ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA ELEKTROTECHNICKÁ, Katedra řídicí techniky

DIPLOMOVÁ PRÁCE
ŘÍDICÍ SYSTÉM PRO SLUNEČNÍ KOLEKTOR

2005 Martin Špiller

ČVUT FEL Praha

Kapitola 1 ... 3
Úvod ... 3
1.1 Vznik a historie projektu ... 3
1.2 Dosažené výsledky ... 3
1.3 Struktura dokumentu ... 5
Kapitola 2 ... 6
Technický popis systému a problému ... 6
2.1 Stručný přehled současných zařízení využívajících tepelné záření pro účely ohřevu ... 6
2.2 Popis prototypového zařízení ... 6
Kapitola 3 ... 8
Analýza přesnosti polohování ... 8
3.1 Tolerance odchýlení zrcadla od nominální hodnoty ... 8
3.2 Výpočet tolerance pro vertikální osu ... 9
3.3 Výpočet tolerance pro horizontální osu ... 11
3.4 Shrnutí výsledků ... 12
Kapitola 4 ... 13
Výpočet polohy Slunce ... 13
4.1 Výpočet slunečních souřadnic ... 13
4.2 Vliv atmosférické refrakce ... 17
4.3 Implementace algoritmu v jazyce C ... 18
4.4 Dosažené výsledky a pozorování ... 20
Kapitola 5 ... 23
Systémový návrh řízení ... 23
Kapitola 6 ... 26
Návrh a realizace snímání polohy ... 26
6.1 Problémy ... 26
6.2 Klasické inkrementální čidlo ... 27
6.3 Optické čidlo HDNS-2000 ... 28
6.4 Konstrukce polohového senzoru s obvodem HDNS-2000 ... 30
6.5 Umístění senzoru na zařízení ... 32
Kapitola 7 ... 34
Návrh a realizace výkonové části ... 34
7.1 Problémy ... 34
7.2 Realizace pomocí obvodu L298 ... 34
Kapitola 8 ... 36
Návrh a realizace řídicí části ... 36
8.1 Mikroprocesor ATmega16 ... 36
8.2 Obvod reálného času RTC-62421 ... 36
8.3 Převodník MAX-232 ... 37
8.4 Popis prototypové řídicí jednotky ... 37
Kapitola 9 ... 40
Návrh řídicího algoritmu ... 40
9.1 Popis algoritmu ... 40
9.2 Vývojový diagram ... 42
Kapitola 10 ... 46
Měření a testování ... 46
Závěr ... 48
Použitá literatura : ... 49

Kapitola 1

Úvod

Vznik a historie projektu

Tato diplomová práce vznikla v rámci projektu SKORO (Navádění komerčních solárních kolektorů pomocí matematického modelu pohybu Slunce), jehož autorem je Ing. Miroslav Rotrekl, který se zabývá možnostmi získávání a využití sluneční tepelné energie pomocí fokusačních systémů. Ing. Rotrekl sestrojil prototypovou verzi takového zařízení a provedl některá základní měření, ze kterých vyplývá, že řízené sledování Slunce by mohlo významným způsobem ovlivnit celkovou účinnost systému. V lednu roku 2004 se obrátil na Ing. Zdeňka Huráka Ph.D. z Centra aplikované kybernetiky na Katedře řízení pražského ČVUT za účelem vytvoření spolupráce při studiu možností navádění systému pomocí matematického modelu pohybu Slunce. Cílem spolupráce by mělo být i vytvoření prototypu řídicího systému, který automaticky provádí polohování kolektoru, dále pak posouzení a volba typu regulace z hlediska optimalizace výkonu a z hlediska funkčních omezení zařízení a v neposlední řadě pak také řízení sytému v případě přehřívání oběhového média. Cílem je navrhnout zcela autonomní a pro uživatele co nejjednodušší systém, který by po odborné instalaci pracoval zcela automaticky. Výsledkem projektu by pak mělo být posouzení možnosti komerčního využití takového zařízení.

Dosažené výsledky

Práci jsem započal studiem astronomické literatury s cílem nalezení vhodného algoritmu na výpočet polohy Slunce. Zde jsem se zabýval mechanikou pohybu planet a studiem různých faktorů, které ovlivňují planetární cykly a vlivům, které způsobují odlišné vnímání astronomických objektů z povrchu Země. Podařilo se mi nalézt vhodný a dostatečně přesný algoritmus, který zároveň respektuje jen ty jevy, které ovlivňují výpočet do určité míry přesnosti, aby se tak nadbytečně nespotřebovával výpočetní čas řídicí jednotky. Celý algoritmus jsem poté implementoval v jazyce C. Tuto implementaci jsem použil jednak jako součást programu řídicí jednotky a zároveň také jakou součást programu, který počítá roční průběhy polohy a rychlosti pohybu Slunce v závislosti na zeměpisných souřadnicích.

V dalším kroku jsem řešil snímání polohy na polohovacím zařízení v horizontální ose, tedy azimutu. Díky jisté speciální konstrukci polohovacího rámu, jsem zvolil netradiční snímání polohy pomocí miniaturní cmos kamery, používané například v optických myších. Toto řešení bylo již úspěšně použito při snímání polohy hvězdářského teleskopu, tedy v problému v mnoha aspektech podobnému tomuto. Podařilo se mi navrhnou a zkonstruovat tento senzor. Návrh obvodu a plošného spoje jsem prováděl v programu Eagle. Výsledný tištěný spoj byl proveden osvitem na kuprexitovou destičku, kterou bylo třeba dále zpracovat z důvodu speciálních rozměrů cmos kamery.

K podrobnějšímu řešení snímání polohy ve vertikální ose, elevaci, jsem se nedostal, neboť v době mé práce na projektu nebylo k dispozici zrcadlo, bez něhož tuto polohu snímat nelze. Toto snímání však bude možné , po jistých konstrukčních úpravách na polohovacím zařízení, které bude třeba provést pro předchozí případ, řešit stejným způsobem jako pro horizontální osu.

Poté jsem se začal zabývat návrhem řídicí a polohovací jednotky. Zvolil jsem mikroprocesor architektury AVR, který tvoří ústřední část řídicí jednotky. Nejprve jsem se zabýval implementací algoritmu na výpočet polohy na tomto procesoru. K dispozici jsem měl univerzální programovací jednotku pro procesory AVR, STK-500. Pro procesory AVR existuje řada programovacích a ladicích prostředí, ve kterých lze programovat v jazyce C. Algoritmus se mi podařilo implementovat s tím omezením, že používal 32-bitovou aritmetiku oproti původní 64-bitové. Při porovnání hodnot počítaných v obou prostředí se výsledky lišily od třetího desetinného místa. Pro přesné hvězdářské teleskopy by to mohl být problém, avšak pro toto polohovací zařízení jsou tyto výsledky naprosto dostačující. Dále jsem využil toho, že programovací jednotka umožňuje připojovat na vstupy procesoru vnější signály pro zprovoznění komunikace mezi senzorem a procesorem. V této fázi jsem však ještě nemohl ověřit správnou funkci a přesnost senzoru, neboť jsem neměl k dispozici konstantní zdroj otáčivého pohybu.

Dále jsem se věnoval zprovozněním komunikace mezi procesorem a RTC hodinami, které tvoří velice důležitou součást řídicí jednotky. Tuto komunikaci jsem uskutečnil nejprve pomocí propojení kontaktního nepájivého pole s STK-500. Posléze jsem se rozhodl z řady praktických důvodů navrhnout celou řídicí jednotku na univerzálním pájivém poli. Tuto jednotku jsem zhotovil a obsahuje jednak zmíněný obvod reálných hodin, dále pak výkonový obvod pro ovládání motoru, interface pro komunikaci s PC a je na ní připojen polohový senzor. Řídicí jednotku a senzor jsem poté připevnil na polohovací rám. Testoval jsem ovládání motoru a funkčnost polohového senzoru. Připevnění senzoru se však zatím potýká s jistými konstrukčními problémy a v současné fázi se hodí pouze pro laboratorní a testovací podmínky. Proto jsem vytvořil návrh konstrukčních opatření, které bude na zařízení nutno provést pro venkovní použití.

Při testování zařízení jsem zjistil některé nedostatky, zejména velice pomalou rychlost otáčení rámu, která je dána tím, že zařízení používá poměrně nevýkonný motor v kombinaci s převodovým systémem. Při polohové regulaci bude mít systém dlouhé časové konstanty, které nebude možné díky omezenému akčnímu zásahu vhodnou regulací zkrátit. Přesto je zapojení zpětné vazby nezbytné, například pro potlačení chyb (zrcadlem může být pohnuto díky větru apod.).

Další úskalí představuje fakt, že zařízení zatím nemá pevné aretační body, které jsou nezbytné při použití relativního polohového senzoru pro získání informace o úhlové poloze. Z tohoto důvodu není zatím možné plné zapojení zpětné vazby s regulátorem. I tak v návrhu počítám s obecným návrhem PID regulátoru, který bude realizován softwarově, což vychází z celkové digitální realizace řídicí jednotky a ze způsobu snímání polohy. V současné době mám však implementovanou pouze jeho proporcionální složku. Také jsem se již nestihl zabývat matematickým modelem montáže a jeho simulacemi v prostředí matlab, neboť tato

fáze návrhu je spjatá s ověřováním výsledků na reálném sytému, které není z výše uvedených důvodů zatím možné.

Návrh polohování je proveden tak, že systém bude nastavován pomocí vypočítané pozice Slunce s konstantním krokem. Mezi těmito kroky bude pozice Slunce korigována pomocí fotosenzitivního prvku. Tato fáze je v návrhu zmiňována, ale je sní počítáno spíše do budoucna, neboť pro její realizaci je potřeba provést další konstrukční opatření na zařízení, které musí být v ostatních ohledech již plně funkční.

Struktura dokumentu

První kapitola obsahuje především vylíčení dosažených výsledků a stejně tak problémy, které se mi nepodařilo realizovat a důvody, které k tomu vedly.

V druhé kapitole uvádím popis konkrétního zařízení, jeho funkci a konkrétní rozměry. Dále jsem se zde snažil popsat k čemu by toto zařízení mělo sloužit a důvody, které vedou k tomuto technickému řešení.

Třetí kapitola obsahuje popis a výpočet úhlových tolerancí v obou osách. Znalost těchto tolerancí využívám jednak pro orientaci při návrhu polohovacího algoritmu a dále pak pro návrh řídicího algoritmu.

V další, čtvrté kapitole, je uveden vlastní návrh polohovacího algoritmu a jeho implementace. Součástí je také popis některých fyzikálních veličin, ze kterých algoritmus vychází, a odvození výpočtů, použitých v algoritmu. Dále jsou zde uvedena některá pozorování a závěry, které jsem využil pro návrh řízení.

V páté kapitole se zabývám systémovým návrhem, tedy kompletním návrhem řízení a všemi nezbytnými součástmi, které bude potřeba provést pro správnou funkčnost zařízení v jeho finální verzi.

Kapitola šest obsahuje návrh a realizaci snímání úhlové polohy v horizontální ose. Jsou zde uvedena dvě technická řešení, kterými jsem se zabýval, a jedna realizace. Uvádím zde důvody, které mě k této realizaci vedly a problémy, se kterými jsem se potýkal.

Sedmá kapitola obsahuje návrh a realizaci řízení motoru. Dále pak popis a funkční princip použitého výkonového obvodu.

V osmé kapitole se zabývám konkrétním návrhem a realizací řídicí a polohovací jednotky. Je zde uveden stručný popis použitých obvodů a jejich zapojení a osvětlení jejich funkcí.

Kapitola devět obsahuje návrh řídicího algoritmu. Je zde uvedeno jeho vývojové schéma. Vysvětluji zde jakým způsobem algoritmu funguje a uvádím souhrn jeho implementovaných částí.

Desátá kapitola je pak věnována měřením a testováním polohového senzoru.

Kapitola 2

Technický popis systému a problému

2.1 Stručný přehled současných zařízení využívajících tepelné záření pro účely ohřevu

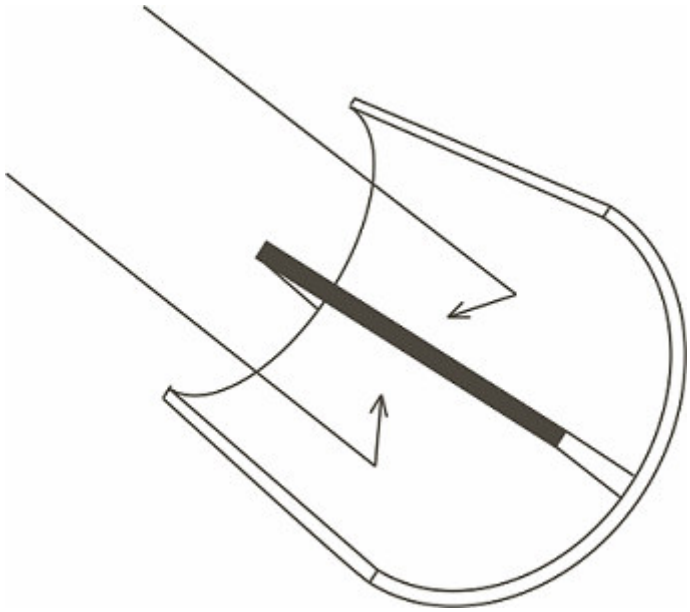
Využití solární energie, ať už ve formě fotovoltaických, nebo tepelných systémů, patří k ekologickému a perspektivnímu způsobu získávání energie. V současné době existuje celá řada deskových kolektorů, které využívají tepelného záření pro ohřev vody, vytápění, apod. . Nejběžněji je možno vidět tato zařízení nainstalována fixním způsobem, tedy s orientací směrem na jih a s pevným úhlem vhledem k vodorovné rovině. Tato poloha zajišťuje, že na plochu panelu dopadají sluneční paprsky během dne co nejdéle. Výhodou těchto kolektorů je, že dokáží využít i difuzního tepelného záření při polojasné obloze. Objevují se však již i jednoosé polohovací zařízení pro tepelné, či spíše pro fotovoltaické kolektory, které zvyšují nemalým způsobem účinnost těchto zařízení. Přesnost polohování u těchto zařízení není, a ani z principu nemusí být příliš vysoká.

Dalším způsobem získávání sluneční energie je využití fokusačních zařízení. Ty jsou v současné době využívány ve formě solárních pecí, či různých experimentálních zařízení. U nich je požadavek na přesnost polohování daleko vyšší.

2.2 Popis prototypového zařízení

Obrázek (2.1) : Fokusační systém na ohřev vody, pohled zboku a zezadu

Systém se skládá ze zrcadla, které má tvar parabolického žlabu. Zrcadlo odráží rovnoběžné sluneční paprsky v celém příčném profilu do společného ohniska. V ohnisku je umístěn trubkový sběrač (kolektor), který absorbuje energii tepelného pole do oběhové kapaliny, která je po ohřátí na optimální teplotu vedena do topného systému.



Obrázek (2.2) : Schématické znázornění parabolického zrcadla, silně je vyznačen kolektor, paprsky, které přicházejí kolmo k rovině zrcadla se do něj odrážejí

Rozměry zrcadla jsou 1,8 m na výšku 1,00 m na šířku a ohnisková vzdálenost je 0,45 m. Průměr kolektoru je 0,022 m. Vnější průměr kolektoru je optimalizován na co nejmenší z hlediska ochlazování a maximálního průtoku kapaliny. Velikost zrcadla je volena z hlediska celkového maximálního výkonu zařízení (na jeden metr čtvereční dopadá při jasném počasí přibližně 1 kW, v tomto případě dopadá na rovinu zrcadla výkon 1,80*1,00 = 1,80kW) a z hlediska jisté praktičnosti celého řešení (nároky na manipulační techniku pro větší a těžší zrcadla jsou podstatně vyšší a spadají spíše do průmyslové sféry). Celé zrcadlo je pak zavěšeno na kovovém rámu, který umožňuje nezávislý pohyb zrcadla ve dvou osách. V horizontální ose s mezním úhlem 0° – 270° a ve vertikální ose v úhlu 0° – 80°. Tyto meze zajišťují použití téměř ve všech zeměpisných šířkách, viz kapitola 4.

Z řady praktických důvodů byl pro napájení celého systému zvolen 12V zdroj (použít lze například automobilovou baterii), na který se nevztahují mezinárodní předpisy v oblasti síťového napětí, a to zejména z důvodů krytí do prostředí IP se stříkající vodou.

Pro pohyb ve vertikální ose je použit stejnosměrný motor, který přes převodovku pohybuje otočným rámem. Pro horizontální pohyb je použit šroubový motor, jenž na principu páky otáčí zrcadlem ve směru vertikálním.

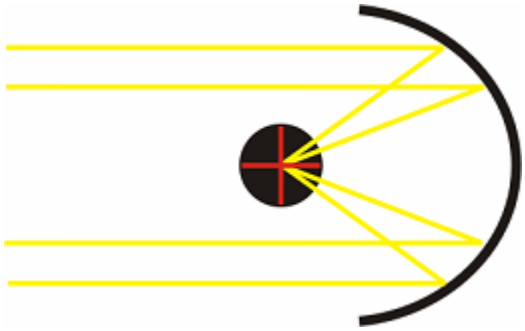
Výhodou tohoto způsobu získávání sluneční energie je, že zařízení dokáže využít tepelných špiček okamžitě a maximálně v průběhu celého dne a v případě přehřívání systému se může odklonit.

