VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV ELEKTROENERGETIKY
DEPARTMENT OF ELECTRICAL POWER ENGINEERING

SOLÁRNÍ KONCENTRÁTOROVÝ SYSTÉM S AUTOMATICKÝM NATÁČENÍM
SOLAR CONCENTRATOR SYSTEM WITH AUTOMATIC TRACKING SYSTEM

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Martin Žák**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Petr Baxant, Ph.D.**
SUPERVISOR

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Elektroenergetika**
Ústav elektroenergetiky

Student: Bc. Martin Žák ***ID:*** 143642
Ročník: 2 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

Solární koncentrátorový systém s automatickým natáčením

POKYNY PRO VYPRACOVÁNÍ:

1. Zdokumentování současného stavu techniky v oblasti koncentrátorových systémů
2. Technologie sledování slunce a jejich aplikace
3. Ovládání krokových motorů pomocí platformy Arduino
4. Vytvoření funkčního modelu sledovacího zařízení
5. Návrh nebo výběr pohonů pro reálné zařízení

DOPORUČENÁ LITERATURA:

podle pokynů vedoucího práce

Termín zadání: 6.2.2017 ***Termín odevzdání:*** 22.5.2017

Vedoucí práce: doc. Ing. Petr Baxant, Ph.D.
Konzultant:

doc. Ing. Petr Toman, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:
Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Diplomová práce se zabývá současnými technologiemi využití sluneční energie, zejména solárními koncentrátory, a možnostmi aplikace systémů pro sledování polohy slunce za účelem maximalizace energetického zisku těchto zařízení. Rovněž pojednává o algoritmu výpočtu polohy slunce a předkládá návrh automatického sledovače založeného na platformě Arduino, který umožňuje automatické otáčení různých druhů solárních zařízení pomocí krokových motorů, stejnosměrných motorů nebo servomotorů, a to na základě vypočítané polohy slunce nebo dle skutečné intenzity osvětlení.

KLÍČOVÁ SLOVA

tracking; tracker; sledovač; solární koncentrátor; výpočet polohy slunce; Arduino; fotovoltaická elektrárna

ABSTRACT

This master's thesis is focused on current possibilities of solar energy utilization – especially on concentrating solar power systems – and their optimization using solar tracking systems to maximize energy gain. The thesis also deals with solar position algorithm and offers an Arduino based automatic dual axis solar tracker design which provides various solar devices the ability to pivot using stepper motors, DC motors or servos. The motion might be driven by calculated solar position or by actual illuminance.

KEYWORDS

solar tracking; solar tracker; sun tracker; solar concentrator; solar position calculation; Arduino; photovoltaic power station

ŽÁK, Martin. *Solární koncentrátorový systém s automatickým natáčením*. Brno, 2017, 131 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav elektroenergetiky. Vedoucí práce Petr Baxant.

Vysázeno pomocí balíčku `thesis` verze 2.63; `http://latex.feec.vutbr.cz`

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Solární koncentrátorový systém s automatickým natáčením“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

OBSAH

Úvod **15**

1 Technologie aktivního využití sluneční energie **17**
1.1 Výroba tepla . . . 17
1.1.1 Solární kolektory . . . 17
1.1.2 Solární koncentrátory . . . 20
1.2 Výroba elektrické energie . . . 24
1.3 Sledování polohy slunce . . . 24
1.3.1 Pasivní trackery . . . 25
1.3.2 Aktivní trackery . . . 27

2 Výpočet polohy slunce **33**
2.1 Souřadnicové systémy . . . 33
2.2 Algoritmus výpočtu . . . 34
2.2.1 Výpočet juliánského a juliánského efemérního data . . . 35
2.2.2 Výpočet heliocentrických souřadnic Země . . . 36
2.2.3 Výpočet geocentrických souřadnic . . . 37
2.2.4 Výpočet nutace délky a sklonu osy . . . 37
2.2.5 Výpočet skutečného sklonu dráhy k ekliptice . . . 39
2.2.6 Výpočet korekce aberace světla . . . 39
2.2.7 Výpočet zdánlivé délky slunce . . . 40
2.2.8 Výpočet zdánlivého hvězdného (siderického) času . . . 40
2.2.9 Výpočet geocentrické rektascenze slunce . . . 40
2.2.10 Výpočet geocentrické deklinace slunce . . . 41
2.2.11 Výpočet lokálního hodinového úhlu . . . 41
2.2.12 Výpočet topocentrické rektascenze slunce . . . 41
2.2.13 Výpočet topocentrického hodinového úhlu . . . 42
2.2.14 Výpočet topocentrického zenitu . . . 43
2.2.15 Výpočet topocentrického azimutu . . . 44
2.3 Ověření algoritmu výpočtu . . . 44
2.4 Získaná data . . . 45

3 Návrh solárního trackeru a pohyblivého stojanu **47**
3.1 Pohonné jednotky . . . 47
3.1.1 Krokové motory . . . 47
3.1.2 Stejnosměrné motory . . . 50
3.1.3 Servomotory . . . 52

3.2 Řídicí elektronika . . . 52
3.2.1 Čidlo slunečního svitu . . . 53
3.2.2 Snímače polohy . . . 54
3.2.3 Koncové pohybové spínače . . . 57
3.2.4 Ostatní komponenty . . . 58
3.2.5 Celkové schéma zapojení . . . 58
3.3 Softwarová část . . . 61
3.3.1 Hlavička a deklarace proměnných . . . 61
3.3.2 Funkce pro práci se soubory a s časem . . . 63
3.3.3 Funkce setup . . . 65
3.3.4 Funkce loop . . . 69
3.3.5 Varianty pohonů . . . 73
3.3.6 Volitelné příslušenství . . . 78
3.3.7 Spínání trackeru . . . 79
3.4 Pohyblivý stojan solárního zařízení . . . 80

4 Konstrukce, ověření funkčnosti a analýza 83
4.1 Konstrukce trackeru . . . 83
4.2 Konstrukce pohyblivého stojanu . . . 85
4.3 Ověření funkčnosti trackeru . . . 88
4.4 Ekonomická analýza . . . 88
4.4.1 Výrobní náklady . . . 89
4.4.2 Komerční systémy . . . 91

5 Závěr 93

Literatura 95

Seznam použitých zkratek a symbolů 101

Seznam příloh 103

A Data pro výpočet polohy slunce dle SPA 105

B Svorkovnice na desce plošných spojů 113

C Základní zdrojový kód pro Arduino 119

D Řízení sledovače pomocí zpracování obrazu 129

E Obsah přiloženého CD 131

SEZNAM OBRÁZKŮ

1.1 Optické a tepelné ztráty solárního kolektoru 18
1.2 Princip selektivní absorpční vrstvy 19
1.3 Řez vzduchovým kolektorem 19
1.4 Parabolický žlabový koncentrátor 21
1.5 Fresnelův koncentrátor 22
1.6 Parabolický diskový koncentrátor 23
1.7 Koncentrátorový systém s centrální věží 23
1.8 Fotovoltaický panel s koncentrátorovým modulem FLATCON 25
1.9 Schematické znázornění pasivního trackeru s kapalinovým vyvažováním 26
1.10 Princip fungování SMA trackeru před a po východu slunce 26
1.11 SMA tracker pro fotovoltaický panel 27
1.12 Různá provedení trackeru s optickými senzory 28
1.13 Naměřená intenzita slunečního záření v průběhu dne 7. 5. 2001 28
1.14 Schéma trackeru s oboustrannými fotočlánky (půdorysné zobrazení) . 29
1.15 Fotovoltaické zařízení s trackerem s oboustrannými fotočlánky 30
2.1 Schematické znázornění slunečních úhlů 34
3.1 Schéma zapojení unipolárního a bipolárního řízení krokového motoru 48
3.2 Unipolární jednofázové řízení krokového motoru s plným krokem . . . 49
3.3 Bipolární jednofázové řízení krokového motoru s plným krokem 49
3.4 Unipolární dvoufázové řízení krokového motoru s plným krokem . . . 49
3.5 Bipolární dvoufázové řízení krokového motoru s plným krokem 49
3.6 Unipolární řízení krokového motoru s polovičním krokem 50
3.7 Bipolární řízení krokového motoru s polovičním krokem 50
3.8 Příklad řízení PWM 51
3.9 Princip PWM řízení servomotoru 52
3.10 Schéma zapojení čidla slunečního svitu 53
3.11 Hotové čidlo slunečního svitu 54
3.12 Aplikace Kálmánova filtru na údaje o úhlu natočení 56
3.13 Schéma zapojení potenciometru pro měření úhlu natočení 56
3.14 Varianta zapojení koncových spínačů pro přímé zastavení motoru . . 58
3.15 Přehledové schéma zapojení celé sestavy 59
3.16 Rozvržení prvků na desce plošných spojů 60
3.17 Schéma řízení stejnosměrných motorů 61
3.18 Model pohyblivého stojanu pro malý koncentrátor 80
3.19 Detail ložiska pohyblivého stojanu 81
3.20 Model kompletní sestavy koncentrátoru včetně pohonu 82
4.1 Rozložení svorkovnic a popis svorek zjednodušené desky 83

4.2 Hotový tracker s krytem 84
4.3 Pohyblivý stojan včetně koncentrátoru (přední pohled) 85
4.4 Pohyblivý stojan včetně koncentrátoru (boční pohled) 86
4.5 Modely sledovacích zařízení s krokoými motory a servomotory 87
4.6 Tracker SunTura s čidlem slunečního svitu 91

SEZNAM TABULEK

1.1 Energetické zisky některých solárních kolektorů 20
1.2 Naměřená průměrná intenzita slunečního záření 29
2.1 Vstupní hodnoty pro výpočet polohy slunce 44
2.2 Výsledné hodnoty polohy slunce 45
4.1 Cenová kalkulace trackeru 90
A.1 Data pro výpočet heliocentrických souřadnic Země 105
A.2 Data pro výpočet nutace 110
B.1 Popis jednotlivých svorkovnic na navržené desce plošných spojů . . . 113

ÚVOD

Jedním z cílů současné energetické politiky Evropské unie (resp. ENTSO-E) je v souladu s myšlenkou trvale udržitelného rozvoje dosažení nízkouhlíkového hospodářství a s tím související snižování spotřeby fosilních paliv. Obnovitelné zdroje energie (OZE) představují možný směr, kterým se energetika v budoucnu může vydat, a je tedy pochopitelné, že se v poslední době těší zvýšené pozornosti.

Sektor obnovitelných zdrojů energie je v historii lidstva nejstarším energetickým sektorem vůbec – již pračlověk spaloval biomasu, čímž získával teplo a světlo. V oblasti výroby elektrické energie OZE rovněž nejsou žádným nováčkem – vodní elektrárny se užívaly v počátcích systematické výroby elektřiny.

Využití slunečního záření pro výrobu elektřiny je trendem o poznání mladším, solární elektrárny se ve větším měřítku používají přibližně od osmdesátých let dvacátého století. Od té doby je patrný jejich poměrně intenzivní výzkum, vývoj i nárůst výroby.

Vzhledem k neustále rostoucí spotřebě elektrické energie jsou i solární elektrárny vystaveny silnému tlaku na zvyšování účinnosti. Toho lze docílit mimo jiné naváděním solárního zařízení za sluncem v průběhu dne pomocí trackeru (sledovače).

Účelem této práce je navrhnout dvouosý solární tracker, který bude sloužit pro automatické natáčení solárních koncentrátorových systémů nebo fotovoltaických panelů.

V první části práce jsou zmapovány a stručně charakterizovány technologie solárních kolektorů, solárních koncentrátorů a fotovoltaických panelů. Jsou zde rovněž rozebrány současně využívané způsoby sledování slunce na obloze.

Druhá část práce předkládá algoritmus výpočtu polohy slunce a zběžně vysvětluje potřebné základy nebeské mechaniky a popisuje používané souřadnicové systémy.

Třetí část se věnuje samotnému návrhu předmětného trackeru – uvádí možné pohonné jednotky a způsob jejich řízení, rozvádí způsoby snímání skutečné polohy solárního zařízení, dokumentuje softwarovou část trackeru a na závěr představuje návrh pohyblivého stojanu.

Čtvrtá a poslední část se zaměřuje na konstrukci trackeru a pohyblivého stojanu a ověření funkčnosti v laboratorních podmínkách i na reálném zařízení. Zároveň obsahuje analýzu ekonomické výhodnosti a cenové srovnání s komerčně dostupnými systémy.

1 TECHNOLOGIE AKTIVNÍHO VYUŽITÍ SLUNEČNÍ ENERGIE

Slunce je zdrojem zářivé energie plného spektra o vlnových délkách od 200 nm do 3 µm, jeho zářivý výkon je přibližně 4×10^{26} W, z čehož na Zemi dopadá pouze nepatrné množství, přesto sluneční energie představuje naprostou většinu energie, která se na Zemi využívá [1][2]. Teoretické množství energie dopadající na jednotkovou plochu mimo zemskou atmosféru za jednotku času je přibližně 1361 $\mathrm{W} \cdot \mathrm{m}^{-2}$ [3].

Tuto energii lze aktivně využít v zásadě dvěma způsoby – přeměnou na teplo a přímou přeměnou na elektřinu.

1.1 Výroba tepla

Teplo získané ze slunečního záření lze využít přímo nebo jej lze přeměnit na elektrickou energii. K tomu se využívá Seebeckova jevu (v obvodu ze dvou různých materiálů vzniká elektrický proud, mají-li jejich spoje rozdílnou teplotu [1]), parních turbín a dalších způsobů. Pro přeměnu sluneční energie na energii tepelnou se využívají solární kolektory a solární koncentrátory.

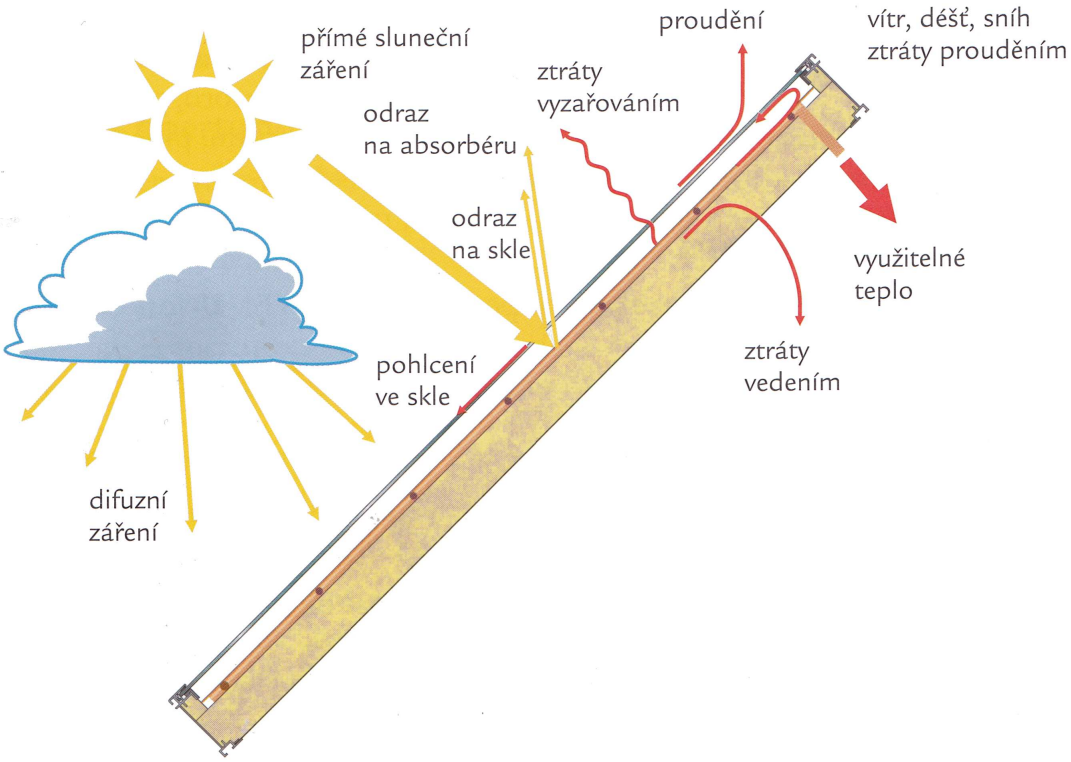
1.1.1 Solární kolektory

Výraz „kolektor“ pochází z latinského *collegere*, což znamená *sbírat*; solární kolektor především „sbírá“ sluneční paprsky. Ústředním prvkem každého kolektoru je absorbér (trubicový nebo deskový), na jehož povrchu se energie dopadajícího záření přeměňuje na teplo, které se dále předává teplonosnému médiu.

Teplonosným médiem je kapalina nebo plyn, zpravidla se jedná o vodu, vzduch, olej nebo sůl [4]. Tomuto médiu není nikdy předána veškerá energie, neboť každý kolektor vykazuje ztráty. Optické ztráty vznikají odražením a pohlcením části slunečního záření (jak na absorbéru, tak na případném skle kolektoru), k tepelným ztrátám dochází vedením, prouděním a vyzařováním, jak ukazuje obr. 1.1.

Dle konstrukce a možností využití lze kolektory rozdělit na bazénové, ploché, vzduchové a vakuové.

