

```
    lcd_puts("KROK/");
        lcd_puts(step_delay_str);
    lcd_puts("ms");

    lcd_gotoxy(2, 3);
    lcd_puts("POLOHA ");

        intToStr/slider_getPositionInPercent(), position_per_str);
    lcd_puts(position_per_str);
    lcd_puts("%/");
    lcd_puts(max_str);

    menu_showHelpMarks('C', 'X', '<', '>');
    break;
}
}

void menu_checkInputs()
{
    switch(menu)
    {
        case MENU0:
            if(BUTTON_TOP_LEFT)
            {
                pointer++;
                if(pointer > 1)
                {
                    pointer = 0;
                }

                while(BUTTON_TOP_LEFT);
            }

            if(BUTTON_TOP_RIGHT)
            {
                switch(pointer)
                {
                    case 0:
                        menu = AUTO;
                        pointer = 0;
                        break;

                    case 1:
                        menu = MANUAL;
                        pointer = 0;
                        break;
                }

                while(BUTTON_TOP_RIGHT);
            }

            if(BUTTON_BOTTOM_LEFT)
            {
                switch(pointer)
                {
                    case 0:
                        auto_pauseAutomaticMode();
                        break;

                    case 1:
```

```
        slider_decDirection();
    break;
}

while(BUTTON_BOTTOM_LEFT);
}

if(BUTTON_BOTTOM_RIGHT)
{
    switch(pointer)
    {
        case 0:
            auto_updateTime();
            auto_updateSpacing();
            auto_startAutomaticMode();
            break;

        case 1:
            slider_incDirection();
            break;
    }

    while(BUTTON_BOTTOM_RIGHT);
}
break;

case AUTO:
    if(BUTTON_TOP_LEFT)
    {
        pointer++;
        if(pointer > 2)
        {
            pointer = 0;
        }

        while(BUTTON_TOP_LEFT);
    }

    if(BUTTON_TOP_RIGHT)
    {
        menu = MENU0;
        pointer = 0;

        while(BUTTON_TOP_RIGHT);
    }

    if(BUTTON_BOTTOM_LEFT)
    {
        switch(pointer)
        {
            case 0:
                menu = TIME;
                break;

            case 1:
                auto_decShots();
                break;

            case 2:
                auto_decExposition();
```

```
        break;
    }

    while(BUTTON_BOTTOM_LEFT);
}

if(BUTTON_BOTTOM_RIGHT)
{
    switch(pointer)
    {
        case 0:
            menu = TIME;
            break;

        case 1:
            auto_incShots();
            break;

        case 2:
            auto_incExposition();

            break;
    }

    while(BUTTON_BOTTOM_RIGHT);
}
break;

        case TIME:
if(BUTTON_TOP_LEFT)
{
    pointer++;
    if(pointer > 1)
    {
        pointer = 0;
    }

    while(BUTTON_TOP_LEFT);
}

if(BUTTON_TOP_RIGHT)
{
    menu = AUTO;
    pointer = 0;

    while(BUTTON_TOP_RIGHT);
}

if(BUTTON_BOTTOM_LEFT)
{
    switch(pointer)
    {
        case 0:
            auto_decMinute();
            break;

        case 1:
            auto_decHour();
            break;
    }
}
```

```
    while(BUTTON_BOTTOM_LEFT);
}

if(BUTTON_BOTTOM_RIGHT)
{
    switch(pointer)
    {
        case 0:
            auto_incMinute();
            break;

        case 1:
            auto_incHour();
            break;
    }

    while(BUTTON_BOTTOM_RIGHT);
}
break;

case MANUAL:
if(BUTTON_TOP_LEFT)
{
    pointer++;
    if(pointer > 1)
    {
        pointer = 0;
    }

    while(BUTTON_TOP_LEFT);
}

if(BUTTON_TOP_RIGHT)
{
    menu = MENU0;
    pointer = 0;

    direction = 0;

    while(BUTTON_TOP_RIGHT);
}

if(BUTTON_BOTTOM_LEFT)
{
    switch(pointer)
    {
        case 0:
            slider_decDirection();
            break;

        case 1:
            slider_decSpeed();
            break;
    }

    while(BUTTON_BOTTOM_LEFT);
}

if(BUTTON_BOTTOM_RIGHT)
{
    switch(pointer)
```



```
        {
        case 0:
            slider_incDirection();
            break;

        case 1:
            slider_incSpeed();
            break;
        }

        while(BUTTON_BOTTOM_RIGHT);
    }
    break;
}

inline void menu_showHelpMarks(char top_left, char top_right, char bottom_left, char bottom_right)
{
    lcd_gotoxy(17, 0);
    lcd_putchar(top_left);

    lcd_gotoxy(19, 0);
    lcd_putchar(top_right);

    lcd_gotoxy(17, 2);
    lcd_putchar(bottom_left);

    lcd_gotoxy(19, 2);
    lcd_putchar(bottom_right);
}

#endif
```