Kapitola 3

Analýza přesnosti polohování

3.1 Tolerance odchýlení zrcadla od nominální hodnoty

Při správném natočení zrcadla třeba garantovat, aby veškeré odražené paprsky dopadaly na kolektor. Pokud by tak tomu nebylo, docházelo by ke zbytečným ztrátám na výkonu zařízení a dokonce by mohlo dojít k poškození zařízení nebo ublížení na zdraví.



Obrázek (3.1) : Profil zrcadla s kolektorem a ideální odraz slunečního paprsku

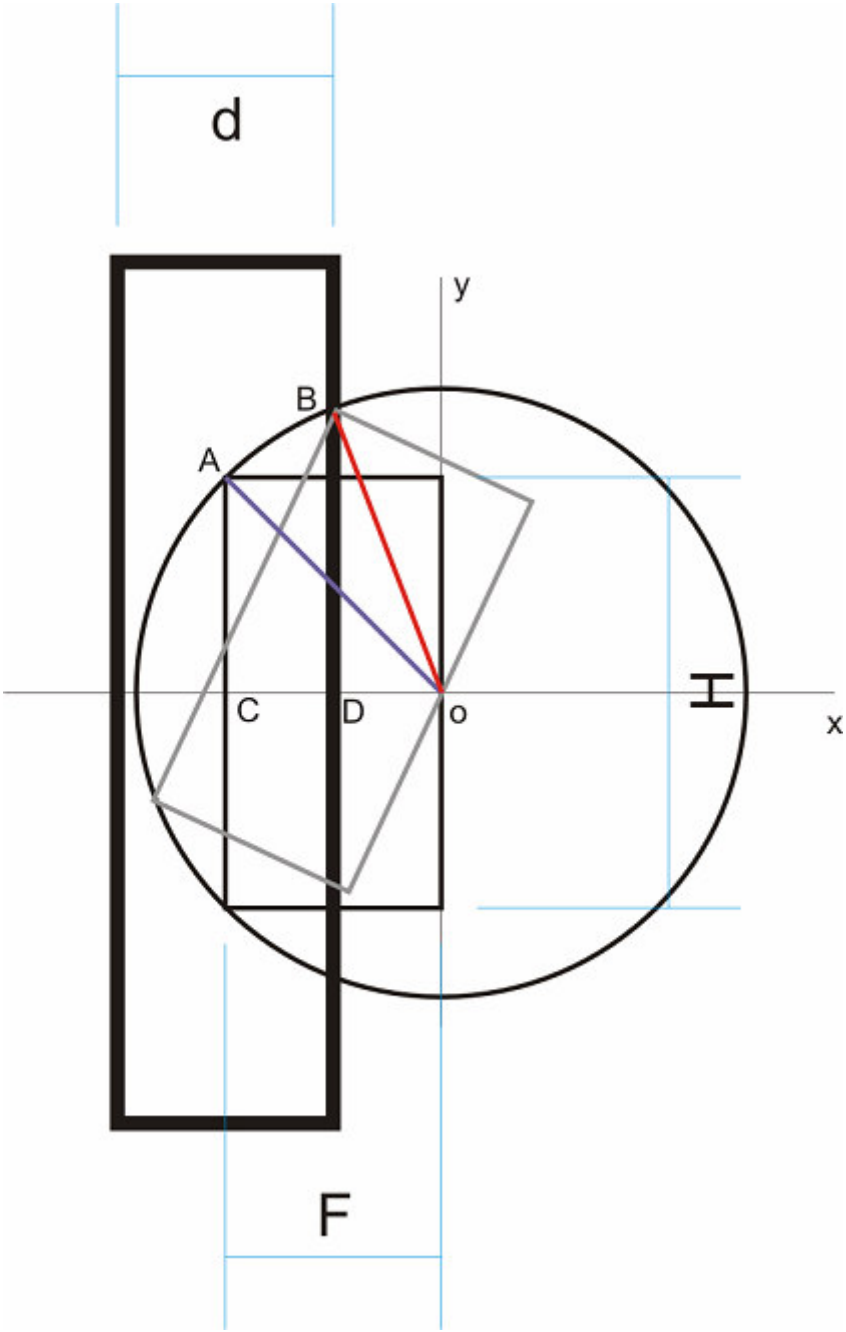
Také pro návrh řídicího algoritmu na výpočet Slunce je dobré si předem stanovit jisté hranice, které by pak algoritmus respektoval. Tyto hranice vyplývají z geometrických rozměrů zrcadla a kolektoru. Vycházím z předpokladu, že paprsky dopadající kolmo na rovinu zrcadla jsou odráženy do středu kolektoru v celé jeho délce, jak je schématicky znázorněno na obrázku (3.1).

Pokud zrcadlo není nasměrované přesně do středu Slunce, nastává vlastně situace, že paprsky dopadají na rovinu zrcadla úhlem různým od devadesáti stupňů a nejsou tedy odráženy přesně do středu ohniska. Ohnisko se vlastně posouvá v závislosti na úhlu dopadu, ale kolektor zůstává na svém původním místě.

Pokud tedy uvažujeme, že ve stavu, kdy je zrcadlo nasměrováno přesně, dopadají paprsky do středu kolektoru, lze připustit úhlovou odchylku od nominální hodnoty v takové míře, kdy se odražené paprsky kříží ještě na povrchu kolektoru. Takto nedochází k tepelným ztrátám. Můžeme připustit, že by teoreticky mohla nastat situace, kdy by se díky špatnému zaměření ohnisko posunulo před kolektor, ale i tak by veškeré odražené paprsky stejně dopadaly na kolektor. Je však vždy lepší nechat si při návrhu jisté rezervy.

Zrcadlo je na rámu umístěno tak, že při pohledu shora vidíme jeho parabolické zakřivení a při pohledu zboku se jeví jako obdélník. V polovině výšky se nachází osa otáčení ve vertikálním směru a v polovině šířky je osa otáčení v horizontálním směru. Je jisté, že tolerance v obou osách budou odlišné, jak to vyplývá z geometrického rozložení zrcadla.

3.2 Výpočet tolerance pro vertikální osu



Obrázek (3.2): Geometrické řešení výpočtu tolerance zrcadla ve vertikální ose

Na obrázku (3.2) je problém znázorněn geometricky. Osa otáčení je umístěna do středu souřadné soustavy v bodu *o*. Přímka, která prochází bodem *o* a je rovnoběžná s osou *y*, představuje zrcadlo v bočním řezu. Přímka, která prochází bodem *C* a je rovnoběžná s osou *y* představuje ohnisko zrcadla v podélném profilu. To je zobrazeno pomocí černého obdélníka vepsaného v kružnici. Výška zrcadla je H, ohnisková vzdálenost je F a poloměr kolektoru je d. Tlustě vytažený obdélník představuje profil kolektoru. Rozměry nejsou v měřítku, a to hlavně rozměr kolektoru, z důvodu lepší názornosti. Zkosený dopad paprsků je znázorněn pomocí šedého obdélníku. Pro různé úhly dopadu rohy obdélníku opisují kružnici, jak je

vyznačeno na obrázku (3.2). Mezní poloha, popsaná v předchozím odstavci 3.1 je ta, kdy se horní roh obdélníku pootočí z bodu A do bodu B. Nejdříve je třeba určit úhel BOC. To provedeme tak, že určíme průsečík přímky pocházející bodem D o rovnoběžné s osou y a kružnice se středem v počátku a poloměrem R.
Souřadnice bodu D jsou : $[-F + d/2, 0]$
Přímka, která Bodem D prochází má rovnici :

$$x = D \tag{3.2.1}$$

Poloměr kružnice, která vznikne otočení obdélníku o 360° lze vyjádřit jako :

$$R = \sqrt{F^2 + (H/2)^2} \tag{3.2.2}$$

Rovnice kružnice v takto definované souřadné soustavě je :

$$x^2 + y^2 = R^2 \tag{3.2.3}$$

Po dosazení rovnice přímky do rovnice kružnice dostaneme souřadnice bodu B :

$$B = [-F + d/2, \sqrt{\sqrt{F^2 + (H/2)^2} - (-F + d/2)^2}\,] \tag{3.2.4}$$

Úhel BOC je roven :

$$BOC = \tan^{-1}\left(\frac{|-F + d/2|}{|\sqrt{\sqrt{F^2 + (H/2)^2} - (-F + d/2)^2}|}\right) \tag{3.2.5}$$

Úhel AOC je roven :

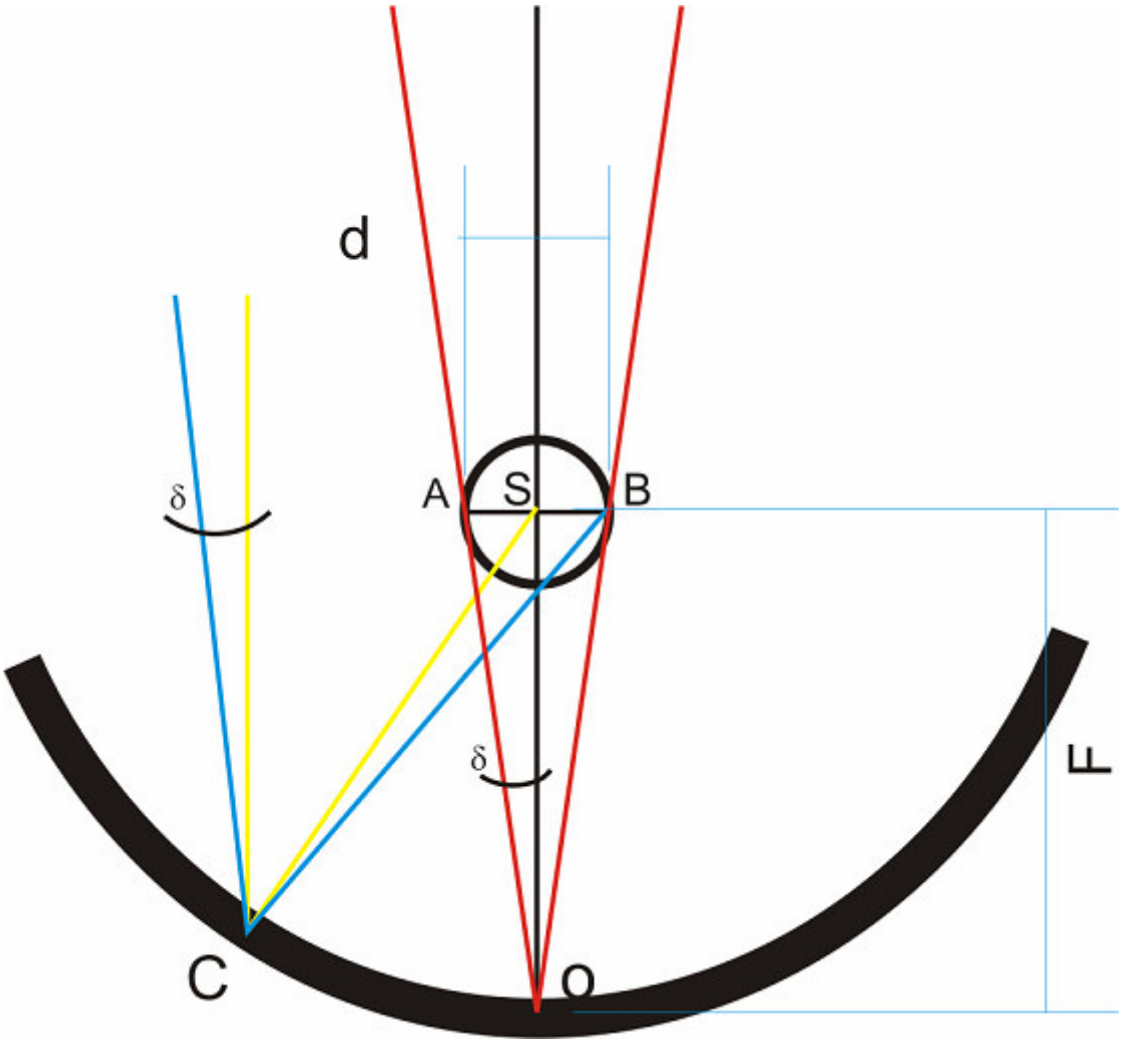
$$\mathrm{AOC} = \tan^{-1}\left(\frac{|-F|}{\sqrt{\sqrt{F^2 + (H/2)^2} - (-F + d/2)^2}|}\right) \tag{3.2.6}$$

Rozdíl obou úhlů AOC a BOC nám dává hledanou toleranci natočení pro vertikální osu. Po dosazení parametrů zrcadla F = 0,45m, H = 1,80m a d = 0,022m dostaneme hledanou toleranci δ_y.

$$\partial_y = AOC - BOC = 1.81° \tag{3.2.7}$$

3.3 Výpočet tolerance pro horizontální osu

Na obrázku (3.3) je žlutě vyznačen paprsek, který přichází kolmo k rovině zrcadla, dopadá na bod C a je odrážen přímo do středu kolektoru. V tom samém bodě C dopadá modrý



Obrázek (3.3): Geometrické řešení výpočtu tolerance zrcadla v horizontální ose

paprsek, který je ale vychýlen vůči rovině zrcadla o úhel δ. Tento úhel představuje mezní úhel, při kterém se ohnisko ještě neposune z povrchu kolektoru. Červeně je vyznačen paprsek, který dopadá také v mezním úhlu, ale dopadá do středu otáčení, vyznačeném zde bodem O. Mezní úhel δ, jak vyplývá z obrázku, tedy úhel SOB je dán vztahem :

$$\partial_x = \tan^{-1}\left(\frac{d}{2F}\right) \qquad (3.3.1)$$

Po dosazení parametrů zrcadla d = 0,022m a F = 0,45m dostaneme výsledek :

$$\partial_x = 1.40° .$$

3.4 Shrnutí výsledků

Tolerance natočení zrcadla ve vertikální ose, tedy v ose která bude kontrolovat azimut viz kapitola 4, je $\pm\partial_y = 1.8°$ .V horizontální ose, která bude kontrolovat výškovou souřadnici je tolerance stanovená $\pm\partial_x = 1.4°$. Pomocí těchto hodnot bude nyní možné určit, jak přesně bude muset polohovací algoritmus provádět výpočet polohy Slunce. Zároveň tyto hodnoty poslouží pro určení časového kroku polohování.

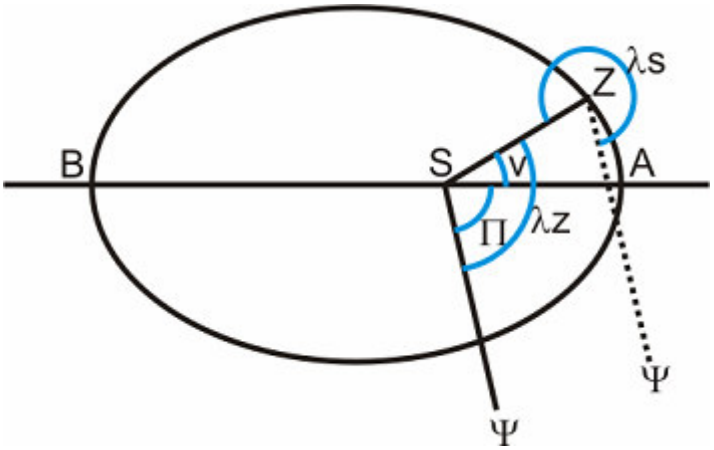
Kapitola 4

Výpočet polohy Slunce

4.1 Výpočet slunečních souřadnic

Pohyb Slunce po obloze je jen zdánlivý. Vzniká pohybem Země kolem Slunce a zemskou rotací. Země spolu s ostatními planetami obíhají okolo Slunce po eliptických drahách v takzvané rovině ekliptiky, přičemž Slunce se nachází v ohnisku této elipsy.
Země dále vykonává rotační pohyb kolem vlastní osy. Tato osa je vůči rovině ekliptiky natočena přibližně o 23,5^{o}. Rotační osa je umístěna v pólech Země. Pomocí pólů a rovníku je definovaná rovina světového rovníku. Průsečík mezi rovinou ekliptiky a světového rovníku se nazývá Jarní bod, jarní rovnodennost, a nastává vždy 21.března. Úhel, který na obrázku (4.1) svírá velká poloosa SA se směrem na jarní bod Ψ se nazývá délka perihelu Π. Délka perihelu se v důsledku precese, což je pohyb zemské osy přibližně po plášti rotačního kužele, mění a otočí se o 360° jednou za 26 000 tisíc let. Zmíněný precesní rovnoměrný pohyb vykonává vlastně jen myšlený střední světový pól, skutečný pól se kolem středního otáčí po malé elipse. Tento pohyb, jenž se opakuje s periodou přibližně osmnácti let, se nazývá nutací. Výsledkem nutačního pohybu je, že se v čase periodicky mění svou polohu jarní bod. Z toho vyplývá se v čase mění astronomická délka. Zároveň ani sklon zemské osy není konstantní ročně se zmenšuje o 0,47”. Kromě nutace a precese podléhá zemská osa ještě několika krátkodobým rotačním pohybům z nichž nejhlavnější jsou sluneční a měsíční nutace. Z těchto faktů plyne, že pro přesný výpočet neexistuje nějaký jednoduchý geometrický model, neboť se vše v čase neustále mění.