Bazénové solární kolektory jsou nejjednodušším typem těchto zařízení. Obsahují pouze absorbér černé barvy, který nemá žádnou další mechanickou ochranu (proto je přesnější označovat tato zařízení jako bazénové absorbéry). Účinnost těchto kolektorů je ve srovnání s jinými typy mírně vyšší při malých rozdílech teplot absorbéru a okolí, neboť se zde oproti jiným kolektorům nenachází skleněná deska, odpadají tedy ztráty energie odrazem a pohlcením ve skle. Účinnost je naopak nižší při rozdílech



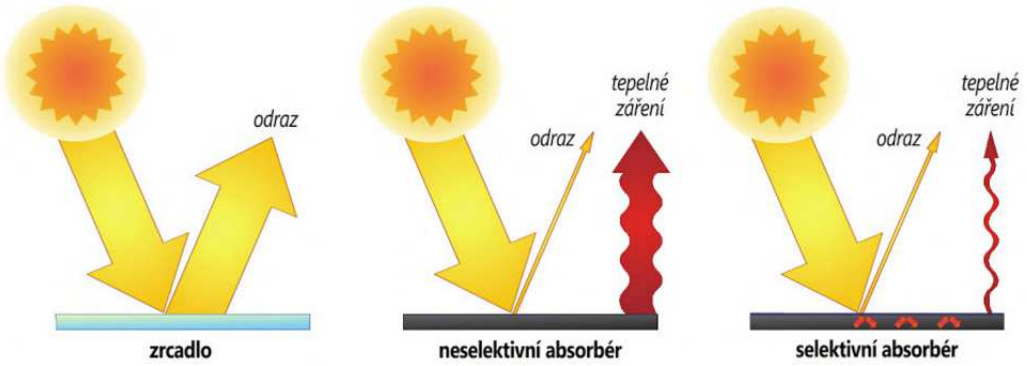
Obr. 1.1: Optické a tepelné ztráty solárního kolektoru

Zdroj: [5]

teplot větších než 10 °C. Mezi materiály používané pro výrobu bazénových absorbérů lze zařadit polyetylén (PE) nebo polypropylén (PP), které odolávají působení UV záření a chlóru [6][4].

Ploché kapalinové kolektory vykazují nižší tepelné ztráty díky využití přední skleněné desky, která však zároveň zvyšuje ztráty optické. Teplonosným médiem je zpravidla voda s příměsí nemrznoucí směsi. Vzhledem ke skutečnosti, že teplota absorbéru může dosáhnout i 200 °C, není již plastové potrubí vhodné. Většinou se tedy využívají měděné trubky, které však nemají přirozeně černý povrch, proto se na ně zpočátku nanášela černá vrstva. Ta ovšem při zvýšení své teploty emituje tepelné záření do okolí, čímž vznikají ztráty vyzařováním. Z toho důvodu se v dnešní době využívá selektivní absorpční vrstva, která absorbuje nejen přímé, ale i difuzní sluneční záření, a minimalizuje ztráty vyzařováním (princip vysvětluje obr. 1.2). Využití tyto kolektory naleznou nejčastěji u ohřevu TUV, méně často (a zpravidla jen se selektivním povrstvením) pak u vytápění [6][4].

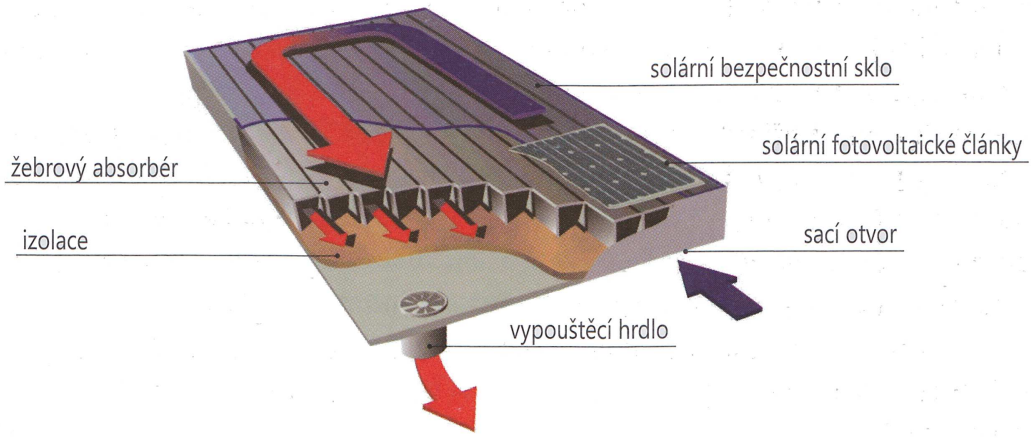
Vzduchové kolektory se od plochých kolektorů s vodním průtokem principiálně nijak neliší, pouze se namísto vody využívá vzduch, a je tedy třeba větších absorpčních



Obr. 1.2: Princip selektivní absorpční vrstvy

Zdroj: [4]

průřezů. Řez vzduchovým kolektorem s žebrovým absorbérem je znázorněn na obr. 1.3. Integrací fotovoltaického panelu lze zajistit energii pro pohon ventilátorů. Takto ohřátý vzduch se využívá víceméně pouze jako podpora vytápění, jakákoliv akumulace tepla je v tomto případě velmi složitá [4].



Obr. 1.3: Řez vzduchovým kolektorem

Zdroj: [4]

Vakuové solární kolektory vznikly zdokonalením plochých kolektorů. Kolektor je tvořen skleněnými vakuovanými trubicemi, uvnitř kterých jsou uzavřeny ploché absorbérové pásy. Uprostřed každého z pásů je tepelná trubice (*heat-pipe*). Vlivem slunečního záření dochází k výparu teplonosného média (zpravidla kapalina obsahující metanol nebo voda při sníženém tlaku) a takto vzniklá pára pak stoupá vzhůru. Na horním konci vystupuje heat-pipe ze skleněné trubice a ústí do kondenzátoru, kde médium předá energii topné vodě (nebo TUV). Tím se ochladí, zkapalní a vrací se do kolektoru. Odstraněním vzduchu z kolektoru bylo dosaženo snížení množství tepelných

ztrát, účinnost je tedy vyšší (a to zejména v případě výraznějších rozdílů teplot kolektoru a okolí), pořizovací cena je však rovněž vyšší. Aby nedocházelo k vniknutí atmosferického vodíku do vakua, využívá se getr – látka, která chemicky váže molekuly vodíku. Vakuové kolektory vyžadují oproti jiným typům menší plochu, pro zajištění správné funkce je ale zapotřebí dodržet při montáži určitý sklon. Nejčastěji jsou využívány pro vytápění, méně často pro ohřev TUV [6][4].

Srovnání energetických zisků plochých kapalinových kolektorů a vakuových kolektorů předkládá tab. 1.1.

Tab. 1.1: Energetické zisky některých solárních kolektorů

Typ kolektoru	Energetický zisk ($kWh \cdot m^{-2}$ za rok)
Kapalinový bez selektivního povrstvení	250–400
Kapalinový se selektivním povrstvením	320–530
Vakuový	400–890

Zdroj: [6]

Pro úplnost je třeba uvést ještě **komínové solární systémy**, které tvoří jakousi zvláštní kategorii. Jsou tvořeny velkou (stovky až tisíce metrů v průměru) rovnou plochou pokrytou skleněnou nebo plastovou střechou, která představuje kolektorové pole. Uprostřed tohoto pole stojí komín (až tisíc metrů vysoký o průměru až dvou set metrů), ke kterému se kolektorové pole pozvolna zvedá. Sluncem ohřátý vzduch pod střechou proudí směrem ke komínu a následně stoupá vysokou rychlostí komínem vzhůru, kde pohání větrné turbíny. Půda pod střechou může sloužit pro akumulaci tepla (v půdě je položeno potrubí), případně mohou být pod střechou instalovány i fotovoltaické panely. Prostor pod střechou může být rovněž využit pro zemědělské účely (jako skleník) [7][4].

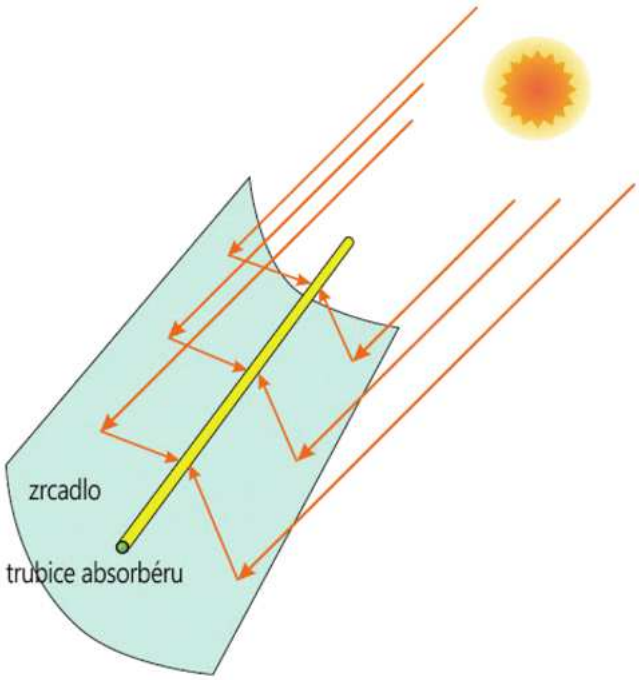
1.1.2 Solární koncentrátory

Solární koncentrátory využívají princip odrazu dopadajícího slunečního záření. Pomocí zrcadla nebo soustavy zrcadel koncentrují dopadající paprsky na absorbér (potrubí nebo nádoba), případně Stirlingův motor, čímž zvyšují intenzitu záření. Lze rozeznat dvě provedení koncentrátorů: jednoosé a dvouosé.

Jednoosý (liniový) systém využívá ohniskovou osu, do které koncentruje záření. Používají se obvykle parabolické žlaby (obr. 1.4) nebo Fresnelovy koncentrátory (obr. 1.5).

Dvouosý systém soustřeďuje paprsky do jediného bodu – ohniska. K tomu se využívají především vydutá zrcadla (obr. 1.6) a systémy s centrální věží (obr. 1.7).

Parabolické žlaby představují nejvíce vyvinutou technologii užívanou v rámci solárních koncentrátorů. Využívají žlabově tvarovaná parabolická zrcadla, která koncentrují světlo na trubicový absorbér umístěný do ohniskové osy. Kvůli sledování polohy slunce se žlaby často otáčí kolem své podélné osy. Trubice absorbéru je uložena do vakuovaného skleněného pouzdra a opatřena selektivní vrstvou za účelem snížení tepelných ztrát. Koncentrační faktor (poměr plochy odrazné vrstvy k ploše absorbéru) žlabových systémů je až 80. Skleněným pouzdrem lze opatřit celý systém, což omezuje ztráty tepla vlivem proudění větru a snižuje množství usazeného prachu. Teplonosným médiem je nejčastěji olej, který se zářením ohřívá na teplotu až 400 ℃ [8][4][1].

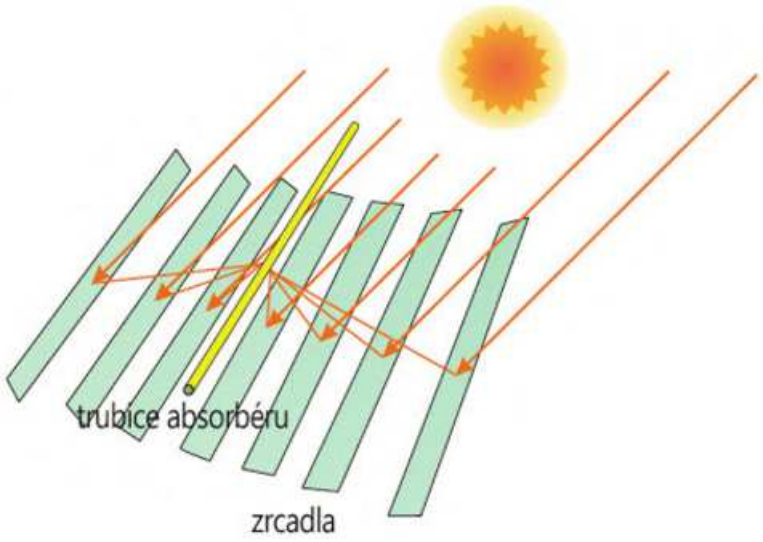


Obr. 1.4: Parabolický žlabový koncentrátor

Zdroj: [4]

Fresnelovy koncentrátory rovněž soustřeďují paprsky na trubicový absorbér, namísto jednoho žlabu je na jeden absorbér však využito několika dlouhých tenkých reflektorů. Ty jsou tvořeny Fresnelovými čočkami (odtud pojmenování Fresnelovy koncentrátory), což snižuje množství potřebného materiálu. Reflektory jsou nejčastěji orientovány v severojižním směru a stejně jako žlaby se obvykle otáčí kolem své podélné osy. Koncentrační faktor je až 30. Jako teplonosné médium se využívá opět olej. Jestliže se namísto jednoho absorbéru použije absorbérů více (zpravidla dva),

pak se jedná o tzv. kompaktní Fresnelův koncentrátor, jehož výhodami jsou především minimalizace zastínění jednotlivých čoček a menší potřebný prostor [9][10][11].

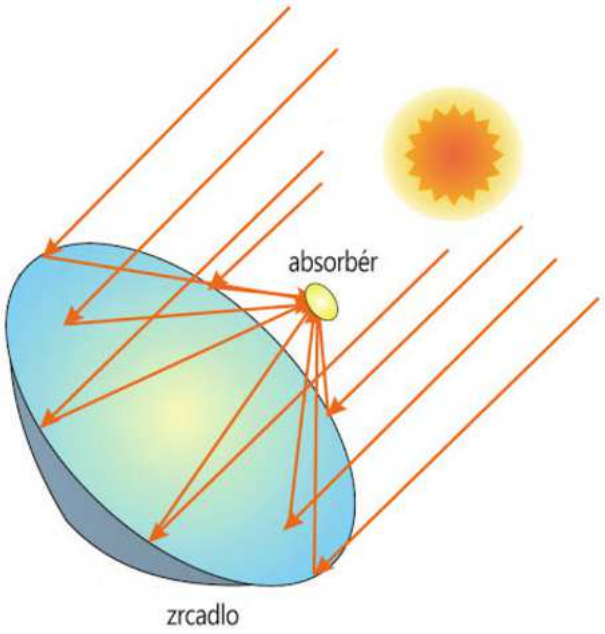


Obr. 1.5: Fresnelův koncentrátor

Zdroj: [4]

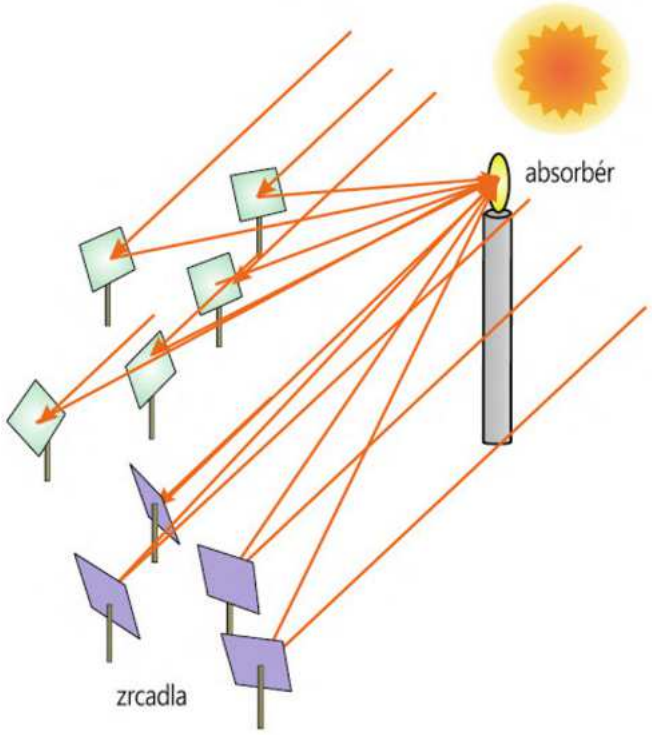
Vydutá zrcadla (neboli diskové koncentrátory) jsou tvořena konkávními parabolickými zrcadly, v jejichž ohnisku (bodovém) je umístěn absorbér nebo Stirlingův motor (pak se systém označuje jako *Dish-Stirling*). Zrcadly, jejichž odrazivost je až 94 %, jsou do ohniska odraženy veškeré paprsky dopadající na povrch paraboly, díky tomu lze dosáhnout koncentračního faktoru až 4600. Pro udržení takto vysokého faktoru je zapotřebí, aby se parabola mohla otáčet ve dvou osách. Koncentrátory se vyrábí ze sklolaminátu a jsou vyztuženy epoxidovou pryskyřicí [12][13].

Systém s centrální věží (případně solární věžový systém) využívá ke směrování slunečních paprsků stovky nebo i tisíce rovinných zrcadel (tzv. *heliostatů*), jež jsou nasměrována na centrální věž. Absorbér je umístěn na vrcholu věže a díky koncentračnímu faktoru až 2000 se ohřívá na teploty přes 1000 ℃. Teplonosným médiem je zde vzduch nebo tekutá sůl. Zrcadla se otáčí ve dvou osách a každé je řízeno samostatně, přičemž se vyžaduje přesnost navádění v řádu jednotek úhlových minut. Zastavěná plocha se zpravidla pohybuje mezi 150 a 320 hektary [14][1].