Pro výpočet souřadnic Sluce pro dané místo na Zemi je třeba započít výpočtem polohy Země v ekliptikálních souřadnicích. V ekliptikálních souřadnicích se poloha hvězdy určuje pomocí astronomické šířky, což je úhel směru na hvězdu vůči rovině ekliptiky a astronomické délky, která je definovaná jako úhel v rovině ekliptiky mezi polohou hvězdy a jarním bodem.



Obrázek 4.1 : Rovina ekliptiky

Pokud chceme určit souřadnice Země v ekliptikálních souřadnicích je jasné, že astronomická šířka je rovna nule, neboť Země leží v rovině ekliptiky a zajímáme se pouze o astronomickou délku. Úhel, který je v obrázku 4.1 označen jako v se nazývá pravá anomálie a je to úhel pod jakým úhlem se v daném časovém okamžiku nachází Země vhledem k hlavní ose elipsy. Astronomická délka Země je tedy v určitém časovém okamžiku rovna :

$$\lambda_z = \Pi + v \,. \tag{4.1.1}$$

Nás ale zajímá domnělý pohyb Slunce okolo Země. Ten se projevuje tak, že Slunce se pohybuje po ekliptice kladným směrem od západu k východu. Délka perihelu se ze Země bude jevit jako :

$$\Pi = \pi + 180^o \,. \tag{4.1.2}$$

Astronomická délka Slunce je potom :

$$\lambda_s = \Pi + v + 180^o \tag{4.1.3}$$

Nyní je tedy potřeba určit pravou anomálii Země v .To můžeme provést tak, že vypočteme rozdíl pravé a střední anomálie. Tento rozdíl se nazývá středová rovnice σ :

$$\sigma = v - M \tag{4.1.4}$$

Při řešení středové rovnice můžeme vyjít z Keplerových zákonů. Nejprve využijeme druhý zákon, který praví že, plocha sektoru elipsy, opsaná průvodičem planety, je úměrná času. Matematicky lze toto formulovat jako :

$$\frac{\pi ab}{P}(t - t_0) = \frac{1}{2}\int_0^v r^2 dv \tag{4.1.5}$$

kde **a** a **b** značí velkou a malou poloosu elipsy, **P** je celková doba oběhu, časový údaj t_0 odpovídá průchodu planety přísluním, t odpovídá obecnému času, v je pravá anomálie a r je průvodič. První Keplerův zákon praví, že planety se pohybují v elipsách, jejichž společným ohniskem je Slunce. Matematicky se to dá formulovat tak, že poloha planety musí odpovídat polární rovnici elipsy :

$$r = \frac{a(1-e^2)}{1 + e.\cos(v)} \tag{4.1.6}$$

Význam koeficientů je stejný jako u vztahu (4.1.5) , e značí excentricitu elipsy. Po dosazení za průvodič vyjádřený v (4.1.6) do 2. Keplerova zákona daného (4.1.5) dostaneme vztah :

$$M = (1-e^2)^{3/2}\int_0^v (1 + e\cos(v))^{-2} dv \tag{4.1.7}$$

kde M značí střední anomálii a získáme ho substitucí :

$$M = \frac{2\pi}{P}(t - t_0) \qquad (4.1.8)$$

Pokud ve vztahu (4.1.5) zanedbáme členy 3. řádu a vyšší dostaneme po integraci řadu :

$$M = v - 2e\sin(v) + \frac{3}{4}e^2\sin(2v) + ... \qquad (4.1.9)$$

Pro středovou rovnici (4.1.4) pak při použití vztahu (4.1.8) dostaneme konvergující řadu ve tvaru :

$$\sigma = v - M = 2e\sin(v) - \frac{3}{4}e^2\sin(2v) + ... \qquad (4.1.10)$$

Pokud ve výrazu (4.1.10) nahradíme pravou anomálii anomálií střední můžeme jej formálně vyjádřit jako řadu :

$$\sigma = 2e\sin(M) + \frac{5}{4}e^2\sin(2M) + ... \qquad (4.1.11)$$

Bližší odvození lze nalézt například v [1]. Pokud dosadíme příslušné parametry elipsy Zemské oběžné dráhy dostaneme pro výpočet středové rovnice výraz :

$$\sigma = 1{,}9148.\sin(M) + 0{,}02.\sin(2.M) + 0{,}0003.\sin(3.M) \qquad (4.1.12)$$

Nyní ještě zbývá určit hodnotu střední anomálie M. Každá elipsa se dá považovat za pravoúhlý průmět kružnice, jejíž rovina je od zobrazovací roviny odchýlena o určitý úhel. Význam M lze chápat tak, že souvisí s pravou anomálií způsobem, že je v závislosti na čase vyjádřena v pootočené kružnici. Středová rovnice tak vlastně vyjadřuje opravu střední polohy planety kruhového pohybu na její skutečnou polohu v elipse. Při zavedení veličiny střední denní úhlový pohyb n lze střední anomálii vyjádřit jako :

$$M = n(t - t_0) \qquad (4.1.13)$$

Kde t je obecný časový údaj a t_0 odpovídá průchodu planety přísluním, v obrázku (4.1) vyjádřeném bodem A.

Všechny výpočty astronomických těles se provádějí vždy v závislosti na nejbližší časové epoše. Pro každou tuto epochu lze v astronomických ročenkách nalézt hodnoty veličin a parametrů, které se s časem pomalu mění, např. střední denní úhlový pohyb nebo délku perihelu, apod. Tím je dosaženo toho, že je po určitou časovou etapu definovaná daná přesnost výpočtu.

Ve většině astronomických výpočtů se používá úplné Juliánské datum JD, což je časový údaj počítaný od doby zavedení kalendáře Juliem Césarem. Přepočet na Juliánské datum je dán vztahem :

$$JD = 367.Y - \mathrm{int}\{7.[Y + \mathrm{int}((M + 9)/12)]/4\} + \mathrm{int}(275.M/9) + D + 1721013{,}5 - 0{,}5.\mathrm{signum}(100.Y + M - 190002{,}5) + 0{,}5 \qquad (4.1.14)$$

Kde písmena Y, M, D vyjadřují rok, měsíc a celou část dne, vyjádřenou v našem Gregoriánském kalendáři a int značí převod čísla na jeho celou část. Střední anomálie Země, v závislosti na standardní epoše 2000, se tedy určí jako :

$$M0 = 357.5291;$$
$$M1 = 0.98560028;$$
$$M = M0 + M1.(JD-2451545) \qquad (4.1.15)$$

Pravá anomálie Země je potom dána :

$$v = M + \sigma . \qquad (4.1.16)$$

Hledanou astronomickou délku λ_s potom určíme pomocí vztahu (4.1.3).

Pokud nyní známe ekliptikální souřadnice Slunce je třeba provést transformaci do souřadnic rovníkových. Ty jsou dány deklinací, která je definovaná jako úhel mezi přímkou směřující na hvězdu a světovou osou a rektascencí, která vyjadřuje úhel mezi jarním bodem a průmětem přímky na hvězdu do roviny světového rovníku.

Transformace z ekliptikálních do rovníkových souřadnic vychází z prostorové geometrie. Přesné odvození lze nalézt v [1] nebo [4]. Jisté zjednodušení těchto vztahů přináší použití taylorova rozvoje. Při dosazení všech potřebných parametrů pro standardní epochu 2000 lze Ra a De vypočítat v závislosti na λ_s. Rektascenci RA určíme pomocí vztahu :

$$A2 = -2.4680;$$
$$A4 = 0.0530;$$
$$A6 = - 0.0014;$$
$$RA = \lambda_s + A2.\sin(2.\lambda_s) + A4.\sin(4.\lambda_s) + A6.\sin(6.\lambda_s) \qquad (4.1.17)$$

A deklinaci DE určíme pomocí :

$$D1 = 22.8008;$$
$$D3 = 0.5999;$$
$$D5 = 0.0493;$$
$$DE = D1.\sin(\lambda_s)$$
$$+ D3.\sin(\lambda_s).D3.\sin(\lambda_s).3.in(\lambda_s)$$
$$+ 5.\sin(\lambda_s).D5.\sin(\lambda_s).D5.\sin(\lambda_s).D5.\sin(\lambda_s).D5.\sin(\lambda_s) \qquad (4.1.18)$$

Oba vztahy, respektive jejich koeficienty, jsou upraveny vzhledem ke standardní epoše 2000.

Nyní je třeba provést poslední transformaci do obzorníkových souřadnic. Tyto souřadnice jsou již závislé na zeměpisné šířce a zeměpisné délce místa pozorovatele a jsou určené azimutem A, který je definován jako úhel mezi jihem a daným objektem a výškou h, která značí úhel mezi objektem a vodorovnou plochou.

Pro výpočet souřadnic Slunce zde vstupuje do algoritmu časový údaj. K výpočtu se používá hvězdný čas, neboť občanský čas se vzhledem k jevů popsaným v předchozích odstavcích s astronomickým časem stále rozchází. Hvězdný čas je definován jako hodinový

úhel jarního bodu. A protože, jak bylo řečeno, se jarní bod neustále posouvá musí být opraven o vlivy precese a nutace. Při transformaci lokálního času na hvězdný nejdříve provedeme transformaci na univerzální čas UT podle používaného časového pásma :

$$UT = LT + \text{časové pásmo} \qquad (4.1.19)$$

Kde LT je používaný lokální čas. V případě letního času musíme časový údaj posunout ještě o hodinu zpět.

Pomocí juliánského data, daného vztahem (4.1.14) vypočteme hvězdný čas S0 pro 0h UT pomocí rozvoje :

$$T = (JD_0 - 2451545)/36525$$
$$S_0 = 6.697374558 + 2400.05133691.T + 2{,}58622.\ 10^{-5}.T3.2{,}58622.\ 10^{-5}.T - 1{,}7.\ 10^{-10}.T.1{,}7.\ 10^{-9}.T.1{,}7.\ 10^{-9}.T \qquad (4.1.20)$$

kde JD_0 je juliánské datum v 0h UT. Nyní můžeme určil lokální hvězdný čas s, který se váže k místu pozorovatele zeměpisnou délkou :

$$s = S_0 + 1{,}0027379093.(\ H + (Min/60) + (Sec/3600)\ ) + délka/15 \qquad (4.1.21)$$

Transformaci z rovníkových souřadnic do obzorníkových provedeme pomocí následujících vztahů :

$$t = 360.(s - RA)/24 \qquad (4.1.22)$$

$$h = 90 - acos(\ sin(DE).sin(šířka) + cos(DE).cos(šířka).cos(t)\ ) \qquad (4.1.23)$$

$$A = atan(sin(t)/\ sin(šířka).cos(t) - tan(DE).cos(šířka)\ ) \qquad (4.1.24)$$

kde t je převod z hodinového úhlu na stupně, A je Azimut, h je výška, a šířka představuje zeměpisnou šířku.

4.2 Vliv atmosferické refrakce

Atmosférická refrakce je změna směru světelného paprsku, způsobená jeho lomem v zemské atmosféře. Při přechodu z řidšího prostředí do hustšího se paprsek řídí Snellovo zákonem lomu. Index lomu v tomto případě záleží na teplotě a tlaku atmosféry, které se mění jednak s nadmořskou výškou, anebo z důvodu atmosférických proudění a jiných jevů. Z toho jasně plyne že není možné přesně určit univerzální algoritmus, který by atmosférickou kompenzoval. V astronomii se astronomická refrakce většinou počítá interpolací z refrakční tabulky, která je sestavená v závislosti na zenitové vzdálenosti (zenitová vzdálenost je úhel měřený od nadhlavníku směrem k obzoru). Platí, že atmosférická rekrace je tím větší, čím je větší zenitová vzdálenost. Z refrakční tabulky uvedené v [1] vyplývá, že při zenitové

vzdálenosti hvězdy 70° činí hodnota refrakce zhruba 2'. Pro zenitové vzdálenosti větší než 80° nejsou již refrakční tabulky definované, neboť při této zenitové vzdálenosti se atmosferická refrakce nechová zcela jednoznačně. Dá se přepokládat, že bude tímto směrem růst, ale vzhledem k výsledkům z kapitoly 3 a vzhledem k faktu, že v této fázi dne je intenzita slunečního záření již velmi malá, se dá atmosferická refrakce zanedbat.

4.3 Implementace algoritmu v jazyce C

```
void ObzorS(int Y,int M,double D,double H,double Min,double Sec,
                        double sirka,double delka,int cPasmo,
                        double *h,double *A){

/************************ Transformace na UT ************************/
        int Letni = 0;
        H = H + cPasmo + Letni;
        if(H < 0)
        H = H + 24;

/************* juliánské datum pro daný den v  0hUT *************/

        double JD_0;
        JD_0 = 367*Y - (int)((7*(Y+(int)((M+9)/12)))/4) + (int)(275*M/9) + D
                + 1721013.5 - 0.5*sgn(100*Y + M - 190002.5) + 0.5;

/**************** hvězdný čas v 0:00 UT pro dané datum ***********/

        double T3,S_0;
        T3  = (JD_0 - 2451545)/36525;
        S_0 = 6.697374558 + 2400.05133691*T3
                + 0.0000258622*T3*0.0000258622*T3
                - 0.0000000017*T3*0.0000000017*T3*0.0000000017*T3;
        while( S_0 > 24)
                S_0 = S_0 - 24;

/**************************lokální hvězdný čas *****************/

        double s, cas;

        cas = H + (Min/60) + (Sec/3600);
        cas = cas*1.0027379093;
        s = S_0 + cas + delka/15;
        while (s > 24)
                s = s - 24;

/************************** Juliánské datum *******************/
        double JD;
```

```
        D = D + H/24+Min/1440+Sec/86400;
        JD = 367*Y - (int)((7*(Y+(int)((M+9)/12)))/4) + (int)(275*M/9) + D
              + 1721013.5 - 0.5*sgn(100*Y + M - 190002.5) + 0.5;



/******************* Střední anomálie Země **************/

        const double M0 = 357.5291;
        const double M1 = 0.98560028;
        double Mm;

        Mm = M0 + M1*(JD-2451545);

        while(Mm > 360)
                Mm = Mm - 360;
        while (Mm < 0)
                Mm = Mm + 360;


/********************** Skutečná anomálie Země******************/

        const double C1 = 1.9148;
        const double C2 = 0.0200;
        const double C3 = 0.0003;
        double C,nu;
        C = C1*sin(k*Mm) + C2*sin(k*2*Mm) + C3*sin(k*3*Mm);
        nu = Mm+C;

/*********************** Ekliptikalni souradnice ********************/

        const double PI = 102.9372;

        double lambda = nu + PI + 180;

        while (lambda > 360)
                lambda = lambda-360;

        while (lambda < 0)
                lambda = lambda+360;





/************************ Rovníkové souřadnice *********************/

        const double A2 = -2.4680;
        const double A4 = 0.0530;
```

```
        const double A6 = -0.0014;
        const double D1 = 22.8008;
        const double D3 = 0.5999;
        const double D5 = 0.0493;
        double RA, DE;

        RA = lambda + A2*sin(k*2*lambda) + A4*sin(k*4*lambda) + A6*sin(k*6*lambda);
        RA = RA*24/360;
        DE = D1*sin(k*lambda)
                + D3*sin(k*lambda)*D3*sin(k*lambda)*D3*sin(k*lambda)
                + D5*sin(k*lambda)*D5*sin(k*lambda)*D5*sin(k*lambda)
                * D5*sin(k*lambda)*D5*sin(k*lambda);


/******************* Obzorníkové souřadnice *******************/

        double v1,v2,t;
        t = 360*(s - RA)/24;
        *h =90 - (1/k)*acos(sin(k*DE)*sin(k*sirka)+cos(k*DE)*cos(k*sirka)*cos(k*t));



        v1 =  sin(k*t);
        v2 =  sin(k*sirka)*cos(k*t) - tan(k*DE)*cos(k*sirka);

        *A = (1/k)*atan(v1/v2);

        if(v1 > 0){
                if(v2 < 0){
                        *A = *A + 180;
                }
        }
        if (v1 < 0){
                if (v2 < 0){
                        *A = *A - 180;
        }
}
```

4.4 Dosažené výsledky a pozorování

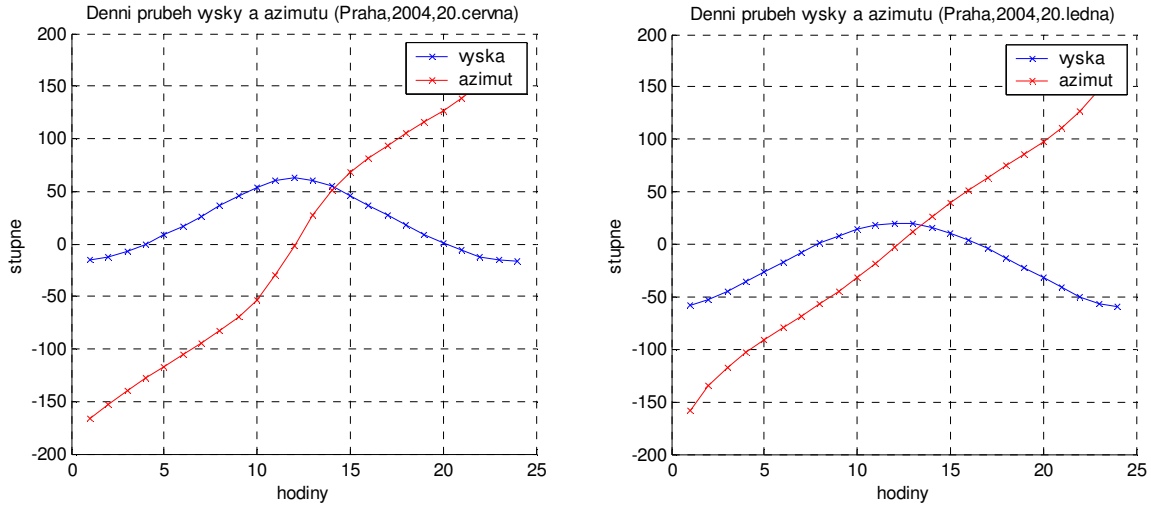
Celý zdrojový kód přikládám v elektronické příloze [12]. Přesnost výsledků algoritmu jsem porovnával s astronomickou ročenkou pro rok 2004 [3], v níž jsou uvedeny základní efemeridy Slunce. Výsledky porovnání jsou uvedeny v tabulce (4.1).