Obr. 1.6: Parabolický diskový koncentrátor

Zdroj: [4]



Obr. 1.7: Koncentrátorový systém s centrální věží

Zdroj: [4]

1.2 Výroba elektrické energie

Tepelná energie získaná pomocí solárních koncentrátorů se nejčastěji prostřednictvím turbíny a Rankin-Clausiova cyklu mění na energii elektrickou. Tu však lze získávat i přímou přeměnou. Pro tyto účely se využívají fotovoltaické články, které produkují elektrickou energii na principu fotoelektrického jevu.

Nejběžnějším typem je křemíkový fotočlánek, který se skládá z polovodiče typu P a polovodiče typu N, čímž je zabezpečen potenciálový spád. V oblasti přechodu P-N (oblasti prostorového náboje) vzniká elektrické pole, které je schopno odvádět elektrony. Elektrony vznikají oddělením z páru elektron – díra při dopadu fotonu s dostatečnou energií. Kdyby nedošlo k separaci elektronů vnitřním elektrickým polem P-N přechodu, došlo by k okamžité rekombinaci. Ne všechny dopadající fotony jsou ale pohlceny [1].

Napětí jednoho fotovoltaického článku je poměrně malé – pohybuje se mezi 0,6 a 0,7 V [4]. Proto se využívá sériového spojení mnoha takových článků, které pak tvoří fotovoltaický panel (modul). Pro omezení nepříznivého vlivu nefunkčních článků jsou do obvodu přidávány by-passové diody.

Fotovoltaické panely mohou být osvětleny i pomocí solárního koncentrátoru. V takovém případě může být plocha článku menší úměrně ke koncentračnímu faktoru, čímž je umožněno využívat pro výrobu článků i materiály, které by standardně byly ekonomicky neefektivní, a tedy zvýšit účinnost. Problémem tohoto řešení (a v menší míře i klasických fotovoltaických panelů) je však chlazení, které si musí poradit s velkým množstvím tepla vznikajícího při výrobě elektrické energie.

Dalším využitím koncentrovaného slunečního záření pro fotovoltaiku je technologie FLATCON, která pro soustřeďování paprsků využívá Fresnelovy čočky. Koncentrované záření dopadá na článek o ploše několika mm^2, jenž je na zadní straně opatřen měděnou plochou, která odvádí vznikající teplo. Se systémem FLATCON (znázorněn na obr. 1.8) byla dosažena maximální účinnost 44,7 % (přičemž konvenční technologie dosahují účinnosti 20–25 %) [15].

1.3 Sledování polohy slunce

Energetický zisk výše uvedených zařízení (zejména fotovoltaických panelů a solárních koncentrátorů) lze v některých případech značně zvýšit použitím trackeru – naváděcího systému, který zajišťuje, aby dané zařízení v průběhu dne kopírovalo dráhu slunce. Oproti nepohyblivému solárnímu zařízení lze při využití jednoosého trackingu dosáhnout o 20–35 % vyššího zisku, dvouosý tracker pak dokáže zisk zvýšit o 25–40 % v závislosti na okolnostech konkrétní aplikace [16]. Ve velmi specifických podmínkách lze dokonce dosáhnout zvýšení až o 60 % [17].

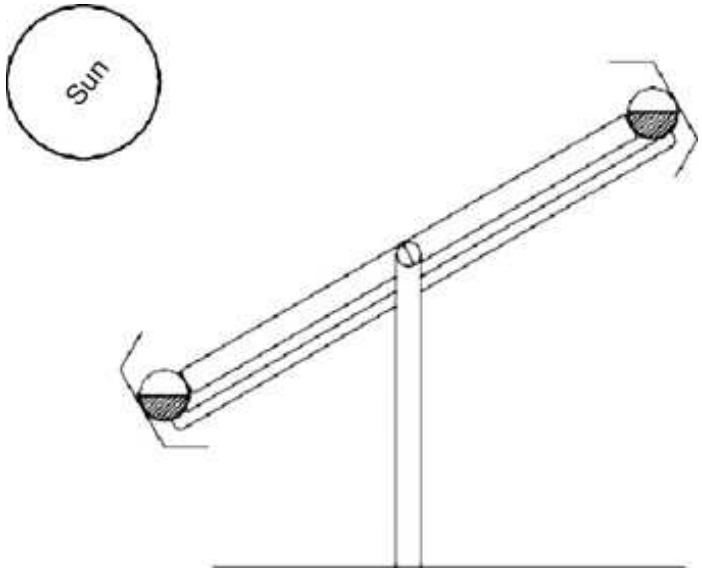
Obr. 1.8: Fotovoltaický panel s koncentrátorovým modulem FLATCON
Zdroj: [15]

Není přitom nutné, aby tracker zařízení nasměroval přímo ke slunci. Při odchylce 10° dosahuje výkon stále 98,5 % svého maxima [16]. Ideální tracker by však měl být schopen namířit solární zařízení přímo ke slunci, provádět neustálou korekci směru během dne a zároveň reagovat na změnu polohy slunce vlivem ročních období.

Sledovače slunce se zpravidla dělí na pasivní (mechanické) a aktivní (elektronické).

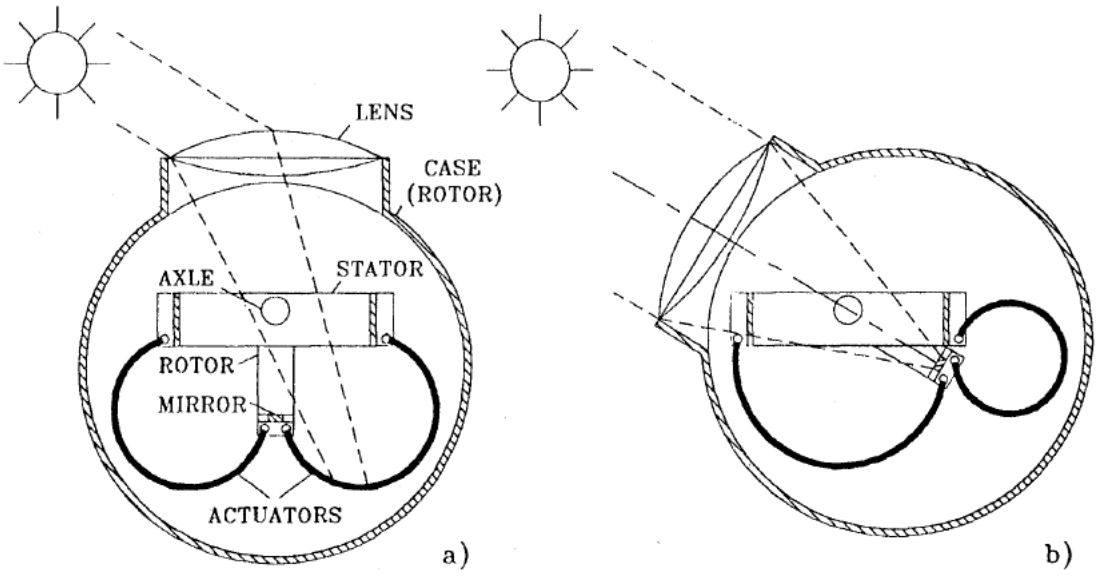
1.3.1 Pasivní trackery

Princip fungování pasivních trackerů je založen na teplotní roztažnosti některých látek (nejčastěji freonů). Tracker je obvykle jednoosý a na jeho koncích jsou umístěny vzájemně propojené nádrže s pracovní látkou. Každá nádrž je opatřena stínítkem, které vrhá stín v závislosti na úhlu natočení. Rovnováhy je dosaženo v případě rovnoměrného osvětlení obou nádrží. Je-li osvětlení jednotlivých nádrží různé, pracovní látka se v jedné z nich ohřeje a začne se odpařovat do druhé nádrže. Tímto způsobem dojde k opětovnému dosažení rovnováhy. Tracker s kapalinovým vyvažováním je znázorněn na obr. 1.9.

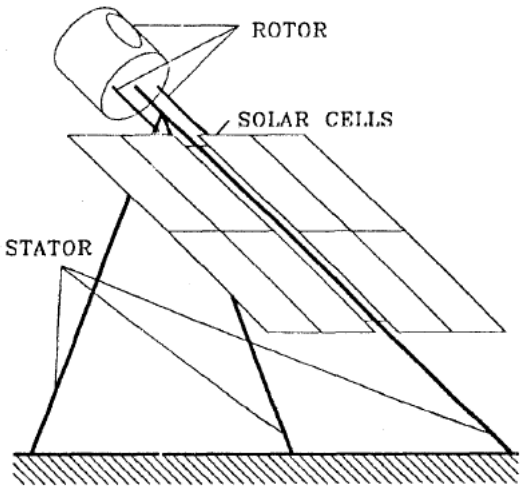


Obr. 1.9: Schematické znázornění pasivního trackeru s kapalinovým vyvažováním
Zdroj: [18]

Další možností je využití bimetalů nebo slitin s tvarovou pamětí (SMA). Slitina se dokáže chovat jako tepelný motor a snadno změní tvar při poklesu teploty pod 70 °C (znázorněno na obr. 1.10, celé zařízení pak na obr. 1.11). Po zahřátí se opět vrátí do původního tvaru. Bylo prokázáno, že účinnost tohoto pohonu je přibližně 2 %, což je o dva řády vyšší účinnost než při použití bimetalu [19].



Obr. 1.10: Princip fungování SMA trackeru před (a) a po (b) východu slunce
Zdroj: [19]



Obr. 1.11: SMA tracker pro fotovoltaický panel

Zdroj: [19]

Pasivní trackery jsou obvykle jednoduchá zařízení výkonově srovnatelná s aktivními trackery, nicméně jejich účinnost je nižší a při nižších teplotách přestávají plnit svoji funkci. Kromě toho vykazují obvykle nízkou až nulovou odolnost proti poryvům větru.

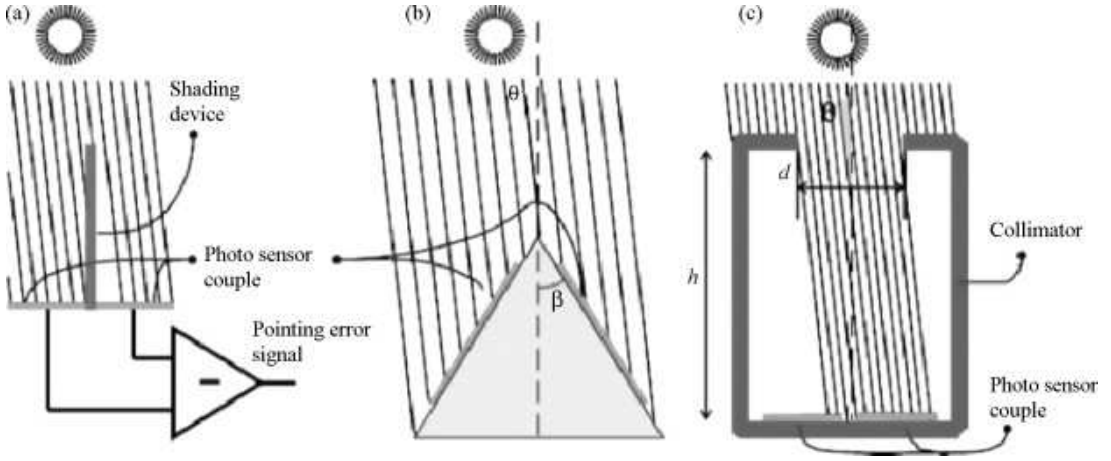
1.3.2 Aktivní trackery

Většinu aktivních trackerů lze dle principu jejich fungování rozdělit do tří skupin na trackery s optickými senzory (které využívají dvou nebo více senzorů, přičemž při stejném osvětlení tyto poskytují stejné hodnoty), trackery s pomocnými oboustrannými fotočlánky (které opět vyhodnocují rozdíl v osvětlení mezi nimi) a trackery řízené pomocí data a času (na základě předem vypočítané polohy slunce na obzoru v daný den a daný čas). Společným úkolem těchto systémů je určení vektoru intenzity slunečního záření

Trackery s optickými senzory

Některé typy těchto tzv. diferenciálních sledovačů předkládá obr. 1.12, nejběžnější je umístění senzorů pod stínící přepážku (a), pro zvýšení citlivosti lze senzory umístit na nakloněné roviny (b) nebo je možné místo stínítka využít kolimátor (c), který nepropustí difuzní záření, čímž se zvyšuje přesnost určení polohy slunce. Poslední jmenovaný typ je pro svou velmi vysokou přesnost často využíván u koncentrátorových fotovoltaických panelů [20].

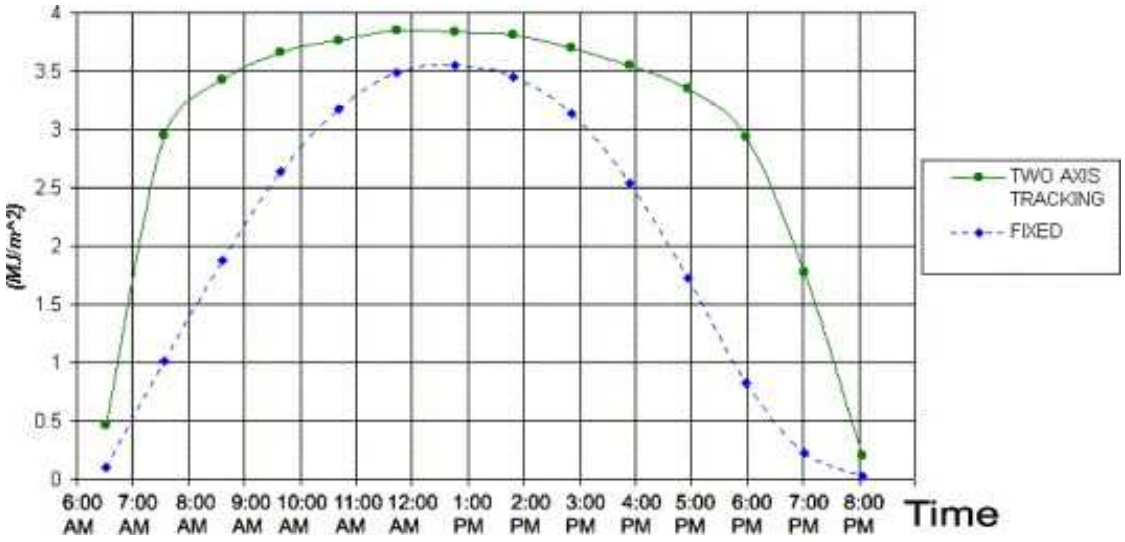
Signály ze senzorů zpracovává PLC, který řídí pohon (resp. pohony). Energie je poskytována ze sítě nebo z akumulátorů (zpravidla solárně dobíjených).



Obr. 1.12: Různá provedení trackeru s optickými senzory

Zdroj: [20]

Abdallah experimentálně ověřoval účinnost tohoto trackeru pomocí dvou pyranometrů – jednoho pohyblivého ovládaného trackerem, jednoho namontovaného nepohyblivě. Hodnoty z pyranometrů byly zaznamenávány každých pět minut a pro intervaly jedné hodiny byly vždy zprůměrovány a přepočteny na hodinový zisk. Experiment probíhal čtyři dny (7., 10., 11. a 12. května 2001) v Renewable Energy Laboratory na univerzitě v jordánském Ammánu [21]. Obrázek 1.13 znázorňuje naměřenou intenzitu slunečního záření dne 7. 5. 2001. Jednotlivé denní hodnoty měření předkládá tab. 1.2.



Obr. 1.13: Naměřená intenzita slunečního záření v průběhu dne 7. 5. 2001

Zdroj: [21]

Tab. 1.2: Naměřená průměrná intenzita slunečního záření

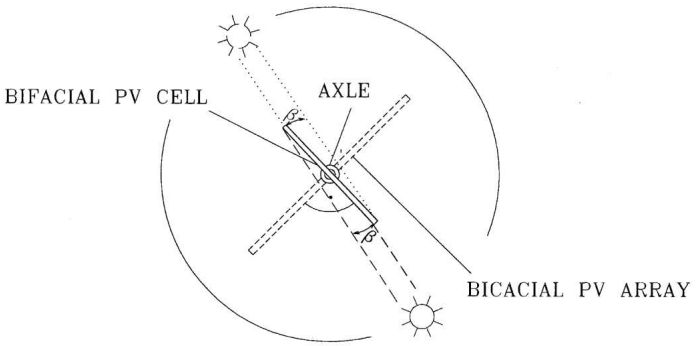
Datum	S použitím trackeru ($MJ \cdot m^{-2}$)	Bez trackeru ($MJ \cdot m^{-2}$)	Zisk (%)
7. 5. 2001	41,266	27,746	48,73
10. 5. 2001[1]	27,490	20,730	32,61
11. 5. 2001	34,534	23,661	45,95
12. 5. 2001	36,057	26,110	38,09
průměr	34,837	24,562	41,34

[1] Nižší intenzita záření dne 10. 5. 2001 je částečně způsobena vyšší oblačností.