Datum a čas	h[°]	h_{ref}[°]	Δ_{abs}[‘]	A[°]	A_{ref}[°]	Δ_{abs}[‘]
1.1. 2004 0h 0m 0s	-54,53	-54,87	20,4	-130,01	-129,66	21
1.2. 2004 0h 0m 0s	-50,94	-51,09	9	-137,98	-137,88	6

1.3. 2004 0h 0m 0s	-42,06	-42,07	0,6	-143,48	-143,5	1,2
1.4. 2004 0h 0m 0s	-30,01	-30,01	0	-146,58	-146,6	1,2
1.5. 2004 0h 0m 0s	-19,66	-19,56	6	-148,84	-148,9	3,6
1.6. 2004 0h 0m 0s	-13,39	-13,06	19,8	-151,36	-151,05	18,6
1.7. 2004 0h 0m 0s	-12,91	-12,54	22,2	-153,04	-153,19	9
1.8. 2004 0h 0m 0s	-17,74	-17,58	9,6	-152,02	-152,1	4,8
1.9. 2004 0h 0m 0s	-26,25	-26,26	0,6	-147,01	-147,04	1,8
1.10. 2004 0h 0m 0s	-35,83	-35,85	1,2	-139,06	-139,08	1,2
1.11. 2004 0h 0m 0s	-45,02	-45,13	6,6	-130,53	-130,5	1,8
1.12. 2004 0h 0m 0s	-51,74	-52,04	18	-126,5	-126,24	15,6

Tabulka (4.1) : Porovnání výpočtů algoritmu s katalogovými hodnotami

Z výsledků je vidět, že maximální odchylka od tabelovaných hodnot nepřekračuje hodnotu 22 úhlových minut, což odpovídá hodnotě 0,36°. Z odchylek je také patrná jistá periodičnost, se kterou se oddalují a přibližují a k tabelovaným hodnotám. Tyto nepřesnosti jsou způsobené tím, že jsem ve výpočtu neuvažoval korekci na nutační a precesní jevy. Toto zjednodušení si mohu dovolit vzhledem k výsledkům kapitoly 3.

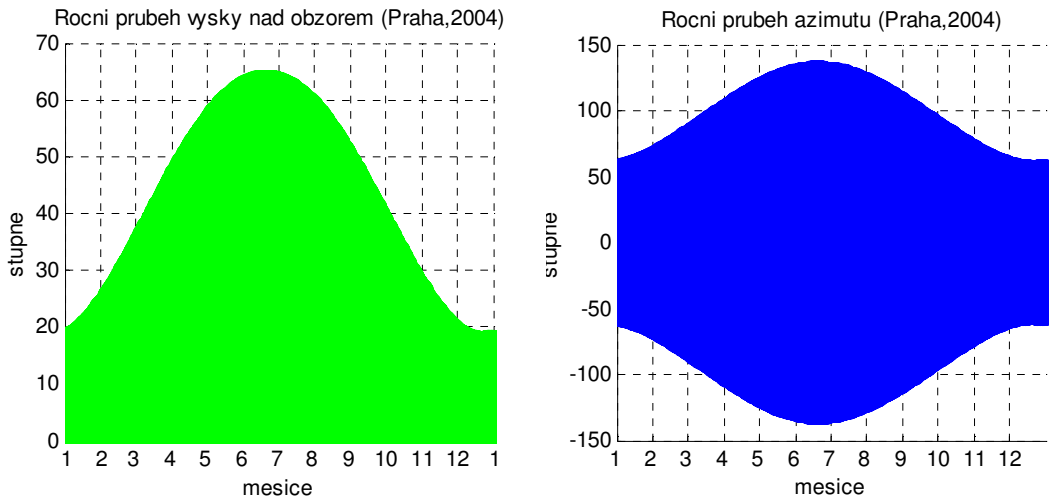


Obrázek (4.2) : Příklad denního průběhu výšky a azimutu v létě a v zimě

Algoritmus poté implementoval jako MEX-file [13] pro použití v Matlabu s cílem vygenerovat a vykreslit ročních průběhy výšky a azimutu. Vstupem je funkce je lokální čas, zeměpisné souřadnice pozorovatele a časové pásmo, výstupem jsou pak příslušné obzorníkové souřadnice Slunce, výška a azimut.

Z obrázků (4.2) je také patrné, že průběhy i extrémní hodnoty obou souřadnic se během roku podstatně liší. A navíc tyto průběhy se budou lišit také v závislosti na zeměpisné šířce. Pro návrh řídicí jednotky z toho tedy plyne, že bude buď potřeba používat algoritmus bez nějakých zjednodušení, z toho tedy vyplívají nároky na přesnost aritmetické jednotky a nebo bude potřeba veškerá data přepočítat předem a uložit a z toho zase plynou nároky na paměť řídicí jednotky.

Algoritmus jsem použil v programu napsaném jazyce C, který generuje roční průběh obou souřadnic s časovým krokem jedna minuta, viz. Obr. (4.3), a zároveň počítá maximum derivace (derivace je aproximovaná diferencí) průběhu obou souřadnic. Zdrojový kód je obsažen v CD příloze [14].



Obrázek (4.3) : Roční maxima výšky a azimutu

Pro účely řízení zrcadla je potřebná vždy jen ta část dat, kdy je výška větší než nula. Z grafu na obrázku (4.1) jenž, vlastně podstavuje dráhu Slunce uraženou během dne je patrné, že ani rychlost pohybu Slunce během dne ani roku není konstantní. Proto jsem provedl derivaci podle času ročního průběhu obou souřadnic. Tak jsem vlastně určil rychlost pohybu Slunce v závislosti na časovém období v roce. Poté jsem vyhledal maxima. Ta nastávají vždy 21. března v 6hod 11min pro výšku a 21. června ve 12hod 4 min pro Azimut. Velikost těchto maxim se mění v závislosti na zeměpisné šířce jak ukazuje tabulka (4.2). Tyto maxima udávají o kolik stupňů se Slunce během roku nejrychleji posune za jednu minutu. Z tohoto údaje je pak možné odvodit, s jakým časovým krokem má řídicí jednotka provádět polohování.

Zeměpisná šířka [stupně]	Max. A [stupně/min]	Max. H [stupně/min]
30	1,8	0,21
40	0,78	0,19
50	0,5	0,16
60	0,38	0,12
70	0,31	0,085
80	0,27	0,04
90	0,25	0,0003

Tabulka (4.2) : Maximální rychlost pohybu Slunce v závislosti na zeměpisné šířce

Kapitola 5

Systémový návrh řízení

Je třeba navrhnout řídicí systém, který bude provádět polohování a regulaci ve dvou nezávislých osách. Jde o systém se dvěma vstupy a se dvěma výstupy. Vstupem systému bude požadované natočení zrcadla v závislosti na poloze Slunce v daném časovém okamžiku a dané zeměpisné pozici, výstupem bude příslušné natočení obou motorů. Pro potřebu zpětnovazebního řízení je třeba snímat polohu natočení rámu v obou osách. Poté je potřeba provést identifikaci systému a návrh regulace. Cílem polohové regulace by mělo být přivedení systému co nejrychleji do požadované polohy. Jelikož bude systém bude vystaven vnějším vlivům jako, jsou například poryvy větru, je namístě analýza robustnosti systému a popřípadě návrh robustního regulátoru.

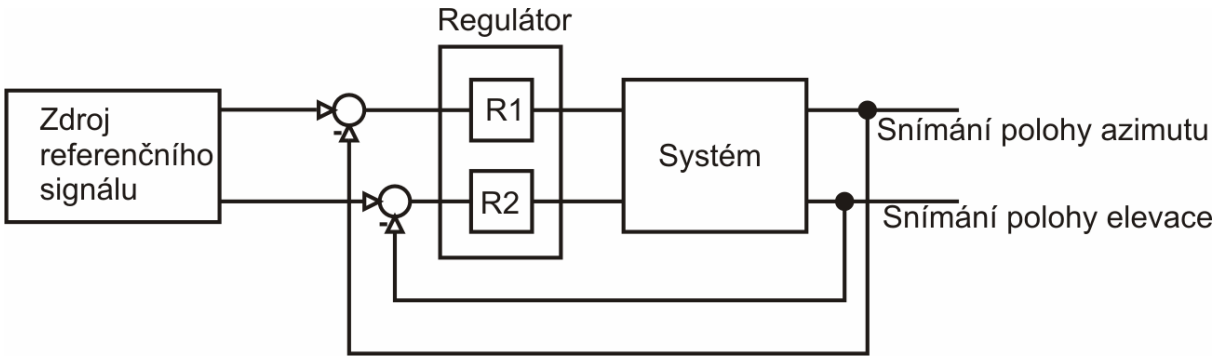
Polohování zrcadla bude prováděno kombinací dvou způsobů. Oba způsoby mají svoje výhody i nevýhody, avšak současné použití obou přístupů vede k potlačení nevýhodných vlastností.

První způsob polohování zrcadla je jeho „slepé“ nastavení na předpokládanou polohu Slunce. Pomocí výpočtu, vycházejícího z astronomických dat, je třeba určit polohu Slunce a zrcadlo podle ní nastavit. Je třeba vytvořit algoritmus, který by v závislosti na časové bázi, zeměpisné šířce a délce pozorovatele, určoval obzorníkové souřadnice slunečního kotouče v daných mezích přesnosti. Termín „slepé řízení“ je použit z důvodu, že zde chybí přímá zpětná vazba, pomocí níž bychom mohli ověřit, že je zrcadlo namířeno skutečně přesně do Slunce, i když je jeho poloha přesně spočítaná. Je třeba si totiž uvědomit, že při nepřesných počátečních podmínkách (zrcadlo nemusí být umístěno zcela vodorovně, nemusí být zcela správně zorientováno apod.) by se systém trvale odchyloval od skutečné pozice Slunce. Základní výhodou tohoto řešení je, že zrcadlo sleduje polohu Slunce i v případě, že je zakryto nějakým objektem, například oblačností, stromem a podobně. Zde bych rád podtrhl, že fokusační kolektory umožňují okamžité a maximální využití teplotních špiček a naopak neumožňují využití difuzního záření, jako kolektory deskové. Pro správnou funkci zařízení je tedy nanejvýš důležité, aby nedocházelo k časovým prodlevám při směrování na Slunce, jak by k tomu mohlo docházet případě, pokud bychom polohovali zrcadlo pouze pomocí vyhledávání maxima teplotního gradientu nebo intensity záření, což představuje jistou alternativu k tomuto řešení. Pokud bychom zrcadlo řídili pouze pomocí sluneční intenzity, tak by se v případě příchodu oblačnosti sledování Slunce ustalo a zrcadlo by setrvávalo ve stejné pozici, zatímco poloha Sluce by se mezitím měnila. Pokud by se skutečná poloha Slunce změnila hodně, docházelo by k časové prodlevě při natáčení zrcadla a tím i ztrátě na výkonu (současné technické řešení polohovacího rámu umožňuje jen velmi pomalý pohyb zrcadla) a navíc přepokládám, že vyhledávání Slunce pouze podle intensity jeho záření by představovalo nemalý technický problém.

Druhý způsob polohování by se dal nazvat „doostřování“ systému, nastaveného pomocí předpočtených dat. Bylo možné použít již zmiňované vyhledávání maxima teplotního gradientu. Systém by byl na dané pozici podle určitého systému rozhýbán v obou osách, přičemž by se v každém bodě měřila a zaznamenávala teplota. Výsledná pozice by pak byla určena tou největší teplotou v daném okolí vypočítané pozice. Výhodou je, že by nebylo nutné opatřovat zařízení žádným dalším čidlem, neboť teplotní senzor musí být každopádně na zařízení instalován z důvodu ochrany proti přehřívání oběhového média. V tomto konstrukčním řešení se však rám zrcadla nedokáže pohybovat příliš rychle, a tak by vyhledávání maxima teploty trvalo poměrně dlouhou dobu a docházelo by ke zbytečným ztrátám na výkonu. Dále by bylo třeba uvažovat nezanedbatelné časové konstanty při měření teploty a tedy další ztráty.

Jako daleko účinnější se zde jeví použití fotosenzitivního prvku, například čtyř-kvadrantové nebo sektorové diody, na kterou by byly světelné paprsky přiváděny pomocí primitivního objektivu. Ten by směřoval kolmo k rovině zrcadla na Slunce a jeho průměr by byl tak velký, aby pokryl daný rádius Slunce v pracovním bodě, určeném výpočtem. Podobný princip je používán při směrování hvězdářských teleskopů. Řídicí jednotka by pak v dané pozici zrcadlo nastavovala tak, aby ve všech kvadrantech diody byla stejná intensita záření. Toto řešení sice vyžaduje montáž dalšího senzoru, jenž se ve výsledku promítne do celkových nákladů, ale představuje účinné a poměrně jednoduché řešení problému „doostřování“ zrcadla.

Samostatné použití čtyř-kvadrantové diody při použití jednoduchého objektivu lze zcela vyloučit, neboť takto lze operovat pouze v poměrně omezené oblasti. Avšak toto řešení v kombinaci s počítáváním polohy představuje spolehlivý a přesný způsob, jak sledovat zrcadlem polohu Slunce.



Obrázek (9.1) schéma řídicího systému

Polohování systému bude tedy v praxi fungovat tak, že řídicí jednotka bude nastavovat v určitém časovém kroku zrcadlo na vypočítanou polohu Slunce. Mezi těmito kroky se bude k referenčnímu signálu přičítat hodnota získaná z „doostřovacího“ systému. Výsledný referenční signál bude přiveden na vstup regulovaného systému viz. obrázek (9.1).

Nezbytnou součástí návrhu řízení je polohová regulace. Obecně lze pro systém počítat s PID regulátorem, který může být realizován softwarově pomocí mikroprocesoru. Po identifikaci systému bude možné vyladit regulátor tak, aby se dosáhlo co nejrychlejší odezvy na změnu referenčního signálu, a aby byla odchylka od požadované polohy minimální.

Polohovací zařízení má však v nynějším technickém provedení velice omezený akční zásah, takže se dá očekávat, že regulací se jeho dynamické vlastnosti příliš nezlepší. Zejména, v případě robustního regulátoru nebude schopné korigovat rychlé poryvy větru. Celý projekt je však zatím ve fázi návrhu, jehož součástí musí být testování zařízení v provozních podmínkách, a zde se teprve ukáže jak jsou tyto úvahy opodstatněné. Dá se očekávat, že pokud by se nynější technické provedení jevilo jako nedostačující, budou následovat technické změny, které daný problém budou řešit, například použitím výkonnějšího motoru, atd.. Proto je myslím vhodné v návrhu již od začátku počítat s možností realizace složitějších regulačních metod.

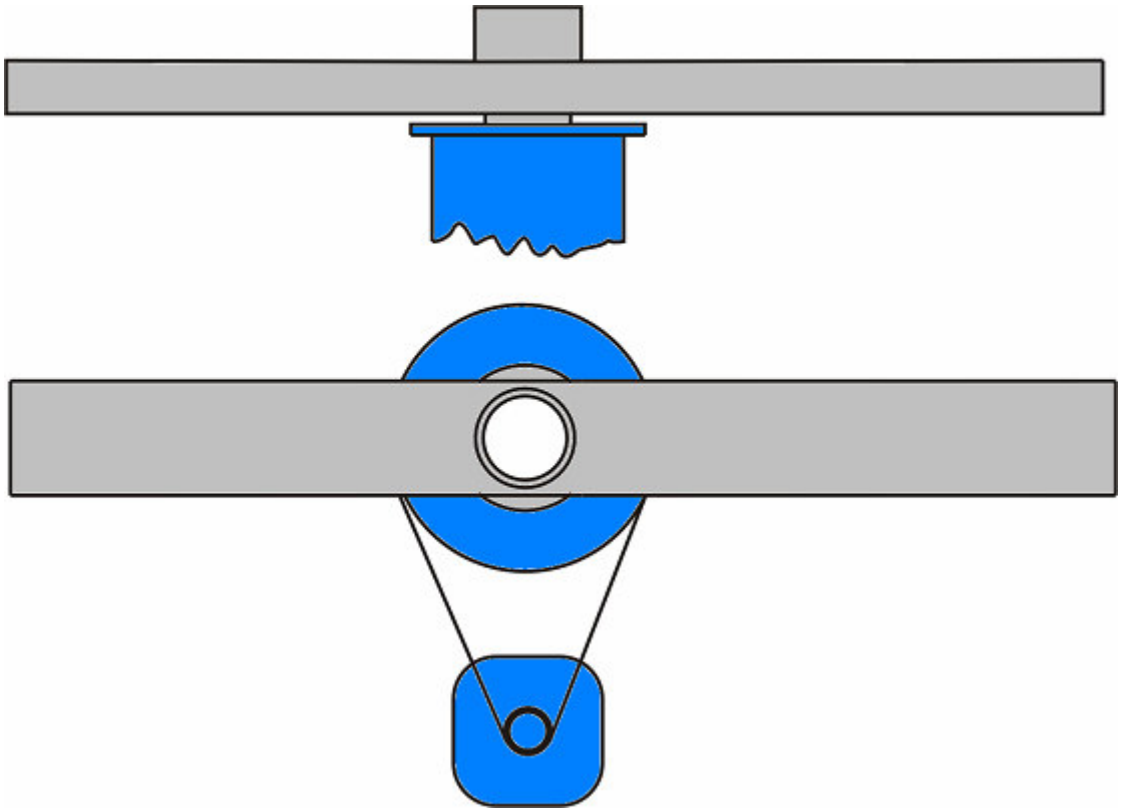
Výsledkem celého návrhu je zajištění maximálního výkonu systému v závislosti na co nejjednodušším a nejlevnějším technickým řešením a diskuze možnosti komerčního využití.