Zdroj: [21]

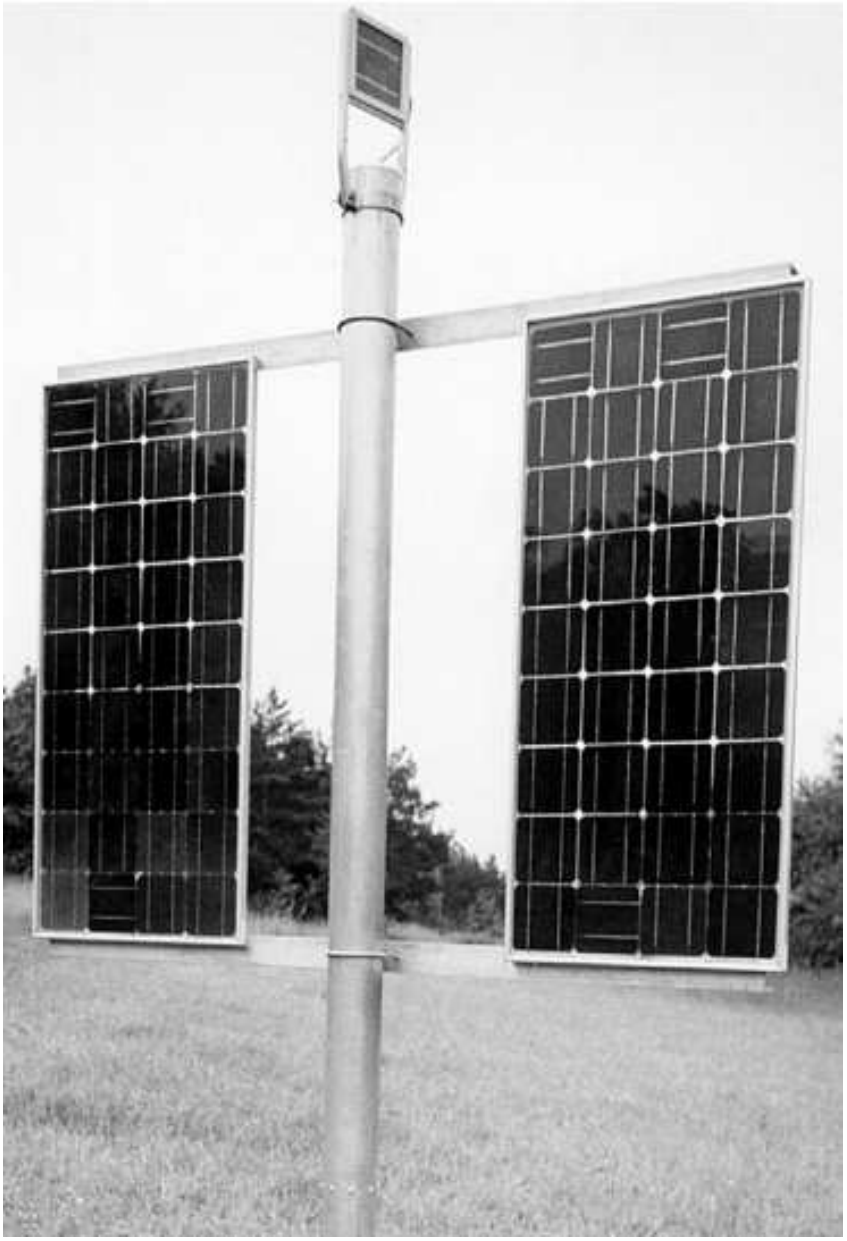
Trackery s oboustrannými fotočlánky

Sledovače s oboustrannými fotočlánky se zpravidla používají pro oboustranné fotovoltaické panely, kterým umožňují pohyb v 360° úhlu. Pomocné fotočlánky o ploše přibližně 1 % celkové aktivní plochy jsou připevněny k otočné hřídeli a připojeny přímo k permanentnímu magnetu DC motoru. Mohou kromě vyhodnocování polohy slunce zároveň dodávat trackeru energii. Základní konstrukční schéma trackeru tohoto typu znázorňuje obr. 1.14. Odstraněním elektroniky nutné pro vyhodnocení signálu a ovládání pohonu bylo eliminováno maximum potenciálně poruchových prvků. V tomto případě zároveň není třeba externí zdroj napětí, jedná se tedy o velmi jednoduché zařízení, díky čemuž se tento typ trackeru využívá na zemi i ve vesmíru. Fotografie zařízení opatřeného tímto trackerem je na obr. 1.15. Měření prokázala, že odchylka dosahuje hodnot ±5° a získaná energie dosahuje 95 % hodnoty dosažitelné s ideálním trackerem [22].



Obr. 1.14: Schéma trackeru s oboustrannými fotočlánky (půdorysné zobrazení)

Zdroj: [22]



Obr. 1.15: Fotovoltaické zařízení s trackerem s oboustrannými fotočlánky
Zdroj: [22]

Časově řízené trackery

V případě posledního jmenovaného typu aktivních trackerů je natočení určeno předem vypočítanou polohou slunce pro konkrétní okamžik na konkrétním místě na zemi. Systém tedy pracuje bez zpětné vazby. Výhodami tohoto typu jsou vysoká přesnost (polohu slunce lze vypočítat s chybou v řádu úhlových vteřin), která je zpravidla omezena jen možnostmi pohonu, a spolehlivá funkce i při dočasném zastínění nebo nízké úrovni osvětlení. Nevýhodami jsou nutnost poskytování informace o datu a

času, potřeba přesného určení výchozí (nebo okamžité) orientace a samotný výpočet úhlu natočení, který se zpravidla provádí předem pomocí počítače, do trackeru se pak jen nahrají získaná data. Má-li tracker počítat polohu samostatně, klade to vyšší nároky na jeho hardware i software.

Určit orientaci zařízení je možné pomocí pevně stanoveného referenčního (výchozího) bodu, relativní pozici pak zjišťují enkodéry, případně lze pro pohon využít krokové motory a udržovat informaci o poloze pomocí krokování. V některých případech se využívá také kombinace magnetometru, gyroskopu a akcelerometru. Je-li zařízení dostatečně rozměrné, nabízí se možnost použít pro určení orientace několik GPS modulů. V praxi se však velmi často využívá kombinace časového řízení a senzoru intenzity slunečního záření.

2 VÝPOČET POLOHY SLUNCE

2.1 Souřadnicové systémy

Země obíhá kolem Slunce po eliptické dráze – rovina této dráhy je tzv. ekliptika. Perioda tohoto oběhu definuje jeden rok. Relativní poloha Slunce vůči Zemi je obvykle vyjadřována pomocí nebeské sféry. Průsečnice nebeské sféry s rovinou proloženou zemským rovníkem se nazývá nebeský rovník; průsečíky nebeské sféry se zemskou osou se nazývají nebeské póly (severní a jižní). Obíhání Země kolem Slunce je zpravidla vyjadřováno jako zdánlivý pohyb Slunce po elipse, která je vůči nebeskému rovníku nakloněna o 23,45° (neboli odklon zemské osy od kolmice na rovinu oběhu je 23,45°)[1]. Úhel, který svírá spojnice středů Slunce a Země s průmětem zemského středu do roviny rovníku, je nazýván deklinace slunce. Deklinace nabývá hodnoty 0° při jarní (20./21. března) a podzimní (22./23. září) rovnodennosti [23][24].

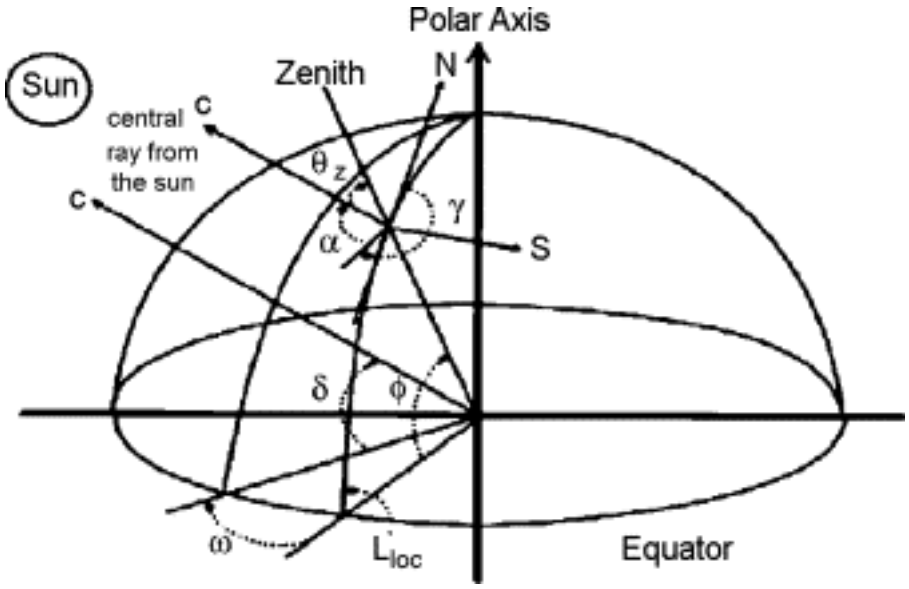
Země současně rotuje kolem své osy, a sice s periodou jednoho dne. Tento pohyb je popsán rotací nebeské sféry kolem zemské osy, přičemž okamžitá poloha slunce je popsána hodinovým úhlem ω (úhlem mezi poledníkem procházejícím sluncem a místním poledníkem). Hodinový úhel je nulový ve sluneční (pravé) poledne a směrem k východu se zvyšuje. Pro pozorovatele na Zemi v místě se zeměpisnou šířkou φ je vhodný souřadnicový systém určen kolmicí na horizontální rovinu daného místa, která protíná nebeskou sféru ve dvou bodech – zenitu (nadhlavníku; Z) a nadiru (podnožníku; N) – a svírá se zemskou osou úhel φ, jak ilustruje obr. 2.1. Vodorovná rovina procházející místem pozorování protíná nebeskou sféru v kružnici zvané horizont (obzorník) [23][25].

Zeměpisná šířka (φ) bodu na povrchu Země vyjadřuje úhel, který svírá spojnice středu Země a tohoto bodu s rovinou rovníku. Zemská osa protíná povrch Země v 90° (severní pól) a $-90°$ (jižní pól) šířky. Kterékoliv místo na Zemi lze určit pomocí zeměpisné délky a šířky.

Výška slunce α je definována jako úhel, který svírá sluneční paprsek s horizontální rovinou proloženou místem pozorování, jak je znázorněno na obr. 2.1. Namísto výšky slunce lze užít i zenitovou vzdálenost slunce (ϕ_z), což je vertikální úhel mezi slunečním paprskem a normálovou přímkou v daném bodě; tyto úhly jsou doplňkové. Azimut slunce γ_s je úhel svíraný kolmým průmětem slunečního paprsku do roviny horizontu s jižním směrem. Azimut slunce nabývá kladných hodnot, nachází-li se slunce západně od jižního směru [27][28].

Deklinace a hodinový úhel tvoří rovníkové souřadnice, zatímco výška a azimut slunce tvoří obzorníkové (horizontové) souřadnice slunce. Obě skupiny souřadnic

[1] Sklon zemské osy se ve skutečnosti pohybuje mezi 22,1° a 24,5°. Tuto fluktuaci způsobuje nutace – kývavý pohyb osy způsobený postupnou změnou slapové síly.



Obr. 2.1: Schematické znázornění slunečních úhlů

Zdroj: [26]

využívají topocentrický sférický systém (středem, kolem kterého se zdánlivě otáčí Slunce, je pozorovatel na povrchu Země), který je oproti systému heliocentrickému (počátek je ve středu Slunce) nebo geocentrickému (počátek je ve středu Země) pro pozorovatele na Zemi vhodnější [28].

2.2 Algoritmus výpočtu

Pro výpočet se nabízí mnoho algoritmů. V solární energetice se velmi často využívá Solar Position Algorithm (SPA) [29]. Jedná se o algoritmus zpracovaný pod záštitou americké Národní laboratoře pro obnovitelné zdroje a oproti jiným postupům pro vyjádření polohy Slunce se vyznačuje zejména velmi malou hodnotou nejistoty (±0,0003°). Zároveň dokáže zohlednit vliv atmosféry na lom světla.

Vstupními parametry pro výpočet jsou datum a čas, zeměpisné souřadnice pozorovatele (bodu) na zemi, nadmořská výška, průměrný roční tlak, průměrná roční teplota, korekční činitel atmosferické refrakce a dvě proměnné vyjadřující časovou diferenci mezi časy UT a UTC a mezi časem TT a časem zemské rotace.

Výstupními hodnotami jsou zenit, azimut, úhel dopadu slunečních paprsků (v případě nakloněné roviny) a časy východu a západu slunce a pravého poledne.

2.2.1 Výpočet juliánského a juliánského efemérního data

Juliánský den

$$JD = \lfloor 365,25 \cdot (Y + 4716) \rfloor + \lfloor 30,6001 \cdot (M + 1) \rfloor + D + B - 1524,5 \qquad (2.1)$$

kde:
JD je juliánský den
Y je rok
M je měsíc v roce. Pro $M > 2$ se Y a M nemění, pro $M \leq 2$ platí $Y = Y - 1$ a $M = M + 12$.
D je den v měsíci včetně desetinné reprezentace času (pro druhý den v měsíci v čase 12.30:30 UT platí $D = 2,521\,180\,556$)
B je v případě gregoriánského kalendáře rovno $(2 - \lfloor Y/100 \rfloor + \lfloor \lfloor Y/100 \rfloor /4 \rfloor)$

Juliánský efemérní den

$$JDE = JD + \frac{\Delta T}{86\,400} \qquad (2.2)$$

kde:
JDE je juliánský efemérní den
JD je juliánský den
ΔT je rozdíl mezi časem zemské rotace a časem TT určený pozorováním

Juliánské století a juliánské efemérní století

$$JC = \frac{JD - 2\,451\,545}{36\,525} \qquad (2.3)$$

$$JCE = \frac{JDE - 2451545}{36525} \qquad (2.4)$$

kde:
JC je juliánské století
JD je juliánský den
JCE je juliánské efemérní století
JDE je juliánský efemérní den

Juliánské efemérní tisíciletí

$$JME = \frac{JCE}{10} \qquad (2.5)$$

kde:
JME je juliánské efemérní tisíciletí
JCE je juliánské efemérní století

2.2.2 Výpočet heliocentrických souřadnic Země

Heliocentrická délka

$$L_{0i} = A_i \cdot \cos\left(B_i + C_i \cdot JME\right) \tag{2.6}$$

kde:

L_{0i}	je v radiánech
i	značí pořadí iterace dle tabulky A.1 (v příloze A)
A_i, B_i, C_i	jsou hodnoty i-té iterace; A, B a C jsou sloupce tabulky A.1 (v příloze A)
JME	je juliánské efemérní tisíciletí

$$L_0 = \sum_{i=0}^{n} L_{0i} \tag{2.7}$$

kde:

L_0	je v radiánech
n	je počet řádků příslušného výrazu v tabulce A.1 (v příloze A)

Výrazy L_1, L_2, L_3, L_4 a L_5 se vypočítají podle vztahů 2.6 a 2.7 analogicky. Heliocentrická délka Země je potom:

$$L = \frac{L_0 + L_1 \cdot JME + L_2 \cdot JME^2 + L_3 \cdot JME^3 + L_4 \cdot JME^4 + L_5 \cdot JME^5}{10^8} \tag{2.8}$$

kde:

L	je heliocentrická délka v radiánech
L_0	je v radiánech
L_1	je v radiánech
JME	je juliánské efemérní tisíciletí
L_2	je v radiánech
L_3	je v radiánech
L_4	je v radiánech
L_5	je v radiánech

Heliocentrická šířka B se spočítá (v radiánech) za využití vztahů 2.6 až 2.8, přičemž výraz L se ve všech rovnicích nahradí výrazem B s tím, že hodnoty B_2 až B_5 jsou rovny nule.

Radius R se spočítá (v astronomických jednotkách) za využití vztahů 2.6 až 2.8, přičemž výraz L se ve všech rovnicích nahradí výrazem R s tím, že hodnota R_5 je rovna nule.