Kapitola 6

Návrh a realizace snímání polohy

6.1 Problémy

Při návrhu snímání úhlové polohy horizontálního pohybu zrcadla jsem se potýkal ze zásadním problémem, a to že při snímání horizontální úhlové polohy nelze čidlo umístit na osu otáčení. V ose otáčení je umístěna dutá trubka o vnitřním průměru zhruba 80.0 mm, kterou prochází přívod oběhového média ke kolektoru a kabeláž.



Obrázek (6.1) : Půdorys a nárys polohovacího rámu

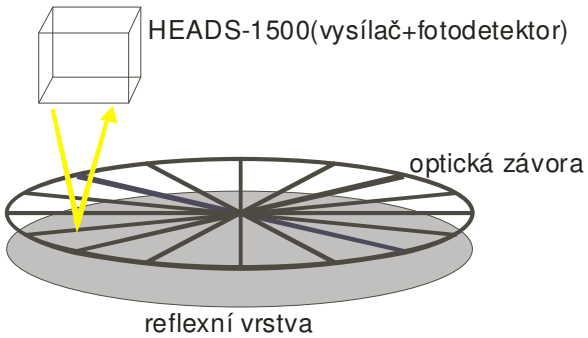
Jelikož problém návrhu a konstrukce zařízení pro sledování sluneční dráhy je velice podobný návrhům řídicích systémů pro hvězdářské teleskopy, hledal jsem původně inspiraci v této oblasti. Při snímaní úhlové polohy natočení dalekohledů se používají přesné potenciometry, diferenciální transformátory a kapacitory, nebo resolvery viz [5]. Tyto senzory jsou velice přesné, ale také drahé. Pro náš případ však nejdou z principu použít, neboť je potřeba připevnit je na osu otáčeni, která zde chybí. Snímání polohy lze v tomto případě vyřešit tak, že senzor připevníme na otočný rám, na obrázku (6.1) vyznačen šedě, a polohu budeme snímat z statické části, na obrázku (6.1) vyznačeno modře. Pro tento způsob snímání polohy

by teoreticky ještě připadal v úvahu induktosyn, jenž má pravítko ve tvaru kotouče a v [5] je také zmiňován, ale zakázková výroba takovéhoto senzoru s požadovanými rozměry a přesností by se jistě velmi prodražila.

Možnosti, které tedy přicházejí v úvahu jsou tedy buď použití klasického optického inkrementálního čidla s nutností výroby speciálního terčíku, který by svým tvarem respektoval konstrukci, anebo přijít s úplně jiným řešením.

6.2 Klasické inkrementální čidlo

Pro srovnání zde uvádím stručný návrh snímání polohy pomocí klasického inkrementálního čidla, neboť jsem se touto možností také zprvu zabýval. Uvažoval jsem čidlo HEADS 1500, viz [23], které je jedním z klasických reprezentantů čidel pracujícím na principu vysílání a přijímání světelného signálu přes optickou závoru. Výstupem čidla je potom úroveň high nebo low v závislosti na příjímání nebo nepřijímání světelného signálu. Čidlo HEADS-1500 obsahuje vysílač a přijímač v jednom pouzdře. Princip senzoru je znázorněn na obrázku (6.2). Světelný signál vlnové délky 655nm je vysílán a pomocí reflexní vrstvy navracen zpět do čidla. Na reflexní vrstvě je umístěna mřížka, která absorbuje světelný paprsek a zabrání tak čidlu jeho detekci.



Obrázek (6.2) : Schéma principu inkrementálního optického senzoru

Mřížku požadovaných rozměrů je možné zhotovit potiskem na průsvitku v grafickém studiu. Na spodní stranu folie by se poté připevnila vhodná reflexní vrstva. Přesnost takového potisku je 3 200 dpi, tedy 3200 bodů na inch. Rozlišitelnost čidla HEADS 1500 je 0.178 milimetrů.

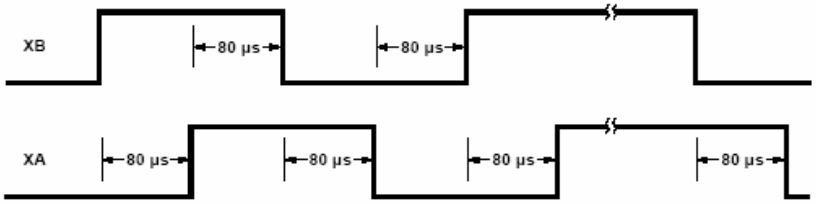
Pokud bychom při konstrukci polohového senzoru s obvodem HEADS-1500 požadovali nejmenší dílek o velikosti 0.1^{o}, znamená to, že pro celkových 360° by bylo třeba natisknout terčík s 3600 černými čarami, střídajícími se s průsvitnou částí. Pro jeden inkrement bychom v mezním případě potřebovali 2*0.178mm = 0.356mm (0.178mm pro absorpční a 0.178mm pro reflexní vrstvu). Pro celkový obvod potom vychází délka 3600*0.356 = 1281,6mm. Tomu odpovídá průměr terčíku 407.95mm. Pokud bychom při návrhu nešli až na hranice rozlišení čidla HEADS 1500, byl by průměr ještě větší. Terčík s takto velkým poloměrem by nebylo možné připevnit na statickou část rámu, jelikož by docházelo ke kolizi se zrcadlem, které rám nese, nebo s hybným zařízením. Nejvhodnější umístění terčíku a vlastně i senzoru je vzdálenost 65.0mm – 75.0mm od pomyslné osy

otáčení. To vyplývá z fyzických rozměrů rámu. Pokud bychom chtěli použít terčík respektující tyto rozměry, pak tedy s průměrem terčíku 150.0mm dostáváme podle předchozího výpočtu velkost nejmenšího dílku na 0.27^{o}. Je z celé řady důvodů výhodnější, aby tyto dílky tvořily celé násobky stupňů. Hranici 0.27^{o} lze již jenom zvětšovat, a nejbližší hodnota pro celý násobek stupně je 0.5^{o}. Domnívám se, že pro správný návrh řízení polohy je třeba, aby rozlišení čidla pohybovalo v hodnotách alespoň deset-krát citlivějších, než jsou tolerance natočení zrcadla, viz. kapitola 3. Tento požadavek lze opřít i o výslednou přesnost polohovacího algoritmu popsanou v kapitole 4. Z tohoto pohledu se snímání polohy pomocí inkrementálního optického čidla jeví jako nedostačující.

6.3 Optické čidlo HDNS-2000

Čidla řady HDNS od firmy Agilent Technologies jsou standardně používaná v optických myších, které nahradily předchozí myši kuličkové. Pro komunikaci s řídicím mikroprocesorem používají tyto čidla rozhraní PS/2. Čidlo HDNS-2000 umožňuje navíc použít dvoukanálový kvadraturní výstup, viz. Obr. (6.3) a je tedy možno získat informaci o poloze v příslušném směru přímo na výstupech integrovaného obvodu, nezabalenou do datového rámce.

Princip snímání je založen na optické technologii, která zaznamenává změny v poloze sekvenčním snímáním povrchu. Čidlo obsahuje miniaturní kameru, do které je pomocí čočky a vhodného osvětlení přiváděn obraz povrchu, po kterém se pohybuje. Obraz je dále zpracován DSP procesorem. Ten generuje informaci o velikosti posunu a jeho směru. Poté je informace o poloze vložena do datového rámce, anebo je přivedena na výstup v kvadraturní formě.



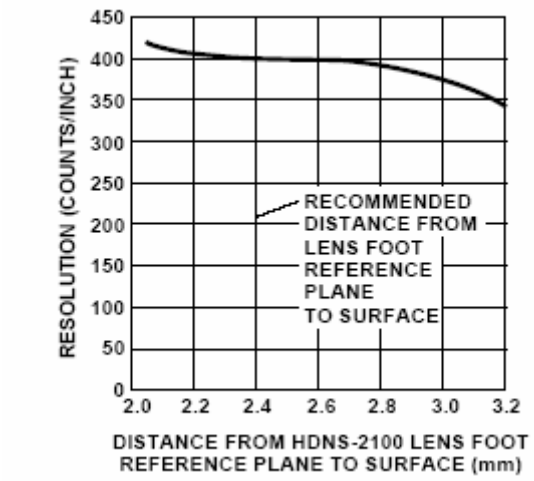
Obrázek 6.3 : Příklad kvadraturního signálu

Signál Xb nese informaci o relativní změně polohy a signál Xa, který je vždy o půl periody posunut dopředu nebo dozadu vůči Xb, o jejím směru. Stejný princip funguje i ve směru druhé osy. Po zpracování těchto signálů v mikroprocesoru získáme relativní změny polohy v rovině. Maximální frekvence kvadraturního signálu je 12.5 kHz. Od této frekvence se pak odvíjí maximální doporučená rychlost pohybu 30cm za sekundu a maximální zrychlení 0.15g. Při nerespektování těchto hodnot ztrácí čidlo svoje rozlišení, které výrobce udává jako 400cpi,

tedy 400 inkrementů na jeden inch, což odpovídá 16 inkrementům na jeden milimetr.

Z konstrukčního hlediska je výhodné, aby bylo čidlo umístěno v oblasti 65.0 mm – 85.0 mm od osy otáčení. Pro poloměr 150.0 mm vychází obvod kruhu na 471,24 mm. Při rozlišení 16 inkrementů na jeden milimetr je potom celkový počet inkrementů na 360 ^{o} při poloměru 150.0 mm roven 7068,58. Jednomu inkrementu potom odpovídá 0,05^{o} stupně, což představuje hodnotu nejmenšího dílku desetkrát větší, než lze ve stejných podmínkách dosáhnout pomocí optického inkrementální čidla.

Pro správnou funkci čidla je nutné dodržovat definovanou vzdálenost od povrchu, aby byl obraz povrchu přicházel do kamery nerozostřený. Při použití čočky HDNS-2100 je vzdálenost vyznačena v grafu na obrázku (6.4) .

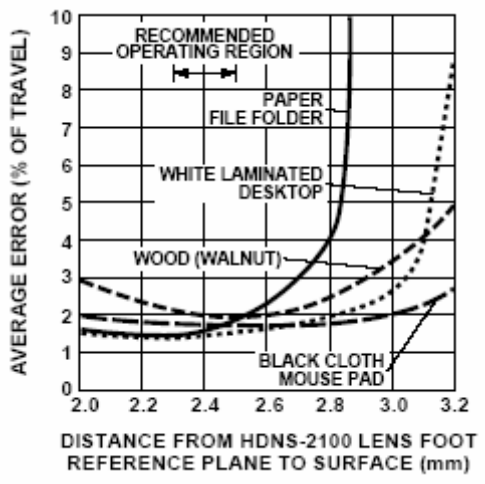


Obrázek (6.4) : Závislost rozlišení čidla HDNS-2000 na vzdálenosti od povrchu (graf je vyňat z dokumentace k obvodu HDNS-2000)

Při nesprávném zaostření čidlo generuje méně inkrementů vzhledem ke konstantní vzdálenosti a tedy přichází o svoje rozlišení.

Další důležitý faktor, který ovlivňuje přesnost čidla je druh a barva povrchu, po kterém se čidlo pohybuje. Pokud se výsledný senzor přímo dotýká povrchu, jako například PC myš, povrch by měl být samozřejmě rovný, bez nějakých výstupků. Vysloveně nevhodné jsou pravidelné opakující se vzory, radikálně zvyšují chybovost senzoru. Zároveň je třeba použít matný povrch, který neodráží světelné paprsky. Uvádím zde graf chybovosti čidla v závislosti na různých druzích povrchu, viz. obrázek (6.5).

Obecně lze z principu umístění senzoru provést dvěma způsoby. Prvním, a dle mého názoru vhodnějším, je připevnit k pevné části rámu snímací disk, přesněji řečeno prstenec, o vnějším průměru minimálně 150,0 mm (je třeba si uvědomit, že čím větší obvod disku použijeme, tím dosáhneme většího počtu inkrementů vhledem k 360 stupňům, a tak dosáhneme větší přesnosti, avšak tato velikost je do jisté míry omezena konstrukcí) a senzor umístit ke pohyblivé části rámu tak, že při otáčení pohyblivé části zrcadla snímáme z povrchu disku.



Obrázek (6.5) : Závislost chybovosti senzoru na různých druzích povrchu (graf je vyňat z dokumentace k obvodu HDNS-2000)

Disk by měl být umístěn v těsné blízkosti optického vstupu senzoru. Také by se ale neměl dotýkat, aby nedocházelo jednak k blokování pohybu otočné části zrcadla a jednak k posunu senzoru vůči své výchozí poloze. Je třeba pomyslet také na to, že by povrch měl být vodovzdorný, protože celé zařízení bude umístěno venku.

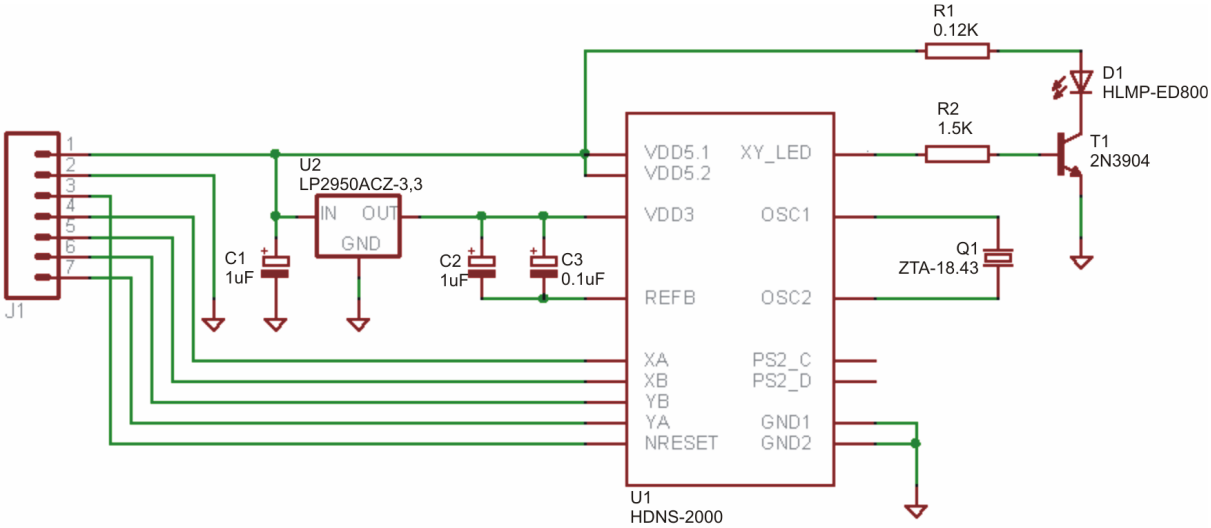
Druhá možnost je snímání z obvodu válce. Tato možnost je také jistě zajímavá, ale může zde docházet v případě malého průměru snímacího válce k nepřesnému generování inkrementů, a tedy k snižování celkové úhlové přesnosti. Toto řešení je pro tento systém konstrukčně nevýhodné a proto jsem se mu dále nevěnoval.

Při snímání z povrchu disku tedy dochází ke generování inkrementů, které jsou přenášeny do výpočetní části systému tam převedeny na úhel. Zvýšením nebo snížení obvodu snímacího kotouče lze tedy zvyšovat počet inkrementů na jednotku dráhy a tím dosáhnout požadovaného úhlového rozlišení.

Nevýhodou tohoto typu snímání úhlu je, že se jedná o inkrementální senzor a ne o senzor absolutní. Tedy při výpadku napájení senzor ztrácí informaci o poloze. Tento problém se ale dá vyřešit softwarově, kdy procesor bude zapisovat aktuální úhlovou hodnotu do paměti EEPROM. Další možnost je hardwarové řešení přidáním záložního zdroje, popřípadě kombinací obou přístupů.

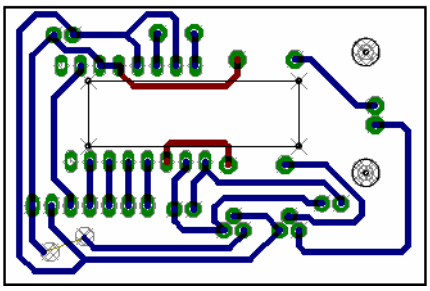
6.4 Konstrukce polohového senzoru s obvodem HDNS-2000

Obvod HDNS 2000 není u nás možné běžně zakoupit v obchodech s elektronickými součástkami. Je možné však možné jej získat z běžně dostupné pětitlačítkové optické myši od výrobce Genius. Navíc pro správnou funkci senzoru jsou potřeba ještě čočka a stínítko diody. Všechny tyto součástky jsou použity ve zmíněné myši. Z rozebrané myši jsem odpájel nejen obvod HDNS 2000, ale také jsem využil některých dalších součástek zde obsažených. Vlastní návrh plošného spoje jsem provedl v programu Eagle. Všechny použité soubory jsou přiloženy v elektronické příloze [15]. Zapojení obvodu je znázorněno na obrázku (6.6) .



Obrázek (6.6) : Schéma zapojení polohového senzoru

Zhotovení plošného spoje bylo provedeno osvitem na kuprexitovou destičku. Tu jsem potom doupravil vyvrtáním příslušných otvorů pro součástky a zejména otvoru pro kameru na spodní straně čidla HDNS-2000. Při rozvržení součástek na desce bylo třeba respektovat některé důležité rozměry, jako je rozměr a umístění obdélníkového otvoru pro kameru a vzdálenost diody od tohoto otvoru.



Obrázek 6.7 : Návrh plošného spoje

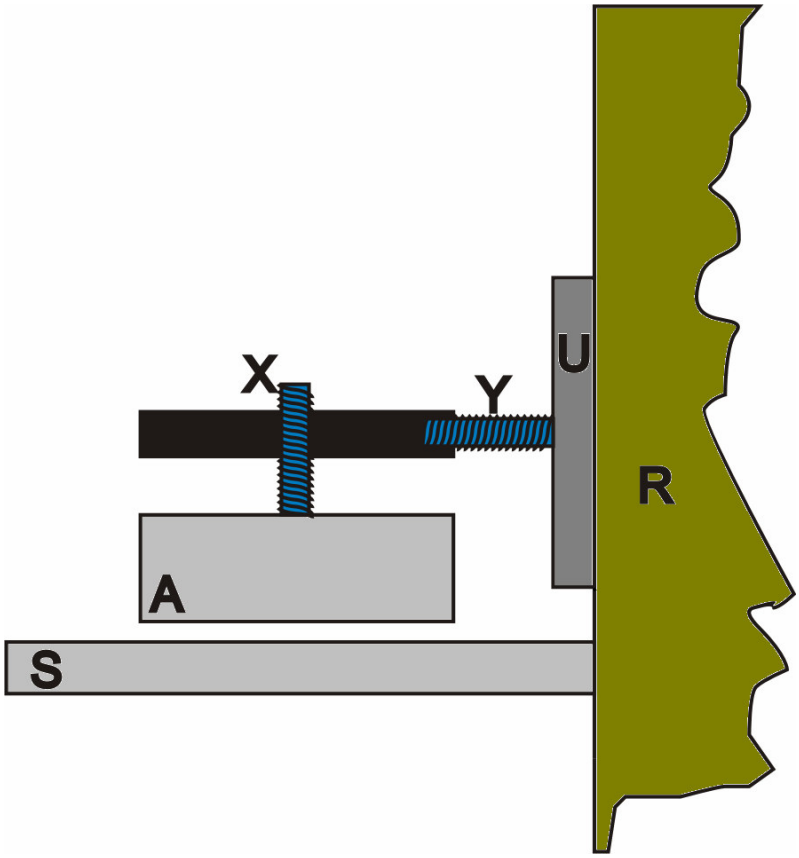
Velmi důležité je umístění kamery na spodní straně čidla a čočky do jedné osy. Všechny potřebné rozměry jsem získal z původního plošného spoje a tyto rozměry jsou součástí projektu pro návrh plošného spoje vytvořeném v programu Eagle, který je obsažen v CD příloze [15]. Správnou vzdálenost kamery od povrchu, stejně tak jako umístění čočky a stínítka jsem zajistil použitím původního krytu z PC myši. Vzdálenost kamery od povrchu respektuje graf na obrázku (**6.4**).

Obrázek (6.8) : Polohový senzor s čidlem HDNS-2000

Výsledný senzor je zobrazen na obrázku (6.8). V této fázi je zatím možné jej používat pouze v laboratorních podmínkách. Pro venkovní použití jej bude nutno zakrytovat, abychom jej ochránili před venkovní vlhkostí.

6.5 Umístění senzoru na zařízení

Snímání povrchu a následné vyhodnocování posunu není nikdy absolutně přesné. Může se stát, že DSP procesor uvnitř čidla špatně vyhodnotí obrazovou informaci z kamery. Chybovost čidla je minimální při správně „zaostřeném“ senzoru, tedy je třeba dbát na to, abychom čočku, která je nasazená na obvod HDNS-2000, umístili do správné vzdálenosti od snímaného povrchu. Co se týče povrchu snímacího disku, je z hlediska minimalizace chyb nejlepší použít bílou PVC desku, na které je chyba snímání minimální i při vzdálenosti menší než je ta doporučovaná viz. obrázek (6.4) a (6.5) a navíc se tento materiál hodí pro venkovní použití.



Obrázek (6.5) : Návrh připevnění senzoru ke konstrukci rámu

Na současnou konstrukci bych proto navrhoval připevnit a to nejlépe pomocí sváru (celá konstrukce je ze železa) úchyt pro senzor, který například pomocí šroubových závitů umožní jednak korigovat vzdálenost senzoru od snímacího disku, a také jeho vzdálenost od středu otáčení jak je naznačeno na obrázku (6.5). Písmeno A značí polohový senzor jehož vzdálenost od snímacího disku S je nastavitelná pomocí šroubu X. Celá konstrukce je připevněná svárem pomocí plochy U k otočnému rámu R. Pomocí šroubu Y lze nastavit

snímání v určité vzdálenosti od osy otáčení rámu. Pomocí šroubu X bude tedy možné zaostřit kameru čidla na ideální vzdálenost a tím maximalizovat výkonnost senzoru. Dále zde můžeme využít skutečnosti, že čidlo HDNS-2000 generuje přesný počet inkrementů na milimetr (2, 4, 8 a 16) viz [16]. Nastavením správného poloměru otáčení pomocí šroubu Y můžeme nastavit senzor tak, že jednotlivým inkrementům odpovídá požadovaný úhlový posun. Například pokud chceme nastavit senzor tak, aby jeden inkrement čidla odpovídal 0.2^{o}, potřebujeme vygenerovat celkem 1800 inkrementů vzhledem ke 360°. Při nastavení senzoru na 4 inkrementy na jeden milimetr potřebujeme obvodovou vzdálenost 450.0 mm, které odpovídá poloměr 143,24 mm. V neposlední řadě bude třeba na konstrukci vhodně připevnit snímací disk, u kterého je pouze třeba zaručit jeho vodorovné umístění.

Během pohybu rámu může dojít k načítání chyby senzoru. Toto lze částečně eliminovat pomocí kalibrace senzoru. Kalibrace se provede pomocí pevně definovaného bodu, který bude představovat referenční úhel. Tento bod bude definován pomocí spínače umístěného v jednom extrému natočení, který při sepnutí vyvolá přerušení procesoru. Při kalibraci nám dobře nahrává fakt, že zařízení bude pracovat pouze ve dne a na každém sklonku dne se systém bude automaticky natáčet se na polohu určenou pro další den. Je tedy možné provést kalibraci senzoru právě v této fázi řídicího algoritmu. Spínače v extrémních polohách natočení mohou také sloužit jako ochrana zařízení. Pokud by totiž došlo k otočení o úhel větší než je maximální povolený, mohlo by dojít k poškození zařízení.

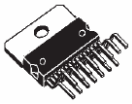
Kapitola 7

Návrh a realizace výkonové části

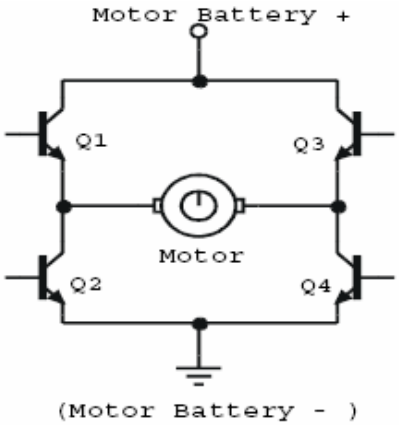
7.1 Problémy

Polohovací jednotka používá dvanácti voltový stejnosměrný motor. Maximální hodnota proudového zatížení je 500mA, doporučená hodnota je 360mA. Přesnější informace o motoru nemám k dispozici. Jelikož točivý moment motoru je velice malý a sám o sobě by motor nebyl schopen pohnout zrcadlem, používá převodový systém. Otáčky motoru je možné řídit změnou napájecího napětí, nebo je možné použít PWM modulaci. Vzhledem k tomu, že použitý mikroprocesor obsahuje generátor PWM signálu, jsem v návrhu použil druhou možnost.

7.2 Realizace pomocí obvodu L298



Pro realizaci výkonové části řídicí jednotky jsem použil integrovaný obvod L298, jehož principielní schéma je na obrázku (7.1).

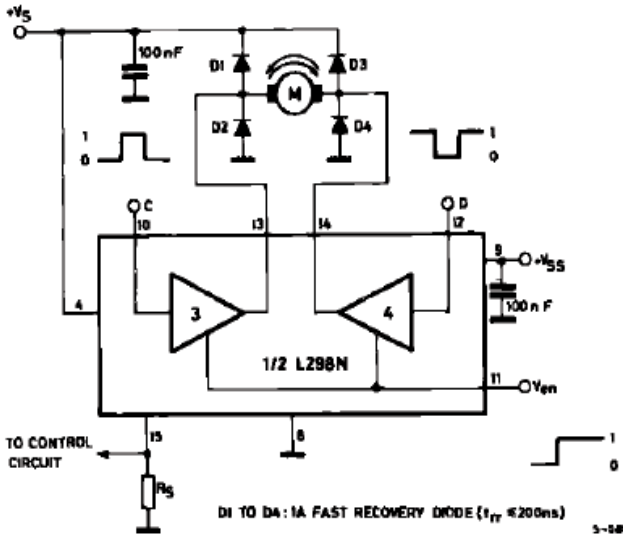


Obrázek (7.1) : Princip obvodu L298

Napájecí napětí tohoto obvodu je do 48V a maximální proud je 4 A. Detailní informace o tomto obvodu od výrobce přikládám v příloze [17].

Obvod umožňuje nezávislé řízení dvou motorů na principu PWM pomocí dvou H-můstků a zároveň zajišťuje oddělení výkonové části od zbytku obvodu. Princip ovládání motoru je patrný ze schématu na obrázku. Při nastavení signálů Q1 a Q4 na úroveň High se

motor otáčí vpřed a při nastavení Q2 a Q3 na úroveň High vzad. Obvod také umožňuje provést tzv. „fast motor stop“ a „free runnig motor stop“, tedy okamžité zastavení motoru a zastavení s dojezdem. Vstupní řídicí signály obvodu jsou v úrovni TTL. Tyto signály jsou generované mikroprocesorem. Pro řízení motoru do obou směrů je potřeba generovat signál PWM (připojí se ve schématu na obrázku na vstupy Q1 a Q4) a jeho negaci (tento signál se připojí na vstupy Q2 a Q3). Takto je tedy převeden signál PWM z úrovně 0 až 5V na úroveň –Vs až +Vs, kde Vs je napájecí napětí obvodu a motor se podle sřídy genorovaného signálu točí vpřed či vzad. Velikost protékaného proudu lze regulovat pomocí přídavných rezistorů, které se připojí mezi vstupy sense A a sense B a mezi zem a jejichž velikost udává velikost proudu, která protéká výkonovou částí.



Obrázek (7.2) : Schéma zapojení obvodu L298 pro obousměrné řízení motoru, (použité schéma pochází z dokumentace k obvodu L298 [17])

Pro obousměrné ovládání motoru výrobce doporučuje do zapojení umístit vnější můstek, tvořený diodami D1 až D4 viz obrázek (7.2). Pro rychlejší průběhy je pro konstrukci diodového můstku doporučováno použít Schottkyho diody, které mají malé hodnoty Trr (reverse recovery time), což je parametr které charakterizuje, za jak dlouho se dioda přepne z nevodivého do vodivého stavu. Diodový můstek slouží jako ochrana obvodu a v návrhu řídicí jednotky jsem ho použil. Zapojení obvodu je uvedeno jako součást celkového zapojení řídicí jednotky v kapitole 7, nebo v přiložené CD dokumentaci [18].

Kapitola 8

Návrh a realizace řídicí části

8.1 Mikroprocesor ATmega16

Tento mikroprocesor patří do rodiny AVR, 8-bitových riscových procesorů, vyráběných firmou Atmel. Na čipu je integrovaná 16K Bytová paměť typu Flash, která slouží jako paměť programu, dále potom 512 Bytů paměti typu EEPROM a ještě 1K Byte vnitřní paměti SRAM. Procesor obsahuje tři zdroje vnějšího přerušení, dva 8 bitové a jeden 16 bitový čítač/časovač, 4 programovatelné PWM kanály a celkem 64 IO portů. Podrobné informace o procesoru jsou obsaženy v elektronické příloze [19], nebo je lze nalézt na internetových stránkách výrobce [8].

Mikroprocesor je nejdůležitější součást řídicí jednotky fokusačního zařízení. Jeho primární úlohou je výpočet potřebných souřadnic, tedy referenčního signálu pro řízení. Dále pak slouží jako obslužná jednotka pro polohový senzor, generuje řídicí signál PWM pro výkonovou část jednotky a zastává funkci regulátoru. Dále pak komunikuje s obvodem reálných hodin, pro získání časového údaje, potřebného pro polohování, a pomocí rozhraní RS-232 komunikuje s servisním PC, z něhož jsou zadávány hodnoty parametrů.

8.2 Obvod reálného času RTC-62421

Přesný čas je pro tuto řídicí aplikaci velice důležitý. Představuje základní referenční signál, od kterého se pak odvíjejí všechny další řídicí akce. Je možno namítnout, že funkci hodin by mohl, s použitím krystalu s přesně definovanou frekvencí, zastat sám mikroprocesor. Procesor však obsluhuje celou řadu dalších funkcí. Pokud by byly hodiny řešeny pomocí vnitřního čítače v kombinaci s přerušením v případě přetečení čítače a následným odskočením do obsluhy přerušení mohlo by v této chvíli dojít souběhu s vnějším přerušením od polohového čidla (procesor ATmega16 nemá nijak řešený problém s prioritami přerušení). Takto by pak nebyla přesnost hodin nijak garantovaná a časem by se hodiny mohly rozcházet.

Garanci přesného času naopak zajišťuje obvod RTC-62421. Pro bezpečnost celkového návrhu je výhodnější, aby obvod hodin byl nezávislý na ostatních částech řídicí jednotky. Navíc lze počítat se záložním napájením hodin z baterie v případě přerušení napájení. Obvod RTC-62421 je obvod hodin reálného času, které obsahují vnitřní oscilátor. Hodiny v sobě obsahují časový údaj, kompletní kalendář včetně přestupných roků a také přerušovací systém pro nadřazenou jednotku, kterou zde zastává mikroprocesor. S mikro-

procesorem komunikuje pomocí 4 bitové sběrnice. Obvod obsahuje signál ALE a je tedy možné sloučit adresní a datovou sběrnici dohromady. Jednotlivé údaje hodin i kalendáře jsou uloženy v registrech, jejichž adresy jsou u vedeny v dokumentaci [20]. Obvod generuje nastavitelný periodický signál, který je připojen na procesor na vstup vnějšího přerušení. Pomocí něho je možné nastavit časový krok polohování, viz kapitola 4. Spádová hrana tohoto signálu vyvolá přerušení, při jehož obsluze je generován nový referenční signál. Nastavení přerušovacího signálu a hodnot časových registrů se provádějí v zápisem do systémových registrů RTC, podrobnější popis lze nalézt v dokumentaci k obvodu RTC-62421 [20], nebo na internetových stránkách výrobce [9].Výstup tohoto signálu je umístěn na pinu STD. Zápis a čtení se provádí pomocí signálů RD, WR, CS1 a CS2. Tyto signály jsou používány zcela standardně a jejich přesný význam stejně tak jako časové diagramy komunikace se nachází v dokumentaci [20].