2.2.3 Výpočet geocentrických souřadnic

Geocentrická délka

$$\Theta = L + 180° \tag{2.9}$$

kde:
Θ je geocentrická délka ve stupních
L je heliocentrická délka převedená na stupně

Geocentrická šířka

$$\beta = -B \tag{2.10}$$

kde:
β je geocentrická šířka ve stupních
B je heliocentrická šířka převedená na stupně

2.2.4 Výpočet nutace délky a sklonu osy

Střední elongace měsíce

$$X_0 = 297,850\,36 + 445\,267,111\,48 \cdot JCE - 0,001\,914\,2 \cdot JCE^2 + \frac{JCE^3}{189\,474} \tag{2.11}$$

kde:
X_0 je střední elongace měsíce ve stupních
JCE je juliánské efemérní století

Střední anomálie slunce

$$X_1 = 357,527\,72 + 35\,999,050\,34 \cdot JCE - 0,000\,160\,3 \cdot JCE^2 - \frac{JCE^3}{300\,000} \tag{2.12}$$

kde:
X_1 je střední anomálie slunce ve stupních
JCE je juliánské efemérní století

Střední anomálie měsíce

$$X_2 = 134,962\,98 + 477\,198,867\,398 \cdot JCE + 0,008\,697\,2 \cdot JCE^2 + \frac{JCE^3}{56\,250} \tag{2.13}$$

kde:
X_2 je střední anomálie měsíce ve stupních
JCE je juliánské efemérní století

Argument šířky měsíce

$$X_3 = 93,271\,91 + 483\,202,017\,538 \cdot JCE - 0,003\,682\,5 \cdot JCE^2 + \frac{JCE^3}{327\,270} \quad (2.14)$$

kde:

X_3 je argument šířky měsíce ve stupních

JCE je juliánské efemérní století

Délka vzestupného uzlu dráhy měsíce na ekliptice

$$X_4 = 125,044\,52 - 1934,136\,261 \cdot JCE + 0,002\,070\,8 \cdot JCE^2 + \frac{JCE^3}{450\,000} \quad (2.15)$$

kde:

X_4 je délka vzestupného uzlu dráhy měsíce na ekliptice ve stupních

JCE je juliánské efemérní století

Pro výpočet nutace délky a sklonu osy je třeba vyčíslit výrazy

$$\Delta\Psi_i = (a_i + b_i \cdot JCE) \cdot \sin\left(\sum_{j=0}^{4} X_j \cdot Y_{i,j}\right) \quad (2.16)$$

a

$$\Delta\epsilon_i = (c_i + d_i \cdot JCE) \cdot \cos\left(\sum_{j=0}^{4} X_j \cdot Y_{i,j}\right) \quad (2.17)$$

kde:

$\Delta\Psi_i$ a $\Delta\epsilon_i$ jsou v desetitisícinách úhlové vteřiny

i značí pořadí iterace dle tabulky A.2 (v příloze A)

a_i, b_i, c_i, d_i jsou hodnoty i-té iterace; a, b, c a d jsou sloupce tabulky A.2 (v příloze A)

JCE je juliánské efemérní století

X_j značí výrazy vypočítané pomocí vztahů 2.11 až 2.15

$Y_{i,j}$ je hodnota i-té iterace v j-tém sloupci tabulky A.2 (v příloze A)

Nutace délky

$$\Delta\Psi = \frac{\sum_{i=0}^{63} \Delta\Psi_i}{36\,000\,000} \quad (2.18)$$

kde:

$\Delta\Psi$ je nutace délky ve stupních

$\Delta\Psi_i$ je v desetitisícinách úhlové vteřiny

Nutace sklonu osy

$$\Delta\epsilon = \frac{\sum_{i=0}^{63} \Delta\epsilon_i}{36\,000\,000} \tag{2.19}$$

kde:
$\Delta\epsilon$ je nutace sklonu osy ve stupních
$\Delta\epsilon_i$ je v desetitisícinách úhlové vteřiny

2.2.5 Výpočet skutečného sklonu dráhy k ekliptice

Střední sklon dráhy k ekliptice

$$\begin{aligned}\epsilon_0 = {} & 84\,381,448 - 4680,93 \cdot U - 1,55 \cdot U^2 + 1999,25 \cdot U^3 - \\ & - 51,38 \cdot U^4 - 249,67 \cdot U^5 - 39,05 \cdot U^6 + 7,12 \cdot U^7 + \\ & + 27,87 \cdot U^8 + 5,79 \cdot U^9 + 2,45 \cdot U^{10}\end{aligned} \tag{2.20}$$

kde:
ϵ_0 je střední sklon dráhy k ekliptice v úhlových vteřinách
$U = \frac{JME}{10}$

Skutečný sklon dráhy k ekliptice

$$\epsilon = \frac{\epsilon_0}{3600} + \Delta\epsilon \tag{2.21}$$

kde:
ϵ je skutečný sklon dráhy k ekliptice v úhlových vteřinách
ϵ_0 je střední sklon dráhy k ekliptice v úhlových vteřinách
$\Delta\epsilon$ je nutace sklonu osy ve stupních

2.2.6 Výpočet korekce aberace světla

$$\Delta\tau = -\frac{20,4898}{3600 \cdot R} \tag{2.22}$$

kde:
$\Delta\tau$ je korekce aberace světla ve stupních
R je radius v astronomických jednotkách

2.2.7 Výpočet zdánlivé délky slunce

$$\lambda = \Theta + \Delta\Psi + \Delta\tau \tag{2.23}$$

kde:

λ je zdánlivá délka slunce ve stupních

Θ je geocentrická délka ve stupních

$\Delta\Psi$ je nutace délky ve stupních

$\Delta\tau$ je korekce aberace světla ve stupních

2.2.8 Výpočet zdánlivého hvězdného (siderického) času

Střední hvězdný čas v Greenwichi

$$\begin{aligned} v_0 = {} & 280,460\,618\,37 + 360,985\,647\,366\,29 \cdot (JD - 2\,451\,545) + \\ & + 0,000\,387\,933 \cdot JC^2 - \frac{JC^3}{38\,710\,000} \end{aligned} \tag{2.24}$$

kde:

v_0 je střední hvězdný čas v Greenwichi ve stupních

JD je juliánský den

JC je juliánské století

Zdánlivý hvězdný čas v Greenwichi

$$v = v_0 + \Delta\Psi \cdot \cos(\epsilon) \tag{2.25}$$

kde:

v je zdánlivý hvězdný čas v Greenwichi ve stupních

v_0 je střední hvězdný čas v Greenwichi ve stupních

$\Delta\Psi$ je nutace délky ve stupních

ϵ je skutečný sklon dráhy k ekliptice v úhlových vteřinách

2.2.9 Výpočet geocentrické rektascenze slunce

$$\alpha = \operatorname{arctg2}\left(\frac{\sin(\lambda) \cdot \cos(\epsilon) - \operatorname{tg}(\beta) \cdot \sin(\epsilon)}{\cos(\lambda)}\right) \tag{2.26}$$

kde:

α je geocentrická rektascenze slunce ve stupních

λ je zdánlivá délka slunce ve stupních

ϵ je skutečný sklon dráhy k ekliptice v úhlových vteřinách

β je geocentrická šířka ve stupních

2.2.10 Výpočet geocentrické deklinace slunce

$$\delta = \arcsin\left(\sin\left(\beta\right) \cdot \cos\left(\epsilon\right) + \cos\left(\beta\right) \cdot \sin\left(\epsilon\right) \cdot \sin\left(\lambda\right)\right) \tag{2.27}$$

kde:
δ je geocentrická deklinace slunce ve stupních
β je geocentrická šířka ve stupních
ϵ je skutečný sklon dráhy k ekliptice v úhlových vteřinách
λ je zdánlivá délka slunce ve stupních

2.2.11 Výpočet lokálního hodinového úhlu

$$H = v + \sigma - \alpha \tag{2.28}$$

kde:
H je lokální hodinový úhel v místě pozorovatele ve stupních
v je zdánlivý hvězdný čas v Greenwichi ve stupních
σ je zeměpisná délka místa pozorovatele ve stupních
α je geocentrická rektascenze slunce ve stupních

2.2.12 Výpočet topocentrické rektascenze slunce

Rovníková horizontální paralaxa slunce

$$\xi = \frac{8,794}{3600 \cdot R} \tag{2.29}$$

kde:
ξ je rovníková horizontální paralaxa slunce ve stupních
R je radius v astronomických jednotkách

Paralaxa v rektascenzi slunce

$$u = \operatorname{arctg}\left(0,996\,647\,19 \cdot \operatorname{tg}\left(\varphi\right)\right) \tag{2.30}$$

$$x = \cos\left(u\right) + \frac{E}{6\,378\,140} \cdot \cos\left(\varphi\right) \tag{2.31}$$

$$y = 0,996\,647\,19 \cdot \sin\left(u\right) + \frac{E}{6\,378\,140} \cdot \sin\left(\varphi\right) \tag{2.32}$$

$$\Delta\alpha = \operatorname{arctg2}\left(\frac{-x \cdot \sin\left(\xi\right) \cdot \sin\left(H\right)}{\cos\left(\delta\right) - x \cdot \sin\left(\xi\right) \cdot \cos\left(H\right)}\right) \tag{2.33}$$

kde:

$\Delta\alpha$ je paralaxa v rektascenzi slunce ve stupních
u je v radiánech
φ je zeměpisná šířka místa pozorovatele ve stupních
E je nadmořská výška místa pozorovatele v metrech
ξ je rovníková horizontální paralaxa slunce ve stupních
H je lokální hodinový úhel v místě pozorovatele ve stupních
δ je geocentrická deklinace slunce ve stupních

Topocentrická rektascenze slunce

$$\alpha' = \alpha + \Delta\alpha \tag{2.34}$$

kde:
α' je topocentrická rektascenze slunce ve stupních
α je geocentrická rektascenze slunce ve stupních
$\Delta\alpha$ je paralaxa v rektascenzi slunce ve stupních

Topocentrická deklinace slunce

$$\delta' = \text{arctg2}\left(\frac{(\sin(\delta) - y \cdot \sin(\xi)) \cdot \cos(\Delta\alpha)}{\cos(\delta) - x \cdot \sin(\xi) \cdot \cos(H)}\right) \tag{2.35}$$

kde:
δ' je topocentrická deklinace slunce ve stupních
δ je geocentrická deklinace slunce ve stupních
ξ je rovníková horizontální paralaxa slunce ve stupních
$\Delta\alpha$ je paralaxa v rektascenzi slunce ve stupních
H je lokální hodinový úhel ve stupních

2.2.13 Výpočet topocentrického hodinového úhlu

$$H' = H - \Delta\alpha \tag{2.36}$$

kde:
H' je lokální topocentrický hodinový úhel ve stupních
H je lokální hodinový úhel ve stupních
$\Delta\alpha$ je paralaxa v rektascenzi slunce ve stupních

2.2.14 Výpočet topocentrického zenitu

Topocentrická výška slunce bez korekce atmosferické refrakce

$$e_0 = \arcsin\left(\sin\left(\varphi\right) \cdot \sin\left(\delta'\right) + \cos\left(\varphi\right) \cdot \cos\left(\delta'\right) \cdot \cos\left(H'\right)\right) \tag{2.37}$$

kde:

e_0 je topocentrická výška slunce bez korekce atmosferické refrakce ve stupních

φ je zeměpisná šířka místa pozorovatele ve stupních

δ' je topocentrická deklinace slunce ve stupních

H' je lokální topocentrický hodinový úhel ve stupních

Korekce atmosferické refrakce

$$\Delta e = \frac{P}{1010} \cdot \frac{283}{273 + T} \cdot \frac{1,02}{60 \cdot \mathrm{tg}\left(e_0 + \frac{10,3}{e_0 + 5,11}\right)} \tag{2.38}$$

kde:

Δe je korekce atmosferické refrakce ve stupních

P je průměrný roční lokální tlak v milibarech

T je průměrná roční lokální teplota ve ℃

e_0 je topocentrická výška slunce bez korekce atmosferické refrakce ve stupních

Topocentrická výška slunce

$$e = e_0 + \Delta e \tag{2.39}$$

kde:

e je topocentrická výška slunce ve stupních

e_0 je topocentrická výška slunce bez korekce atmosferické refrakce ve stupních

Δe je korekce atmosferické refrakce ve stupních

Topocentrický zenit

$$\theta = 90° - e \tag{2.40}$$

kde:

θ je topocentrický zenit ve stupních

e je topocentrická výška slunce ve stupních

2.2.15 Výpočet topocentrického azimutu

$$\Phi = \text{arctg2}\left(\frac{\sin\left(H'\right)}{\cos\left(H'\right)\cdot\sin\left(\varphi\right) - \text{tg}\left(\delta'\right)\cdot\cos\left(\varphi\right)}\right) + 180° \tag{2.41}$$

kde:

Φ je topocentrický azimut ve stupních

H' je lokální topocentrický hodinový úhel ve stupních

φ je zeměpisná šířka místa pozorovatele ve stupních

δ' je topocentrická deklinace slunce ve stupních

Výklad užitých pojmů lze nalézt např. v [30].

2.3 Ověření algoritmu výpočtu

Uvedený algoritmus byl pro vysoký počet iterací (přes 800) převeden do prostředí MATLAB, kde byl spuštěn výpočet se vstupními hodnotami[2] uvedenými v tab. 2.1. Výstupem výpočtu jsou hodnoty prezentované v tab. 2.2.

Výsledné hodnoty korespondují s údaji získanými z komerčních systémů pro určení polohy slunce.

Podrobné vyhodnocení přesnosti SPA je provedeno v [29].

Tab. 2.1: Vstupní hodnoty pro výpočet polohy slunce

datum	1. června 2017
čas	12.10:00
zeměpisná šířka	49,246 256°
zeměpisná délka	16,562 261°
nadmořská výška	284 m
průměrný roční tlak	981,9 hPa
průměrná roční teplota	9,9 ℃
ΔT	64,797 s
korekční činitel atmosferické refrakce	0,5667°

Zdroj: autor

[2]Hodnota ΔT byla určena dle [31].

Tab. 2.2: Výsledné hodnoty polohy slunce

topocentrický zenit	27,372 846°
topocentrický azimut	189,291 924°
východ slunce	4.53:03
pravé poledne	12.51:37
západ slunce	20.50:44

Zdroj: autor

2.4 Získaná data

Pro uvedené vstupní parametry (zeměpisnou polohu) byly provedeny výpočty polohy a časů východu a západu slunce pro roky 2017–2022. Výsledné hodnoty byly uloženy do textových souborů dostupných na přiloženém CD.

Soubor `casy.txt` obsahuje čárkami oddělené časy východu a západu slunce pro jednotlivé dny.

Poloha slunce (azimut a zenit) je počítána v desetiminutových intervalech a uložena v souborech pojmenovaných podle masky `MM-DD.txt`, kde `MM` značí měsíc a `DD` značí den. Uvnitř těchto souborů jsou čárkami oddělené údaje v pořadí hodina, minuta, zenit a azimut.

3 NÁVRH SOLÁRNÍHO TRACKERU A POHYBLIVÉHO STOJANU

Předmětný tracker musí plnit minimálně dvě funkce:

1. určení orientace solárního zařízení pro přímý směr ke slunci
2. řízení pohybu daného zařízení ve dvou osách při použití různých druhů pohonů

Zároveň musí být jeho technologie natolik flexibilní, aby bylo umožněno budoucí doplnění o další funkce.

Následující návrh si klade za cíl vytvoření univerzálního dvouosého trackeru použitelného pro různá solární zařízení a umožňujícího práci ve třech režimech. V automatickém režimu je reálná poloha slunce sledována pomocí optického senzoru slunečního svitu. Režim časovače nabízí časové řízení na základě předem vypočítané polohy slunce (dle kap. 2). Poslední, manuální režim slouží pro ruční ovládání.

3.1 Pohonné jednotky

Vzhledem k relativně vysoké hmotnosti jednotlivých komponent solárních zařízení je žádoucí zajistit nižší hybnost pomalejším pohybem pohyblivých částí. Optimální otáčky byly určeny experimentálně, jako vyhovující se jeví hodnota maximálně dvou otáček za minutu pro menší solární zařízení. U větších zařízení se jedná o čas potřebný na jednu otáčku až v řádech desítek minut. Z toho důvodu je nutno použít buďto pomaloběžný motor, nebo nízkých otáček docílit vhodným převodem (případně soustavou typu motor se závitovou tyčí).

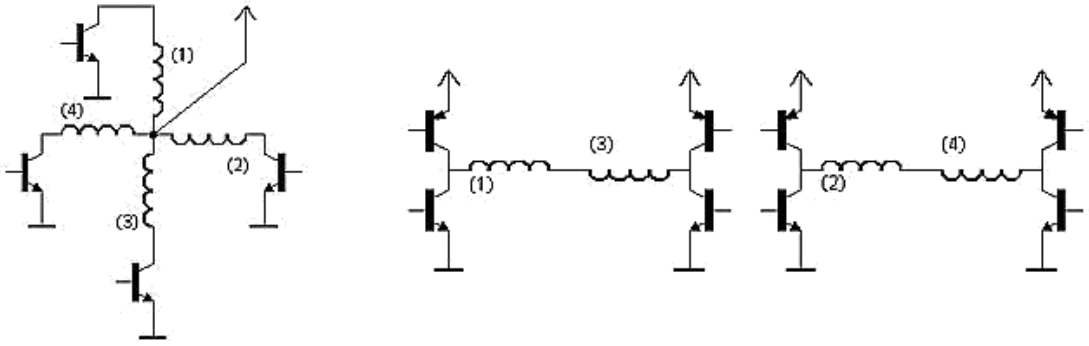
V úvahu přichází v zásadě tři druhy pohonů: krokové motory, stejnosměrné motory a servomotory.

3.1.1 Krokové motory

Krokový motor funguje na principu vytvoření rotujícího magnetického pole tvořeného procházejícím proudem postupně jednotlivými cívkami statoru. Toto pole otáčí rotorem tvořeným prstencem permanentních magnetů. Rotor se po dokončení pohybu vždy nachází v některé z diskrétních poloh, odtud označení krokový motor. Rychlost otáčení je omezena přechodnými magnetickými jevy a závisí na rychlosti přepínání aktivních cívek, změny smyslu otáčení se docílí změnou pořadí, v jakém jsou cívky spínány.

Řídit krokové motory lze několika způsoby. Základní rozdělení je na řízení unipolární a bipolární. Rozdíl spočívá v počtu cívek, kterými současně prochází proud. V případě řízení unipolárního prochází proud vždy právě jednou cívkou, přičemž

bipolární řízení zajišťuje, že proud prochází vždy dvěma protilehlými cívkami. Bipolární řízení zajišťuje větší kroutící moment, nicméně vyžaduje složitější zapojení využívající H-můstků, jak znázorňuje obr. 3.1. Zároveň je spotřeba bipolárně řízeného motoru vyšší než v případě unipolárního řízení. Volba způsobu řízení (resp. zapojení) se nenabízí u každého motoru, ale pouze u motorů univerzálních. Ostatní motory se vyrábějí buďto jako unipolární, nebo jako bipolární bez možnosti změny.