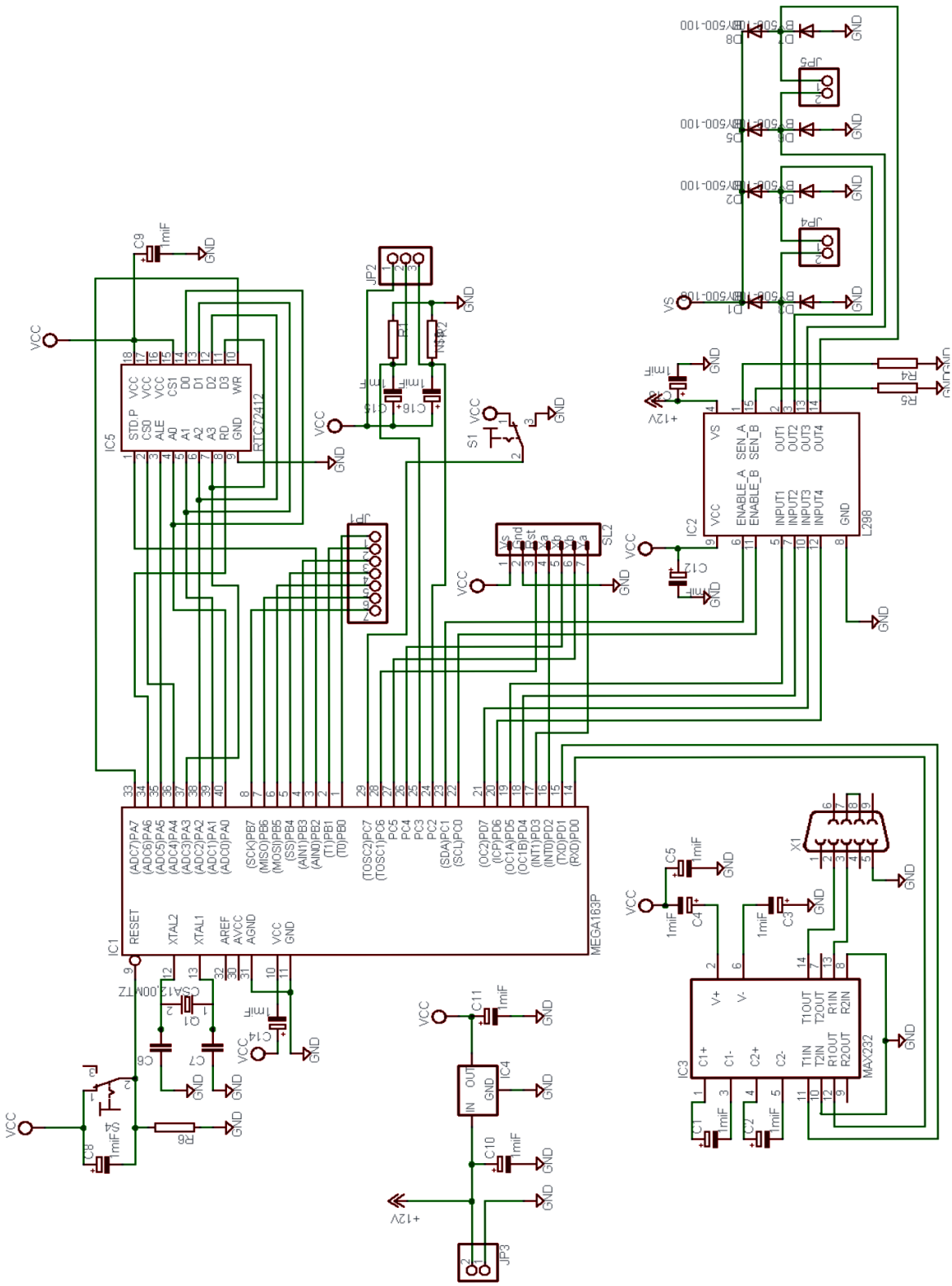
8.3 Převodník MAX-232

Dalším obvodem použitým v návrhu řídicí jednotky je převodník Max-232. Jedná se o velice rozšířený a používaný obvod, který slouží pro převod napěťových úrovní z TTL na úrovně používané ve standartu 232 a opačně. Veškeré potřebné informace o rozhraní RS-232 lze získat například na [10]. Pro správnou funkci převodníku je k jeho vývodům potřeba zapojit elektrolytické kondenzátory. Schéma zapojení se nachází například v [10]. Mikroprocesor komunikuje s PC pomocí sériového UARTU. Na svých výstupech generuje signál, který slouží jako vstup do převodníku v úrovních TTL. Výstupem je potom ten samý signál, avšak převedený na napěťové úrovně dle standardu RS-232, kde logická nula je rovna +10V a logická jednička -10V. Pokud je vyslán signál ze strany PC jsou napěťové úrovně převedeny v opačném pořadí.

8.4 Popis prototypové řídicí jednotky

Všechny obvody popsané v předchozích odstavcích tvoří součásti řídicí jednotky, kterou jsem prozatím realizoval pomocí univerzálního pájicího pole. Schéma zapojení je uvedeno na obrázku (8.1), uvádím ho zde však spíše pro ilustraci, neboť je v této grafické podobě poněkud nepřehledné. Toto schéma je však součástí elektronické přílohy [18], kde je možno jej prohlížet v přehlednější podobě.

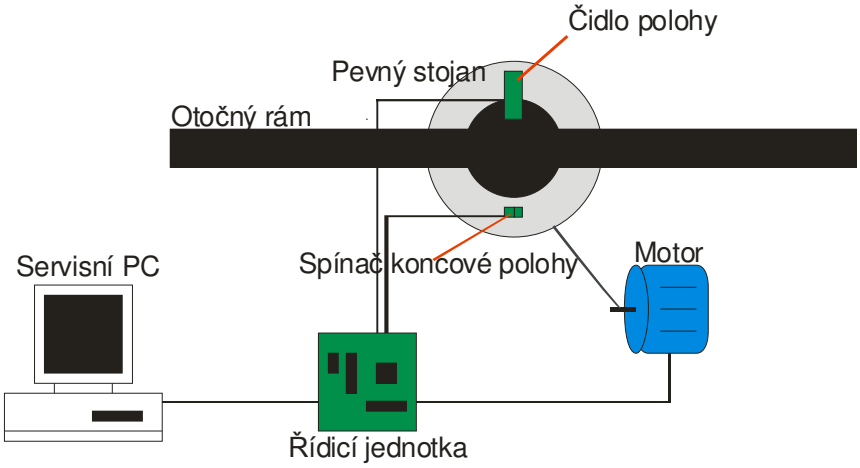
Procesor obsahuje celkem čtyři porty, každý po osmi pinech, z nichž jeden, Port B, zůstává nezapojen pro možnost pozdějšího využití, například pro řízení čtyřkvadrantovou (sektorovou) diodou. Celý obvod je napájen 12V napětím. Napěťový regulátor IC4 převádí toto napětí na úroveň 5V pro možnost napájení všech obvodů. 12-ti voltové napájení je



Obrázek (8.1) Schéma zapojení řídicí jednotky

připojeno pouze na obvod L298, který napájí motor. Spínač S4 provádí reset mikroprocesoru. Pomocí přepínače S1 lze přejít do servisního módu. Konektor SL2 slouží pro komunikaci a napájení polohového senzoru. Na konektor JP2 boudou připojeny spínače koncové polohy. Konektory JP4 a JP5 slouží jako napájení motoru. Konektor X1 slouží k připojení k PC. Seznam všech použitých součástek je součástí návrhu plošného spoje v programu Eagle a je přiložen v CD příloze [18].

Zpětnovazební řízení tvořeno tak, že procesor na základě časového údaje z RTC a na základě informace o zeměpisných souřadnicích uložených v EEPROM spočítá referenční signál, současně mu z čidla přichází informace o skutečné poloze. Po odečtení těchto hodnot získává regulační odchylku. Ta vstupuje do bloku regulátoru, který je v této chvíli implementován pouze jako proporcionální. Výstupem regulátoru je akční zásah, který je procesorem převeden na řídicí signál pro PWM, který pak vstupuje do výkonového obvodu. Ten pak provede příslušné otočení motoru a tím je vlastně uzavřena zpětná vazba.



Obrázek (8.2) Principielní schéma řídicího systému

Kapitola 9

Návrh řídicího algoritmu

9.1 Popis algoritmu

Struktura algoritmu je vytvořena tak, že po resetování mikroprocesoru prochází program inicializační částí, kdy jsou nastaveny hodnoty všech potřebných registrů na své výchozí hodnoty a poté program odskočí do nekonečné smyčky. V této smyčce může program v závislosti na vstupním signálu vykonávat servisní, nebo provozní část algoritmu.

Při sepnutém vstupu pro servisní údržbu odskočí program do servisní části a setrvává v této části dokud není pomocí přepínače přepnuto nazpět. V této fázi programu je regulace a polohování neaktivní. Při přechodu zpět se provede inicializace čidla a následné spuštění regulačního a polohovacího algoritmu. Mikroprocesor v této fázi programu komunikuje s PC pomocí standardního rozhraní RS-232. Komunikace probíhá pomocí standardního komunikačního hyperterminálu ve formě příkazové řádky. Po odeslání příslušného znaku je možné zadat nebo číst potřebné informace. Tabulka (9.1) uvádí seznam příkazů, použitých při komunikaci s mikroprocesorem.

T	Výpis času – vypisuje časový údaj v intervalu jedné vteřiny
T	Výpis datumu – vypíše aktuální datum
S	Nastavení hodin – v této je možné nastavit hodiny, které jsou tímto příkazem zastaveny
M	Nastavení jednotkové části minut
M	Nastavení jednotkové desítkové části minut
H	Nastavení jednotkové části hodin
H	Nastavení jednotkové desítkové části hodin
Q	Spuštění hodin
D	Nastavení jednotkové části dne
D	Nastavení jednotkové desítkové části dne
M	Nastavení jednotkové desítkové části měsíce
M	Nastavení jednotkové desítkové části měsíce
Y	Nastavení jednotkové části roku
Y	Nastavení jednotkové desítkové části roku
V	Nastavení zeměpisné výšky
S	Nastavení zeměpisné šířky

P	Nastavení časového pásma
p	Nastavení proporcionální regulační konstanty

Tabulka (9.1) : seznam příkazů pro komunikaci s procesorem

Poznámka *: V registrech obvodu hodin reálného času RTC-62421 jsou jednotkové a desítkové hodnoty uloženy v oddělených registrech, v BCD kódu. Nastavování parametrů je proto přizpůsobeno této skutečností.*

Při přijetí příkazu „t“ procesor přepsáním systémových registrů obvodu RTC nastaví přerušovací signál hodin na jednu vteřinu. V obsluze tohoto přerušení je pak pomocí standardní funkce odesílán obsah hodinových registrů na terminál.

Příkaz „s“ provede zápisem do příslušného systémového registru zastavení hodin. Opětné spuštění se provede příkazem q. V této fázi se provede vynulování vteřinových registrů, takže po každém spuštění hodin budou vteřiny startovat od nuly.

Mezi příkazem „s“ a „q“ je možné změnit příslušnými příkazy, viz. tabulka (9.1) obsah hodinových a minutových registrů.

Příkaz „T“ provede jednorázový výpis datumu. Nastavení datumu se provádí při běhu hodin prostým přepsáním příslušných registrů. Rok je zde zadáván pouze jednotkovou a desítkovou částí.

Příkazy „V“, „S“ a „P“ se uloží hodnoty lokálních zeměpisných souřadnic do paměti EEPROM, tyto hodnoty slouží k výpočtu souřadnic Slunce a budou zadávány v závislosti na umístění zařízení.

Hodnoty zadaných parametrů jsou zapsány do registrů obvodu RTC (samozřejmě pouze údaje týkající se hodin) a zároveň jsou uloženy do paměti EEPROM.
Hodnoty parametrů jsou do paměti ukládány, aby v případě výpadku napájení hodinového obvodu byly registry hodin, po následném restartu procesoru, nastaveny právě z těchto hodnot. Při výpadku napájení je totiž obsah registrů u RTC nedefinovaný. Výhoda tohoto zálohování spočívá v tom, že se při krátkém výpadku nebudou muset přenastavovat všechny údaje. Při delším časovém výpadku napájení bude nutné provést znovunastavení hodinových registrů pomocí servisního módu

Provozní část programové smyčky se dělí ještě na dvě podčásti. První je aktivní ve dne, kdy se provádí polohování a regulace. Pokud se výšková souřadnice při denním režimu dosáhne hodnoty nula, přechází program do druhé fáze, do nočního režimu.

V nočním režimu nejprve program určí čas východu Slunce, a jeho souřadnice na příští den. To provádí tak, že s minutovým krokem počítá výškovou souřadnici a dokud nedosáhne nulové úrovně. Vyhledávání této úrovně může začít od 0h min příštího dne. Po tomto výpočtu provede řídicí jednotka kalibraci čidla. Pak je nastavena nejdelší perioda přerušovacího signálu RTC, tedy jedna hodina a procesor přechází do režimu power-down. Z této fáze je vždy každou hodinu probouzen vnějším přerušením od RTC. Při probuzení se kontroluje jestli nestala poslední hodina před rozbřeskem. Pokud ne procesor přechází opět do power-down modu. Pokud tato hodina nastala je stále provázen přechod do power-down módu, avšak s minutovým krokem doku se nedosáhne předem určeného času probuzení.

Přesný čas probuzení se sice nemusí sejít s tímto minutovým krokem, ale jak plyne z kapitoly tolerance nastavení zrcadla toto připouští. Program potom přechází do denního režimu.

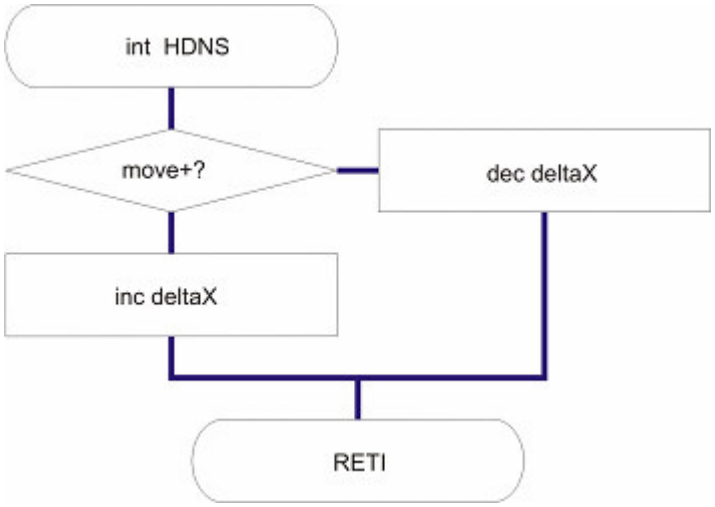
V denním režimu provádí řídicí program polohování a regulaci a obsluho polohového senzoru. Ovládání polohového senzoru je implementováno tím způsobem, že pokud vyvolá náběžná hrana signálu Xa, viz kapitola 6, přerušení program otestuje, zda je úroveň signálu Xb, připojeného na vstup procesoru, high nebo low. Poté buď přičte nebo odečte přírůstek polohy.

V souladu s výsledky z kapitoly 4 je možné nastavit polohovací krok na jednu minutu. Takto je zaručeno, že během přesunů zrcadla mezi těmito kroky není překročena tolerance jeho natočení.

Přerušovací signál RTC je tedy v tomto režimu nastaven na periodu jedna minuta, zapsáním příslušné hodnoty, viz [20], do systémového registru RTC. Tento signál vyvolá příslušné přerušení procesoru, v jehož obsluze se zavolá funkce na výpočet polohy Slunce s aktuálním časovým údajem. Tato hodnota vstupuje jako reference do programového bloku regulátoru. Zde se vypočte regulační odchylka pomocí požadované a skutečné hodnoty. V případě, že je regulační odchylka větší než je maximální povolená, provede program výpočet akčního zásahu regulátoru, který je vhodně převeden pro nastavení srovnávacího registru PWM. Hodnota tohoto registru určuje střídu signálu porovnáváním s čítačem, který generuje pilovitý signál. Generování řídicího signálu PWM má na starosti čítač 1, který pomocí nastavení systémových registrů pracuje ve funkci generátoru PWM signálu.