Obr. 3.1: Schéma zapojení unipolárního (vlevo) a bipolárního (vpravo) řízení krokového motoru

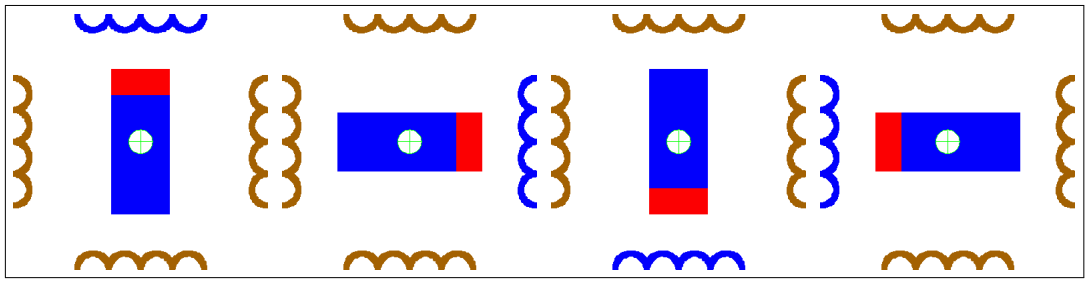
Zdroj: [32], upraveno autorem

Další rozdělení řízení je na jednofázové a dvoufázové. Při jednofázové variantě je buzena vždy jen jedna cívka (resp. dvojice cívek, jedná-li se o bipolární řízení) oproti dvěma sousedním cívkám, které jsou buzeny v případě dvoufázového řízení. Dvoufázové řízení zajišťuje vyšší kroutící moment (protože rotor je přitahován dvojnásobným počtem cívek), spotřeba motoru je pak ale pochopitelně dvojnásobná.

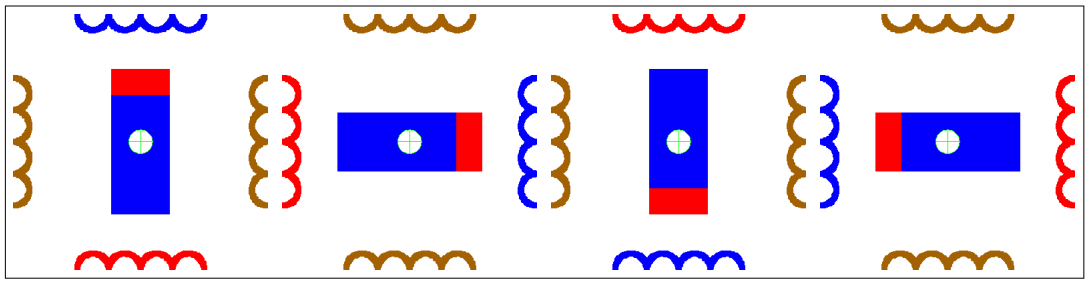
Poslední možností je volba velikosti kroku. Krokové motory jsou provozovány s plným nebo polovičním krokem. Plný krok značí běžný režim chodu motoru, polovičního kroku (tzv. mikrokrokování) lze docílit střídáním jednofázového a dvoufázového řízení. Počet kroků na jednu otáčku rotoru je pak dvojnásobný, což znamená dvojnásobnou přesnost.

Kombinaci možných způsobů řízení prezentují obr. 3.2 až 3.7, které ukazují jednoduchý motor se čtyřmi kroky na jednu otáčku. Modrá barva cívky značí, že její magnetické pole přitahuje červený pól rotoru, červená barva naopak přitahuje modrý pól, hnědé cívky jsou bez proudu.

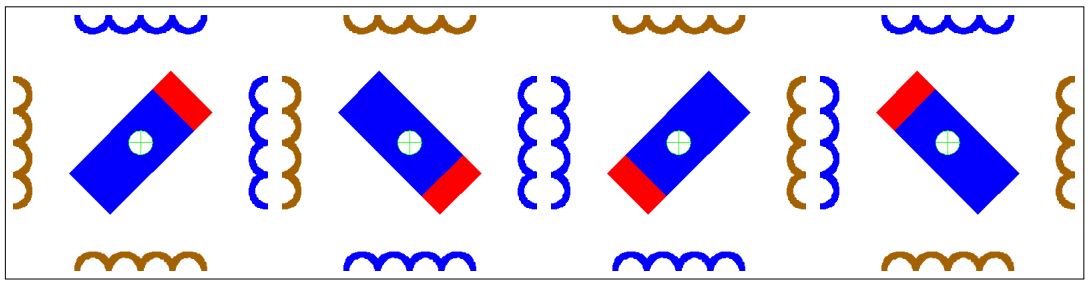
Výhodou krokových motorů je především možnost využít počítání kroků pro určení úhlu natočení hřídele, je-li známa výchozí poloha. Odpadá tedy nutnost použít enkodér nebo jiné zařízení. Nevýhodami jsou poměrně vysoká cena při poměrně malých kroutících momentech, ale především proudový odběr nezávislý na zátěži – i v klidu. Tím samozřejmě výrazně vzrůstá spotřeba a nároky na chlazení [34].



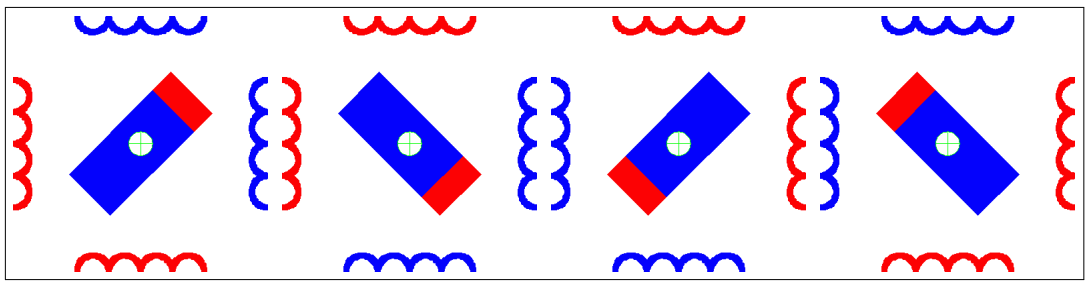
Obr. 3.2: Unipolární jednofázové řízení krokového motoru s plným krokem
Zdroj: [33], upraveno autorem



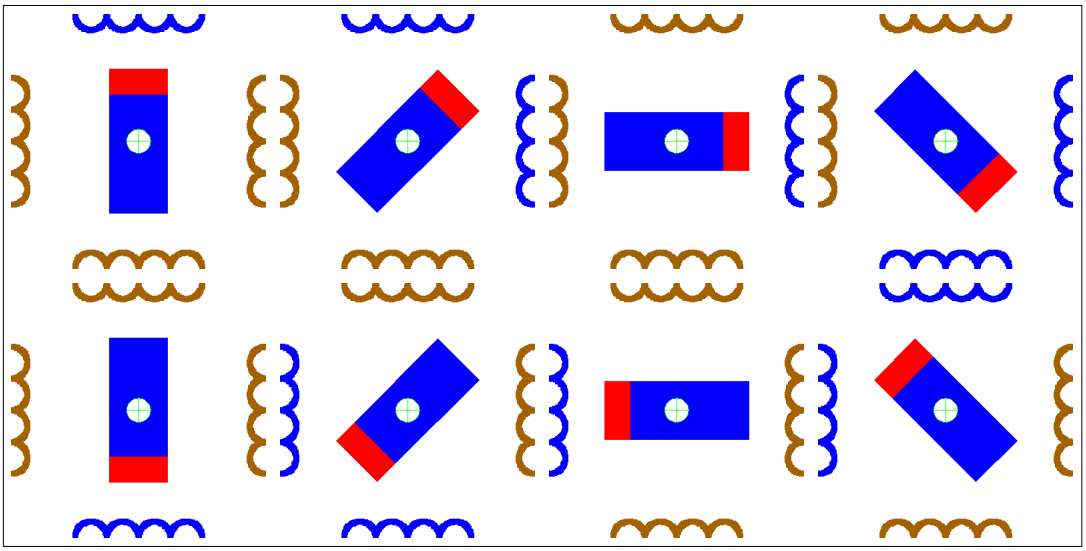
Obr. 3.3: Bipolární jednofázové řízení krokového motoru s plným krokem
Zdroj: [33], upraveno autorem



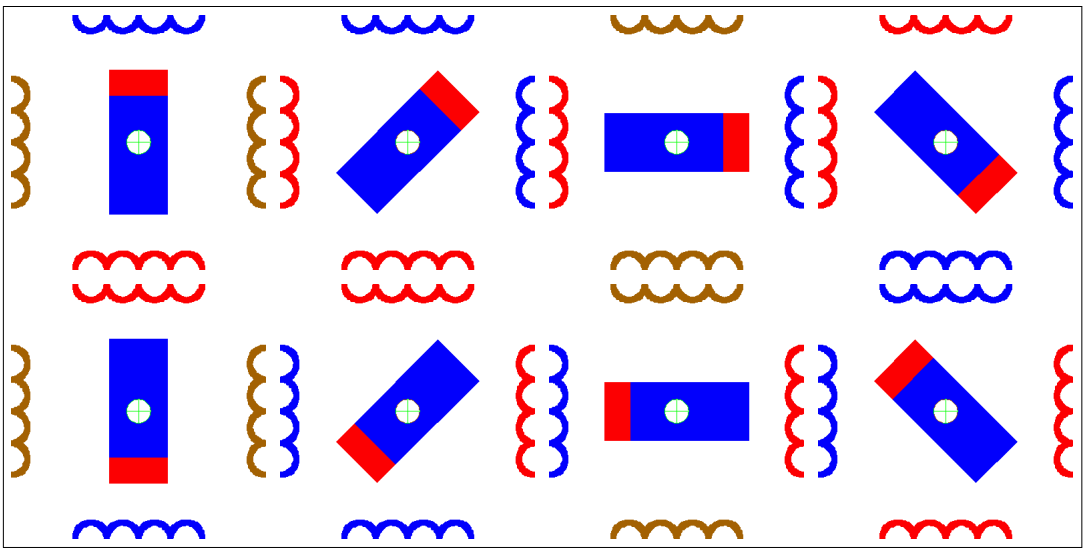
Obr. 3.4: Unipolární dvoufázové řízení krokového motoru s plným krokem
Zdroj: [33], upraveno autorem



Obr. 3.5: Bipolární dvoufázové řízení krokového motoru s plným krokem
Zdroj: [33], upraveno autorem



Obr. 3.6: Unipolární řízení krokového motoru s poloviční krokem
Zdroj: [33], upraveno autorem



Obr. 3.7: Bipolární řízení krokového motoru s poloviční krokem
Zdroj: [33], upraveno autorem

3.1.2 Stejnosměrné motory

Původní stejnosměrný motor má ve statorové části umístěno vinutí pole a na rotoru vinutí kotvy. Po přivedení proudu do obou vinutí vzniknou dvě magnetická pole, která na sebe začnou vzájemně silově působit, čímž se rotor uvede do pohybu. Při napájení stejnosměrným proudem by se však rotor měl přestat točit v okamžiku, kdy se póly rotoru dostanou do zákrytu s pólovými nástavci statoru (kotva se nachází v neutrální poloze). Díky setrvačnosti ale pohyb neustane ihned a komutátor změní

polaritu proudu procházejícího rotorem. Tím na rotor opět působí síla, která jej udrží v pohybu. Komutátor zajišťuje periodickou změnu polarity, čímž je umožněn kontinuální pohyb.

Statorové vinutí motoru lze nahradit permanentními magnety v počtu odpovídajícím počtu pólů. V takovém případě pro napájení motoru postačí dva vodiče. Tato varianta se pro motory malého výkonu stala převažující.

Otáčky rotoru jsou přímo úměrné napětí kotvy. To lze řídit buďto přímo, nebo pomocí PWM (Pulse Width Modulation; pulzně šířková modulace), tedy diskrétní modulace napětí. Určujícím parametrem PWM je střída D:

$$D = \frac{t}{T} \tag{3.1}$$

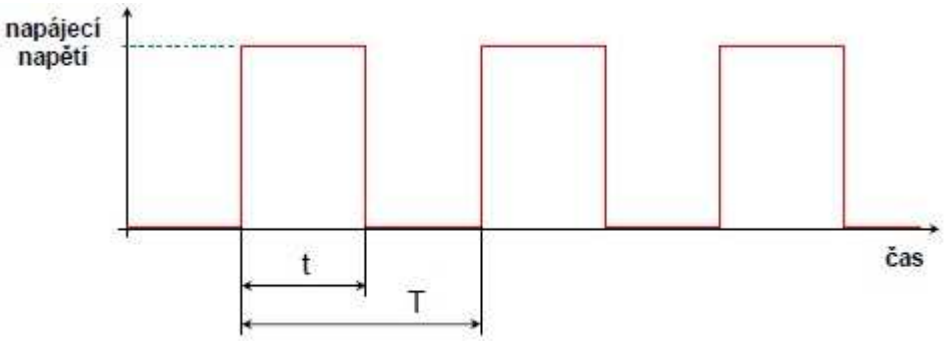
kde:
t je délka pulzu
T je délka periody

O výsledných otáčkách pak rozhoduje střední napětí $U_{stř}$:

$$U_{stř} = D \cdot U_{max} = \frac{t}{T} \cdot U_{max} \tag{3.2}$$

kde:
U_{max} je velikost napájecího napětí

Příklad průběhu napětí při řízení PWM je znázorněn na obr. 3.8.



Obr. 3.8: Příklad řízení PWM

Zdroj: [35]

Tento způsob řízení je energeticky efektivnější, nicméně klade vyšší nároky na použitou elektroniku (v závislosti na rychlosti spínání a velikosti procházejícího proudu).

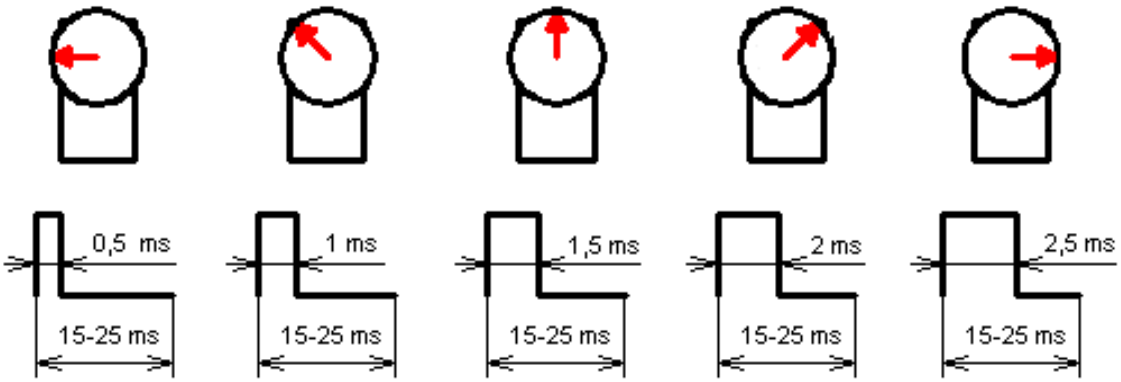
Hlavní výhodou stejnosměrných motorů je snadné řízení díky lineární závislosti otáček na napětí. Naopak nevýhodou je zejména přítomnost komutátoru a s tím spojené opotřebení kluzných kontaktů (uhlíků) a jiskření.

Stejnosměrný motor se obvykle nachází i v lineárních aktuátorech, které jsou schopny posuvného pohybu a lze je pro uvažovanou aplikaci rovněž s výhodou použít.

3.1.3 Servomotory

Servomotor je prakticky jakýkoliv motor vybavený zpětnou vazbou, která umožňuje nastavit přesnou polohu ramene (hřídele). Menší servomotory jsou nejčastěji realizovány jako stejnosměrné motory s několikastupňovou převodovkou, která zajišťuje poměrně velkou sílu motoru i při malém výkonu. Zpětnovazebním prvkem je u těchto motorů zpravidla potenciometr připojený k řídicí elektronice. Právě potenciometr je jedním z důvodů, proč mají ramena servomotorů omezený rozsah pohybu (obvykle kolem 180°). K ovládání je opět využito principu PWM, řídicí pulzy jsou kladné o délce 1–2 ms, opakované s frekvencí přibližně 50 Hz. Po vstupu řídicího impulzu vygeneruje klopný obvod elektroniky impulz o délce odpovídající aktuální poloze ramene s opačnou polaritou. Následně se tento impulz porovná s řídicím impulzem a vygeneruje se rozdílový impulz, který zajistí pohyb ramene do správné polohy [36].

Princip řízení malých servomotorů vysvětluje obr. 3.9. Střední (neutrální) polohy ramene se docílí impulzem o délce 1,5 ms, impulzy o 0,5 ms kratší (delší) způsobí natočení ramene o 45° v záporném (kladném) směru. Ostatních poloh (včetně mezipoloh na obrázku neznázorněných) je dosaženo analogicky. Velikost impulzu je shodná s velikostí napájecího napětí [36].



Obr. 3.9: Princip PWM řízení servomotoru

Zdroj: [37]

3.2 Řídicí elektronika

Pro ovládání všech prvků trackeru je využita platforma Arduino, tedy otevřená elektronická platforma založená na jednoduchém hardwaru a softwaru. Hardware je

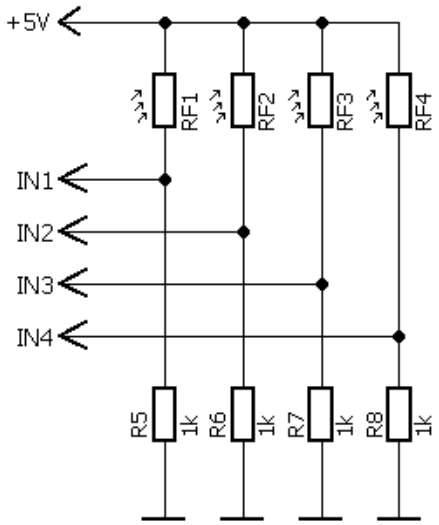
tvořen malým jednodeskovým počítačem založeným na mikrokontroleru ATmega, který je programován pomocí jazyka vycházejícího z jazyka Wiring. Pro zpřehlednění a zjednodušení práce jsou funkce podobného určení sdružovány v knihovnách. Existuje nepřeberné množství oficiálních i neoficiálních knihoven, jež jsou volně k dispozici.

Arduino nabízí několik modelů desek od malých (Arduino Mini) až po největší (Arduino Tre). Kromě velikosti se jednotlivé desky liší samozřejmě i vybaveností a výkonností. Pro řízení trackeru se jako optimální model jeví Arduino Mega 2560. To využívá mikrokontroler ATmega2560 a nabízí mj. 54 digitálních vstupů/výstupů, 16 analogových vstupů a USB konektor.

3.2.1 Čidlo slunečního svitu

Jako optických senzorů je využito fotorezistorů VT93N2, které vykazují za tmy odpor 500 kΩ, při osvětlení o intenzitě 10 lx pak 50 kΩ. Rezistory jsou umístěny do čtyřkvadrantového černého stínítka, které je z důvodu předpokládané vyšší teploty vyrobeno z ABS.

Schéma zapojení čidla předkládá obr. 3.10. Ke každému z fotorezistorů je připojeno napětí +5 V a Arduino pomocí čtyř analogových vstupů (`IN1`–`IN4`) porovnává úbytky napětí na sousedních dvojicích rezistorů. Je-li úbytek na některé dvojici (průměrný úbytek napětí na dvou rezistorech tvořících dvojici) větší než na dvojici sousední (a přesahuje-li stanovenou toleranci), je patrně intenzita osvětlení odlišná vlivem zastínění, a zařízením je třeba otočit příslušným směrem. Pokud se slunce nachází právě v ose stínítka, intenzita osvětlení jednotlivých rezistorů je přibližně stejná, tudíž jsou přibližně stejné i úbytky napětí a zařízení je nasměrováno správně.



Obr. 3.10: Schéma zapojení čidla slunečního svitu

Zdroj: autor

Kvůli omezení povětrnostních vlivů je celé čidlo umístěno do průhledné kopule, která po důkladném utěsnění brání vnitřnímu orosení i při poměrně velkých teplotních výkyvech.

Čidlo včetně zmíněné kopule je zachyceno na obr. 3.11, 3D model použitého stínítka je k dispozici na přiloženém CD.

Obr. 3.11: Hotové čidlo slunečního svitu

Zdroj: autor

3.2.2 Snímače polohy

Pro získání informace o aktuálním natočení systému potřebné zejména pro zajištění funkce řízení časovačem je třeba zjišťovat skutečné úhly natočení (azimut a zenit). Nejdostupnějšími možnostmi jsou pravděpodobně využití digitálních snímačů polohy (magnetometr, akcelerometr, gyroskop) nebo osazení potenciometry.

Digitální snímače

K určení azimutu prvním jmenovaným způsobem lze využít např. tříosý digitální magnetometr HMC5883L, který umožňuje měřit magnetickou indukci ve třech osách, a výpočtem následně určit azimut. Výpočet spočívá ve využití funkce arkus tangens

aplikované na hodnoty pro osy y a x. Následně je provedena korekce pomocí magnetické deklinace – úhlového rozdílu mezi zeměpisným a magnetickým severním pólem. Výsledný azimut je pak určen vztahem 3.3.

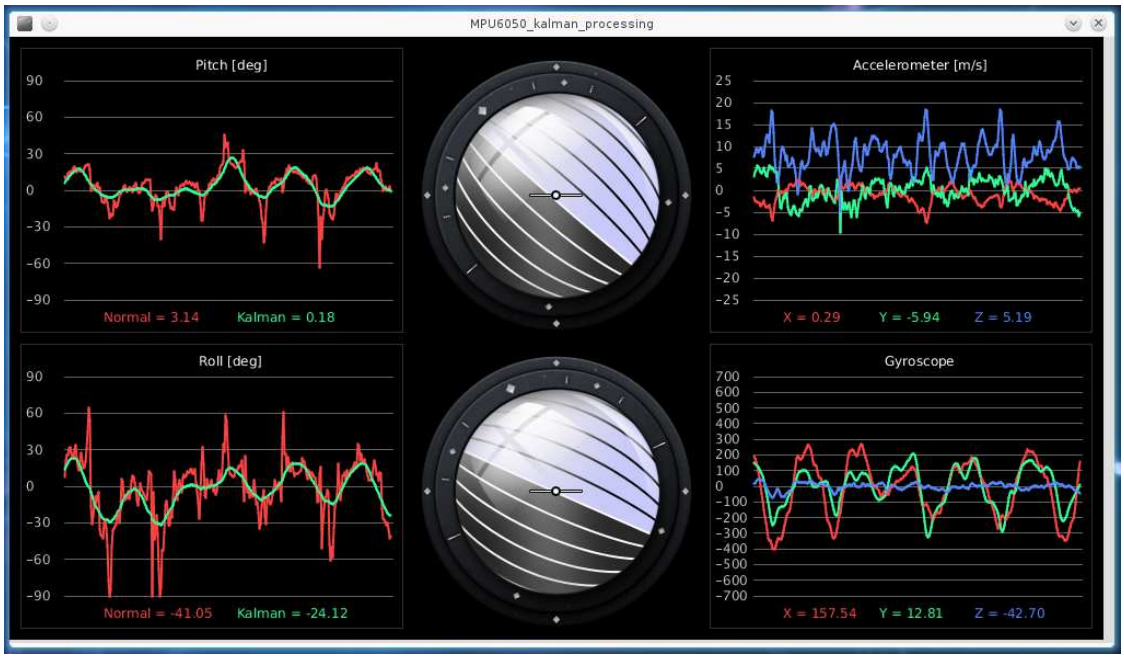
$$\Phi = \text{arctg2}\left(\frac{B_y}{B_x}\right) + \delta \tag{3.3}$$

kde:
B_y je magnetická indukce ve směru osy y
B_x je magnetická indukce ve směru osy x
δ je magnetická deklinace v radiánech dle [38]

Akcelerometr je velmi citlivý na jakoukoliv změnu náklonu, proto musí být vždy umístěn vodorovně. Pokud takové umístění není možné zajistit, lze nepříznivý vliv náklonu eliminovat kompenzačním algoritmem, je-li znám úhel náklonu. Algoritmus lze však použít jen pro omezený rozsah úhlu náklonu [39]. V případě zamýšlené aplikace by nicméně neměl být problém zajistit správné umístění senzoru.

Určení zenitu zajistí např. čip MPU6050 – tříosý akcelerometr a tříosý gyroskop současně, navíc s integrovanou Digital Motion Processor (DMP) jednotkou. Právě modul DMP výrazně zjednodušuje proces zpracování dat ze všech tří senzorů, protože dokáže provést potřebné výpočty sám a předat pouze výsledné úhly natočení vůči zemi. Při výpočtu úhlu náklonu je signál z gyroskopu integrován, čímž dochází k postupné kumulaci chyby měření. Tento jev se nazývá drift. Pro jeho eliminaci lze využít poměrně komplikovaný matematický aparát zvaný Kálmánův filtr, tedy algoritmus, který dokáže hodnoty požadovaných proměnných (v tomto případě především úhel náklonu) predikovat a upřesnit na základě posledních naměřených dat a předchozích hodnot těchto proměnných [40]. Konkrétně pro tuto aplikaci jsou data získaná z akcelerometru zpřesňována pomocí gyroskopu. Efekt využití Kálmánova filtru pro korekci dat z MPU6050 je znázorněn na obr. 3.12 – v levé části jsou červeně zaznamenány nefiltrované hodnoty, zelenou barvou jsou vyneseny hodnoty získané aplikací filtru. Vpravo jsou pak surová data z akcelerometru a gyroskopu.

Oba senzory jsou napájeny na hladině 3,3 V a pro připojení k Arduinu využívají sběrnici I^2C realizovanou dvěma obousměrnými vodiči – SCL a SDA. Pro práci s těmito senzory je třeba načíst příslušné knihovny `HMC5883L` a `MPU6050` a pro použití Kálmánova filtru také knihovnu `KalmanFilter`. Pro snazší zpracování dat a využití pokročilých funkcí (a současně pro neexistenci oficiální knihovny pro použití Kálmánova filtru) je však výhodnější použít namísto oficiálních knihoven od firem Adafruit nebo Honeywell neoficiální knihovny Kornelia Jarzębského [42].

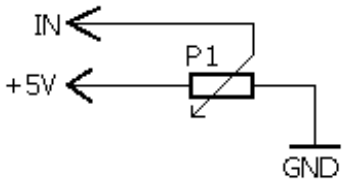


Obr. 3.12: Aplikace Kálmánova filtru na údaje o úhlu natočení

Zdroj: [41]

Potenciometry

Potenciometry lze užít v případě, že to umožňuje konstrukce stojanu – základna potenciometru musí být uchycena nehybně, zatímco jezdec musí být spojen s hřídelí a otáčet se s ní. Zjištění potřebných úhlů pomocí potenciometrů je výrazně jednodušší. Potenciometr je zapojen dle schématu na obr. 3.13, kde vstup `IN` je analogovým vstupem Arduina. Takto zapojený potenciometr funguje jako napěťový dělič a na analogový vstup je přivedeno napětí odpovídající aktuálnímu natočení soustavy (resp. natočení potenciometru). Je třeba pouze zjistit velikost napětí v obou krajních polohách a úhel natočení potom určovat dle vztahu 3.4.



Obr. 3.13: Schéma zapojení potenciometru pro měření úhlu natočení

Zdroj: autor

$$\alpha = \frac{U - U_{min}}{\frac{U_{max} - U_{min}}{\alpha_{max} - \alpha min}} \tag{3.4}$$

kde:

α	je aktuální úhel natočení
U	je měřené napětí při aktuálním natočení
U_{min}	je napětí v první krajní poloze
U_{max}	je napětí v druhé krajní poloze
α_{min}	je úhel odpovídající první krajní poloze
α_{max}	je úhel odpovídající druhé krajní poloze

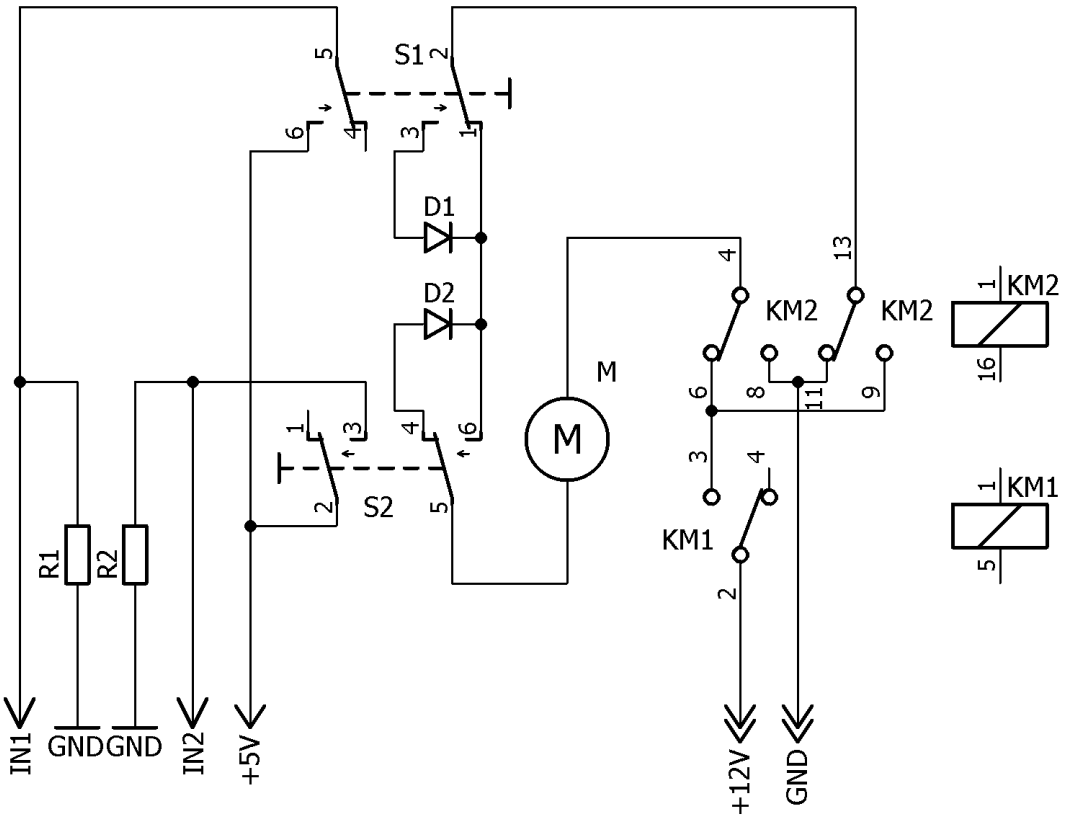
3.2.3 Koncové pohybové spínače

Pokud použitá konstrukce neumožňuje neomezenou rotaci (např. pomocí rotačního kroužkového přenašeče), je nutné pohyb omezit. K tomu účelu mohou posloužit snímače polohy (viz oddíl 3.2.2) doplněné softwarovým blokováním motorů, ale spolehlivějším prostředkem jsou koncové spínače. Použít lze dle konstrukčních možností různé druhy (pákové spínače s kladkou, rtuťové spínače, tlačítkové spínače apod.), princip je vždy stejný: při dosažení krajní polohy jsou kontakty spínače spojeny (resp. rozpojeny), čímž dojde k vyslání (resp. přerušení) signálu Arduinu, které ihned ukončí pohyb motoru.

Určitou nástavbou je zapojení dle schématu na obr. 3.14, kde je kromě signalizace Arduinu pohyb motoru v daném směru přerušen přímo. Tento způsob zapojení je však možný pouze při použití stejnosměrného motoru.

Pro vysvětlení zapojení koncových spínačů dle uvedeného schématu předpokládejme, že při znázorněné polaritě bude po sepnutí relé `KM1` motor `M` otáčet zařízením doprava. Spínač `S1` je umístěn v pravé krajní poloze a spínač `S2` v levé krajní poloze. Při otáčení vpravo dojde při dosažení krajní polohy k sepnutí spínače `S1`, v tu chvíli se motor přestane točit, protože proud neprojde přes diodu `D1` v závěrném směru, zároveň se objeví napětí na vstupu `IN1` Arduina. Motor se nyní může točit až po sepnutí relé `KM2`. V takovém případě se změní polarita, motor se začne točit opačným směrem a dojde k rozpojení spínače `S1`.

Spínač `S2` funguje analogicky při chodu motoru vlevo.



Obr. 3.14: Varianta zapojení koncových spínačů pro přímé zastavení motoru
Zdroj: autor

3.2.4 Ostatní komponenty

Pro manuální ovládání je použit dvouosý analogový joystick, pro režim časovače je potom nutná čtečka SD (resp. microSD) karet a modul reálného času DS3231, jehož údaj o čase je zálohován pomocí knoflíkové baterie CR2032. Informace jsou zobrazovány pomocí podsvíceného LCD o dvou řádcích (2 × 16 znaků).