9.2 Vývojový diagram

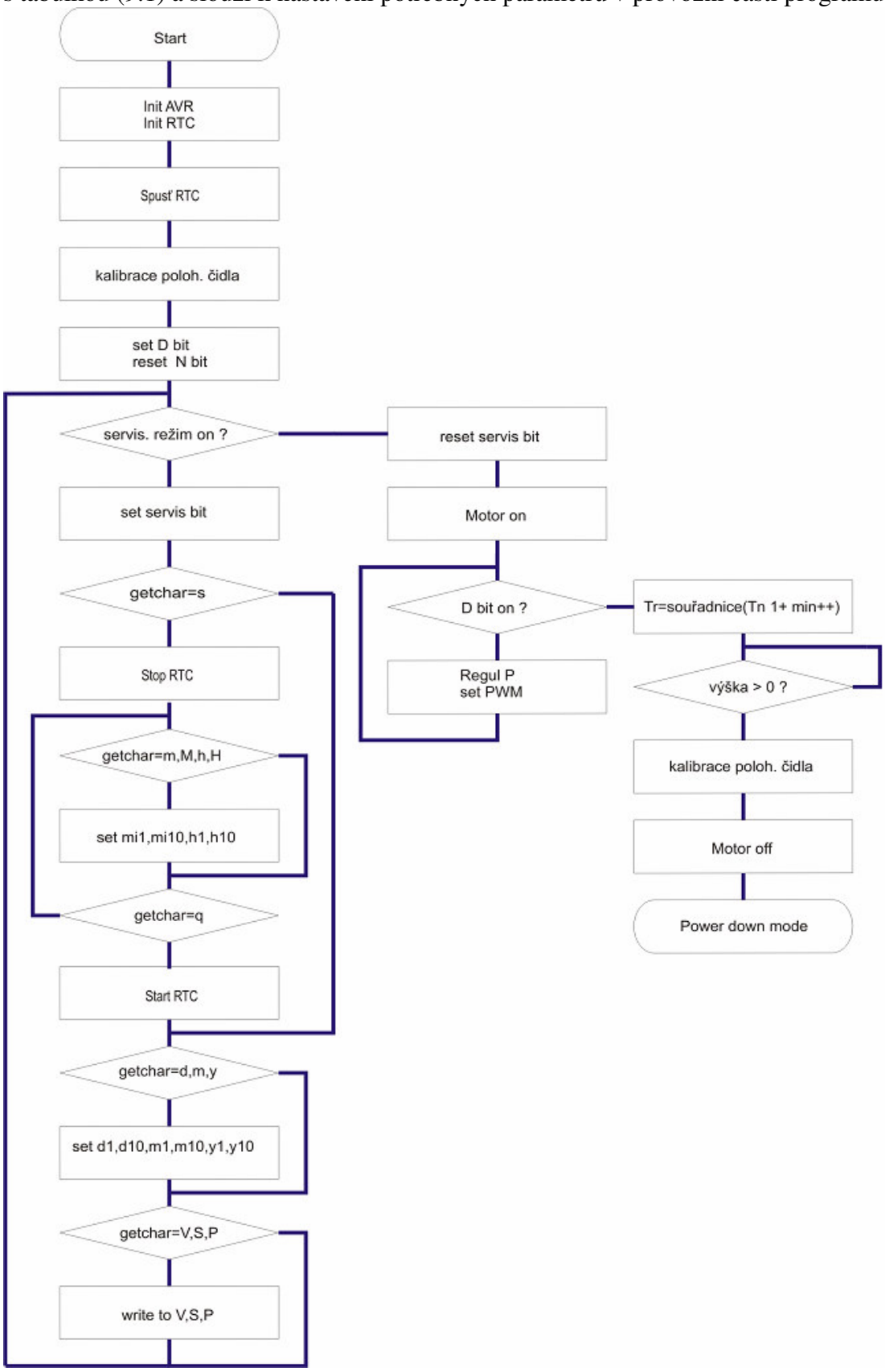
Návrh řídicího algoritmu jsem provedl pomocí vývojového diagramu, viz. obrázky (9.1), (9.2) a (9.3). Na obrázku (9.1) je vyznačena obsluha přerušení od polohového senzoru. DeltaX je proměnná, která čítá přírůstky polohy v kladném a záporném směru.



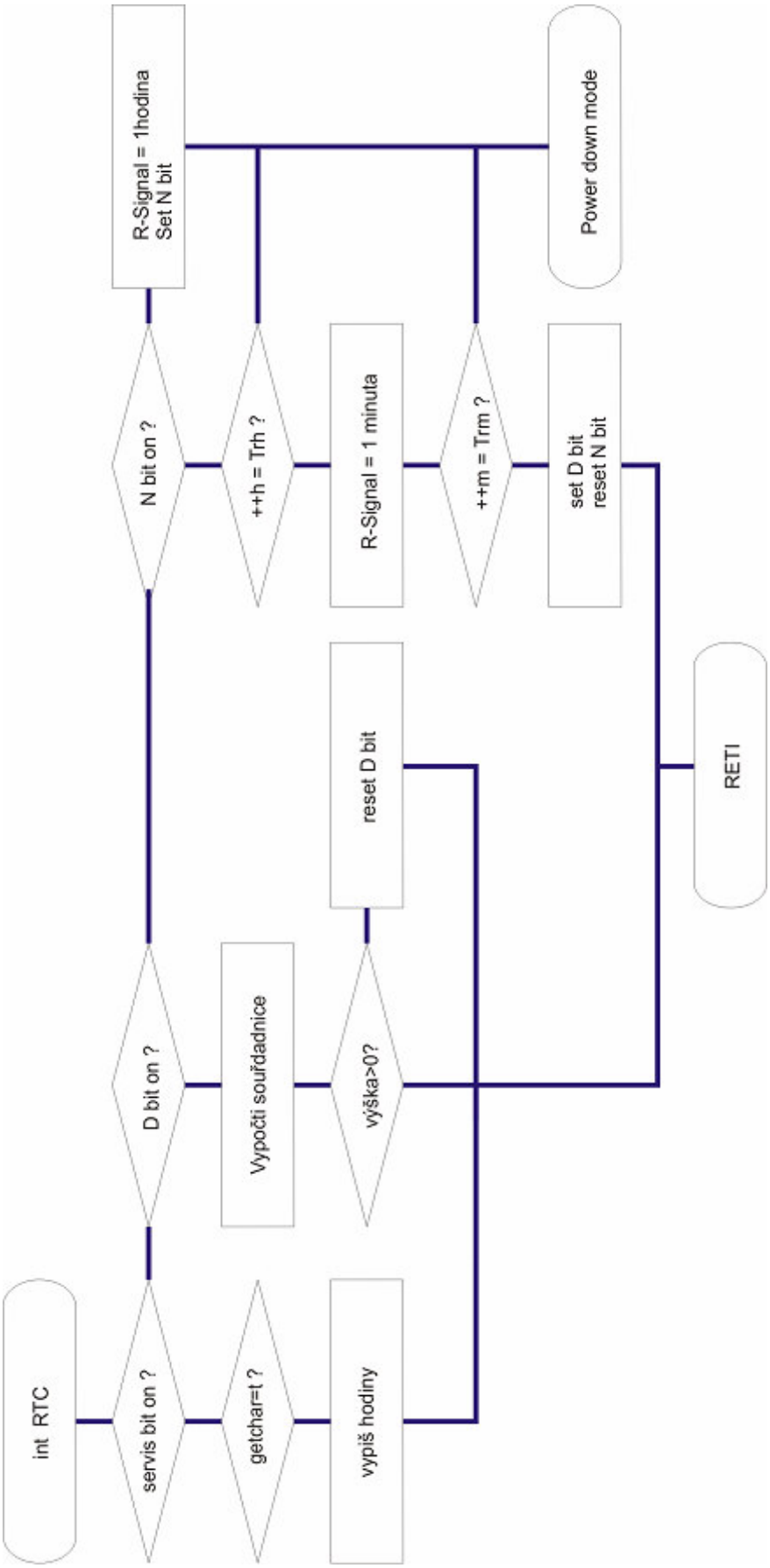
Obrázek (9.1) obsluha vnějšího přerušení od polohového senzoru

Obrázek (9.2) Obsahuje schéma hlavní programové smyčky. Pomocí příznaků servis bit, D bit a N bit se určuje, jaká fáze programové smyčky se má vykonávat, jestli servisní nebo

provozní denní, či noční část. Značení, například „getchar = s“ nebo „ set mi1“, se shoduje s tabulkou (9.1) a slouží k nastavení potřebných parametrů v provozní části programu.



Obrázek (9.2) Schéma hlavní programové smyčky

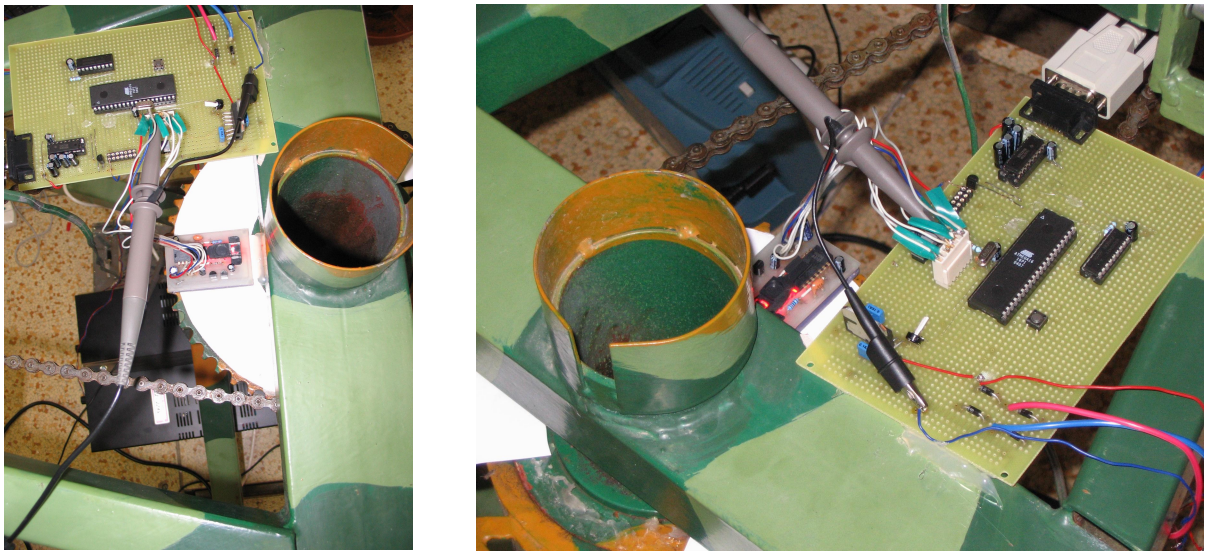


Obrázek (9.3) Schéma obsluhy vnějšího přerušení od RTC hodin

Myslím, že podrobnější popis celého vývojového diagramu není, třeba, protože význam jednotlivých bloků je intuitivní. Zároveň s návrhem vývojového diagramu jsem vytvořil kostru řídicího programu, jejíž zdrojový kód je přiložen v CD příloze [21]. V této kostře je zatím implementována komunikace s PC a obvodem reálného času, zadávání konstant pomocí servisního módu, obsluha polohového senzoru, a proporcionální složka regulátoru.

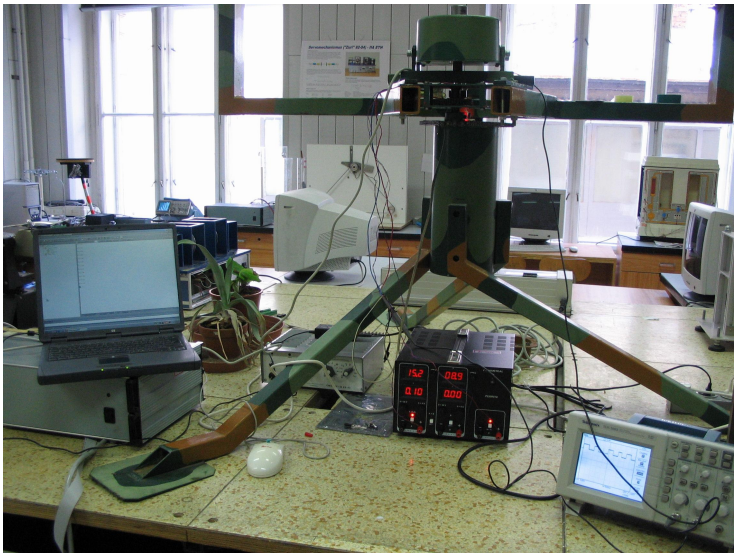
Kapitola 10

Měření a testování



Obrázek 10.1 : Připevnění senzoru a řídicí jednotky k polohovacímu zařízení

Prototyp řídicí jednotky a polohového senzoru jsem připevnil k polohovacímu zařízení, jak je vidět na obrázku (**10.1**). Na obrázku je též vidět umístění snímacího disku pro polohový senzor. Celé zařízení je ovládáno pomocí počítače, viz. obrázek (10.2). Jelikož jsem neměl k dispozici referenční body pro přesné definování úhlu a umístění senzoru do přesně definované vzdálenosti od osy otáčení je v tomto technickém provedení také nemožné (bude potřeba provést technická opatření popsaná v kapitole 6), vytvořil jsem testovací program, jehož zdrojový kód přikládám v CD příloze [21]. Pomocí tohoto počítače jsou řídicí jednotce zadávány konstanty PWM modulace a tím je řízen směr a rychlost otáčení rámu.
Zároveň je na počítači možné sledovat přírůstky polohového senzoru. Na snímací disk jsem přikládal rastr, do něhož zakresloval přesnou pozici senzoru. Takto jsem zařízením otáčel v rozmezí úhlů zhruba 10° - 40° v obou směrech otáčení a zaznamenával jsem přírůstky polohy. Po sejmutí rastrů jsem potom geometricky určil vzdálenost, o kterou se senzor pootočil a k této vzdálenosti jsem přiřadil příslušný počet inkrementů senzoru. Ve všech



Obrázek (10.2) : Polohovací zařízení

případech shodně vycházelo, že senzor provede čtyři inkrementy na jeden milimetr. Tady musím přiznat, že jsem zpočátku předpokládal, a z dokumentace k obvodu HDNS-2000 to nebylo zcela jasné, že v případě kvadraturního výstupu čidla bude senzor pracovat v maximálním rozlišení 16 inkrementů na milimetr. Avšak použitý obvod byl vyjmut z PC myši, kde je příslušným nastavením systémového registru obvodu rozlišení nastaveno na nižší hodnotu. (Při použití v myši není třeba používat velké rozlišovací schopnosti obvodu) Pokud bychom chtěli využít maximální rozlišovací schopnost obvodu, bylo by třeba připojit ještě jeho výstupy PS2_C a PS2_D připojit k procesoru a vytvořit v datovém rámci PS2 příkaz pro přepsání hodnotu příslušného registru. Hodnoty těchto registrů a jejich adresy jsou uvedeny v dokumentaci k obvodu [20]. Avšak i v tomto nastavení, při současném umístění senzoru (zhruba 72,0mm od osy otáčení), je možné dosáhnou nejmenšího dílku na úhlové stupnici 0.2^{o}. To je hodnota, která lze pro řízení a polohování tohoto systému použít.

Závěr

V této práci jsem byl postaven před velice rozsáhlý a komplexní problém, v němž některé úlohy, jak se během práce ukázalo, v sobě skrývaly další podúlohy, které bylo nutno řešit. Práce zahrnovala řešení praktických technických problémů, jako je návrh a výroba elektronických obvodů, nebo montáž senzoru, stejně tak, jako využití teoretických znalostí z oblasti řízení, astronomie, programování a jiných. Velkou motivací pro mě byla možnost vyzkoušet některé teoretické znalosti v praktickém provedení a získání znalostí z oblastí mě do té doby neznámých.

Celkově bych řekl, že jsem práci na projektu uzavřel ve fázi, po které musí následovat nutné doplnění polohovacího mechanismu některými technickými prvky, jako je montáž zrcadla, úchytu pro senzor, aretovacích spínačů, a pod., po nichž bude možné na projektu pokračovat dál.

Toto pokračování v sobě ponese nejprve návrh a realizaci snímání polohy ve vertikální poloze, pokud nebude použit stejný princip, jako v ose horizontální, a dále pak identifikaci celého systému (i se zrcadlem) a optimalizaci návrhu regulátoru, popřípadě analýzu dalších technických opatření, jako například výměnu hybné soustavy, a pod. . Posledním krokem v celkovém návrhu by mělo být navržení „doostřovacího“ sytému pro korekci polohovacího algoritmu a návrh řízení zrcadla pokud dojde k přehřívání oběhového média.

Zatím byla dokončena analýza celého problému. Byl proveden návrh řídicího a polohovacího algoritmu, návrh řídicí jednotky, schopné pracovat s navrženými algoritmy, a realizace snímání polohy v jedné ose.

Některé problémy se nepodařilo vyřešit až do konce, jako je například regulace polohové jednotky, i když k tomu chyběl už jen malý krok, který byl právě způsoben problémy technického rázu.

Použitá literatura :

[1] Emil Buchar : Geodetická astronomie. ČVUT, 1986.
[2] Zdeněk Pokorný : Astronomické algoritmy pro kalkulátory. Hvězdárna a planetárium hlavního města Prahy, 1988.
[3] Pavel Příhoda : Hvězdářská ročenka 2004, Praha, Hvězdárna a planetárium hlavního města Prahy, 2003.
[4] Jan Fixel : Geodetická astronomie I. a základy kosmické geodézie. Vutium, 2000.
[5] Mark Trueblood a Russell Merle Genet : Teleskope control. Willmann-Bell, 1997.
[6] Vladimír Váňa : Mikrokontroléry ATMEL AVR - programování v jazyce C. BEN, 2003.
[7] Vladimír Váňa : Mikrokontroléry ATMEL AVR - popis procesorů a instrukční soubor. BEN, 2003.
[8] Atmel [online], Poslední revize 20.1.2005, [http://www.atmel.com/]
[9] Epson elektronics [online], Poslední revize 20.1.2005, [http://www.eea.epson.com/go/EEA]
[10] RS-232 [online], Poslední revize 20.1.2005, [http://rs232.hw.cz/]
[11] Agilent Technologies [online], Poslední revize 20.1.2005, [http://www.semiconductor.agilent.com]

Obsah CD přílohy :

[12] Zdrojový kód algoritmu na výpočet polohy Slunce
[13] Zdrojový kód algoritmu na výpočet polohy Slunce ve formě MEX-file
[14] Zdrojový kód programu na výpočet polohy a rychlosti Slunce během roku
[15] Návrh polohového senzoru, projekt Eagle
[16] Dokumentace k obvodu HDNS-2000
[17] Dokumentace k obvodu L298
[18] Návrh řídicí jednotky, projekt Eagle
[19] Dokumentace k obvodu ATmega16
[20] Dokumentace k obvodu RTC-62421
[21] Zdrojový kód řídicího algoritmu
[22] Zdrojový kód testovacího programu
[23] Dokumentace k obvodu HEADS-1500
[24] Diplomová práce, formát word
[25] Diplomová práce, formát pdf