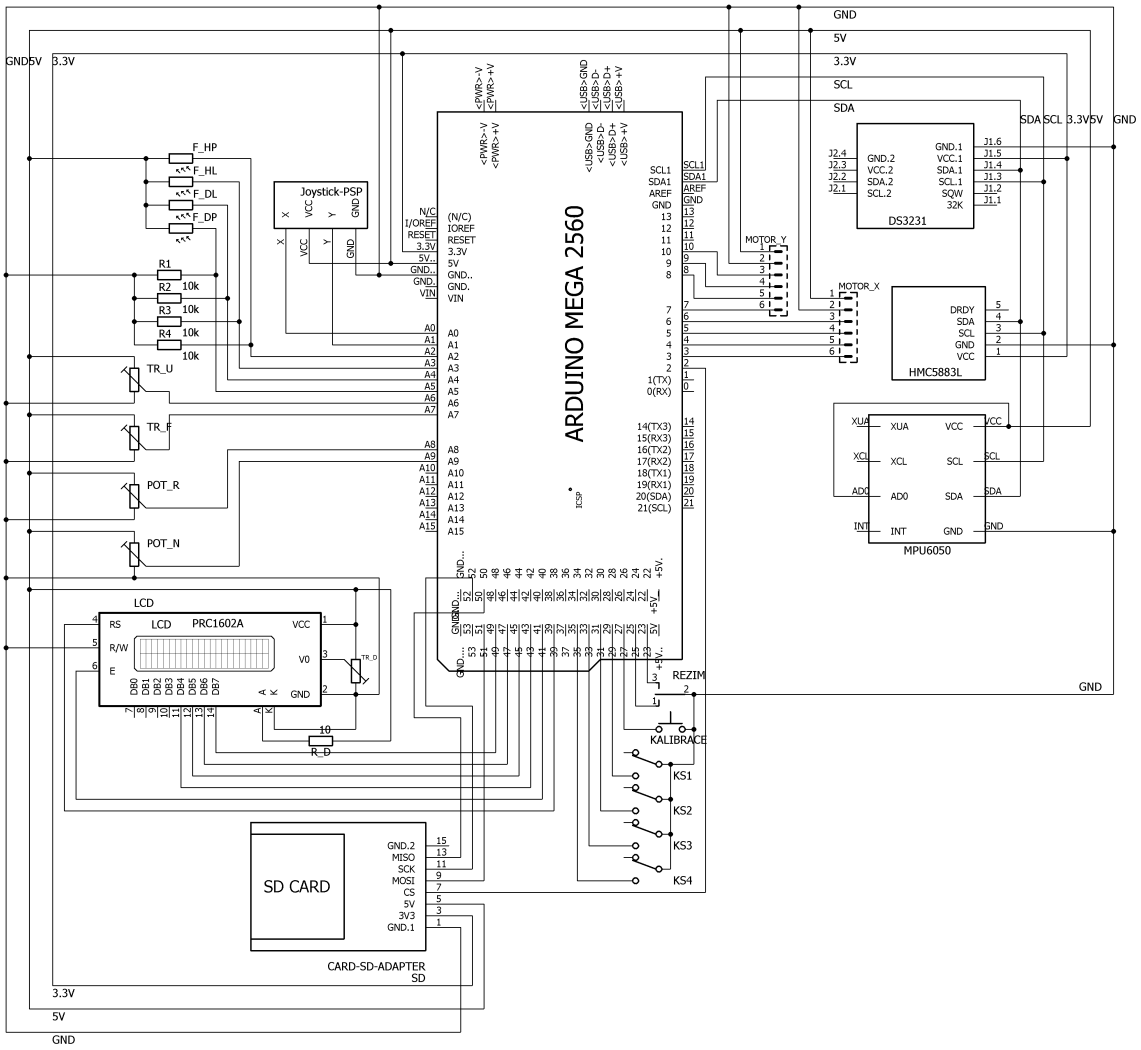
Celý systém může být díky množství volných vstupů a výstupů rozšířen o celou řadu dalších periferií, a tedy i funkcí. Jmenovat lze např. měření energetického zisku wattmetrem, úpravu pozice solárního zařízení na základě údajů z anemometru nebo pyranometru, záznam změřených dat, komunikaci prostřednictvím Wi-Fi apod. Konkrétní řešení je však závislé na konkrétní aplikaci, a především na dostupných komponentách.

3.2.5 Celkové schéma zapojení

Obr. 3.15 představuje logické schéma zapojení všech komponent. Základem celého zařízení je deska plošných spojů, ke které je rozebiratelným způsobem (pomocí

řadových kolíků) připojeno Arduino. Prvky `F_HP`, `F_HL`, `F_DL` a `F_DP` jsou fotorezistory (označení `H/D` značí horní/dolní kvadrant, `L/P` potom vpravo/vlevo), které spolu s odpory `R1` až `R4` odpovídají zapojení na obr. 3.10. `TR_U` je trimr pro nastavení požadované přesnosti úhlů v režimu časovače, `TR_F` je trimr nastavující citlivost fotorezistorů. Potenciometry `POT_R` a `POT_N` jsou volitelné a poskytují informace o aktuálním natočení (`R`) a náklonu (`N`) (viz oddíl 3.2.2). V levé části schématu na obr. 3.15 lze dále najít joystick, LCD a čtečku SD karet. Trimr `TR_D` slouží k nastavení kontrastu displeje a odpor `R_D` určuje proud pro jeho podsvícení.

V pravé části schématu se nachází svorkovnice `MOTOR_X` a `MOTOR_Y` pro připojení motorů, modul reálného času `DS3231`, magnetometr `HMC5883L` a akcelerometr `MPU6050`. Ve spodní části schématu lze najít několik spínacích prvků. Přepínač `REZIM` je třípolohový přepínač pracovního režimu, `KALIBRACE` je tlačítko pro spuštění procesu kalibrace snímačů polohy a spínače `KS1` až `KS4` jsou volitelné koncové spínače.

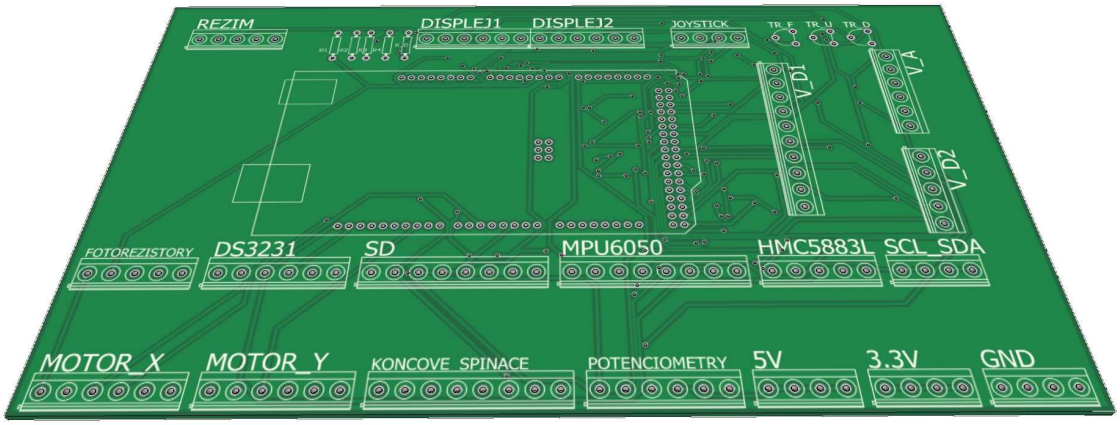


Obr. 3.15: Přehledové schéma zapojení celé sestavy

Zdroj: autor

Ve schématu není znázorněno napájení Arduina (7–12 V; 0,5 A), jeho jištění a hlavní vypínač. Na desce jsou rovněž vyvedeny některé nevyužité piny Arduina pro případ možného budoucího rozšíření o další funkce a periferie. Jedná se o šest analogových vstupů a patnáct digitálních vstupů/výstupů.

Z důvodu snadné výměny jednotlivých komponent a možnosti libovolného umístění jsou kromě odporů a trimrů veškeré prvky připojeny pomocí řadových šroubových svorkovnic. Rozvržení desky plošných spojů nabízí obr. 3.16, popis jednotlivých svorkovnic je pak uveden v příloze B. Návrh desky a matrice pro výrobu jsou k dispozici na přiloženém CD.



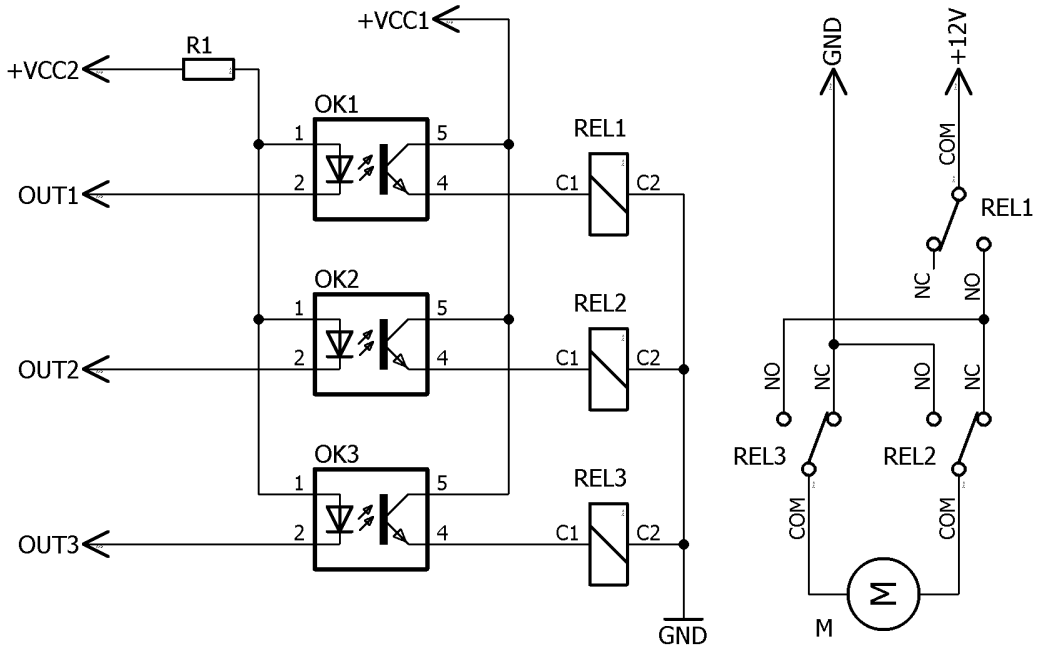
Obr. 3.16: Rozvržení prvků na desce plošných spojů

Zdroj: autor

Připojení motorů prostřednictvím svorkovnic umožňuje použití všech dříve zmiňovaných typů motorů. Krokové motory a servomotory se připojují pomocí meziobvodů (driverů), které přijímají z Arduina pouze signály. Stejnosměrné motory žádné drivery nepoužívají, pro jejich ovládání lze použít zapojení dle schématu na obr. 3.17, kde relé `REL1` spíná motor `M`, pro změnu smyslu otáčení je třeba dále sepnout relé `REL2` a `REL3`. Sepnutí relé je realizováno připojením výstupů `OUT` na úroveň `GND`. Optočleny `OK1` až `OK3` jsou zařazeny jednak z důvodu výkonového omezení Arduina (40 mA na jeden výstup, 200 mA celkově), jednak z důvodu galvanického oddělení obvodů o různých napětích.

Relé `REL2` a `REL3` lze samozřejmě nahradit jediným relé s dvojicí přepínacích kontaktů, na trhu jsou ale dostupné hotové reléové moduly navržené přímo pro Arduino[1], které disponují optickým oddělovačem a využívají relé s jedním přepínacím kontaktem, proto je ve schématu využito zapojení předpokládající tuto variantu.

[1]Nejčastěji jde o relé s kontakty dimenzovanými na 250 V AC/30 V DC a 10 A.



Obr. 3.17: Schéma řízení stejnosměrných motorů

Zdroj: autor

3.3 Softwarová část

Každý algoritmus pro Arduino sestává z jednoho souboru (který se ale může odkazovat na další knihovny), který povinně obsahuje dvě funkce typu void – `setup()` a `loop()`. První jmenovaná funkce se vykoná pouze jednou, a sice při spuštění Arduina. Druhá funkce je vykonávána v nekonečné smyčce. V případě Arduino Mega 2560 lze pro nahrání programu využít USB port. Ten zároveň slouží pro komunikaci po sériové lince.

Dále uvedené části kódu představují minimální funkční verzi předpokládající využití krokových motorů. Podbarvením jsou zvýrazněny řádky kódu odpovídající konkrétnímu typu pohonu, úpravy pro ostatní varianty pohonů jsou uvedeny v části 3.3.5 stejně tak jako úpravy pro volitelné příslušenství (potenciometry, koncové spínače) – viz oddíl 3.3.6. Pro snazší orientaci je využito číslování řádků, které je konzistentní s kódem, který je v příloze C uveden vcelku.

3.3.1 Hlavička a deklarace proměnných

První částí kódu je hlavička, kde jsou definovány importované knihovny, dále je zde nastavena adresa modulu reálného času na sběrnici I^{2}C a je zde určen datový pin pro komunikaci se čtečkou SD karet[2]:

[2]Příkaz `define` není na rozdíl od většiny ostatních ukončen středníkem.

```
#include <Wire.h>                  //komunikace po I2C
#include <Stepper.h>               //krokove motory
#include <HMC5883L.h>              //magnetometr
#include <SPI.h>                   //komunikace po SPI
#include <SD.h>                    //ctecka SD karet
#include <LiquidCrystal.h>         //LCD
#include <MPU6050.h>               //akcelerometr
#include <KalmanFilter.h>          //Kalmanuv filtr
#define DS3231_I2C_ADDRESS 0x68    //nastaveni adresy casoveho modulu
#define CS_PIN 2                   //nastaveni datoveho pinu ctecky
```

Za hlavičkou ihned následuje deklarace potřebných proměnných. Kromě pracovních proměnných jsou zde definována i čísla pinů Arduina pro jednotlivé prvky. Pokud je před číslem pinu uvedeno `A`, jde o analogový vstup.

```
int xPin = A0;          //osa x joysticku
int yPin = A1;          //osa y joysticku
int hpPin = A2;         //fotorezistor HP
int hlPin = A3;         //fotorezistor HL
int dlPin = A4;         //fotorezistor DL
int dpPin = A5;         //fotorezistor DP
int pinCas = 23;        //prepinac rezimu v poloze casovace
int pinMan = 25;        //prepinac rezimu v poloze manualni
int pinKal = 27;        //spinac kalibrace
int PinX0 = 3;          //piny motoru nataceni (osa x)
int PinX1 = 4;
int PinX2 = 5;
int PinX3 = 6;
int PinY0 = 7;          //piny motoru naklonu (osa y)
int PinY1 = 8;
int PinY2 = 9;
int PinY3 = 10;
int trimr_u = A6;       //trimr tolerance uhlu
int trimr_f = A7;       //trimr tolerance fotorezistoru

const byte zaznamu = 200;          //zaznamy na karte pro jeden den
float necitlivost_f = 0.08 * 256; //tolerance fotorezistoru
float necitlivost_u = 5;           //tolerance uhlu

//nastaveni pinu pro displej
LiquidCrystal lcd(39, 41, 43, 45, 47, 49);

//nastaveni motoru
Stepper motorX(200, PinX0, PinX2, PinX1, PinX3);
Stepper motorY(200, PinY0, PinY2, PinY1, PinY3);

```

```
//nastaveni snimacu polohy a Kalmanova filtru
HMC5883L hmc;
MPU6050 mpu;
KalmanFilter kalmanX(0.001, 0.003, 0.03);
KalmanFilter kalmanY(0.001, 0.003, 0.03);

float akc_naklon = 0;    //naklon dle akcelerometru
float azimut = 0;        //skutecny azimut
float zenit = 0;         //skutecny zenit
float azimut_n = 0;      //pozadovany azimut
float zenit_n = 0;       //pozadovany zenit
int xPozice = 0;         //vychylka joysticku v ose x
int yPozice = 0;         //vychylka joysticku v ose y
byte rezim = 1;    //rezim: 1 = automaticky, 2 = casovac, 3 = manualni
int hp = 0;              //hodnota fotorezistoru HP
int hl = 0;              //hodnota fotorezistoru HL
int dp = 0;              //hodnota fotorezistoru DP
int dl = 0;              //hodnota fotorezistoru DL
int nahore = 0;          //osvetleni horni casti
int dole = 0;            //osvetleni dolni casti
int vlevo = 0;           //osvetleni leve casti
int vpravo = 0;          //osvetleni prave casti
int rozdilX = 0;         //rozdil osvetleni v ose x
int rozdilY = 0;         //rozdil osvetleni v ose y
int i = 0;               //pomocna promenna
int j = -1;              //pomocna promenna

//promenne casu
byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
int cas_h, cas_m, den_m;
float pole[zaznamu][4]; //pole dat z SD karty
byte cislo = 0;          //pomocna promenna
byte radek = 0;          //pomocna promenna
char soubor[9] = "";     //nazev souboru na karte
File file;               //ukazatel souboru
size_t n;                //pomocna promenna
char str[20];            //pomocna promenna
```

3.3.2 Funkce pro práci se soubory a s časem

Dále je třeba definovat pomocné funkce. První z nich je `readField`, která prochází soubor `file` po segmentech `str` (o velikosti `size`) dle zadaného oddělovače `delim` a vrací délku segmentu.

```
//prace se soubory
```

```
size_t readField(File* file, char* str, size_t size, char* delim) {
  char ch;
  size_t n = 0;
  while ((n + 1) < size && file->read(&ch, 1) == 1) {
    if (ch == '\r') {
      continue;
    }
    str[n++] = ch;
    if (strchr(delim, ch)) {
        break;
    }
  }
  str[n] = '\0';
  return n;
}
```

Další funkce slouží pro práci s modulem reálného času. Nejprve jsou to funkce `decToBcd` a `bcdToDec` sloužící pro převod čísel z desítkové soustavy do BCD a zpět. Tyto konverze jsou nezbytné pro fungování navazujících funkcí pro nastavení aktuálního času `setDS3231time` a pro jeho získání `readDS3231time`.

```
//prevod mezi soustavami
byte decToBcd(byte val){ return( (val/10*16) + (val%10) ); }
byte bcdToDec(byte val){ return( (val/16*10) + (val%16) ); }

//nastaveni casu
void setDS3231time(byte second, byte minute, byte hour, byte dayOfWeek,
     byte dayOfMonth, byte month, byte year){
    Wire.beginTransmission(DS3231_I2C_ADDRESS); //navazani spojeni
    Wire.write(0);            //nastaveni ukazatele na registr sekund
    Wire.write(decToBcd(second));              //nastaveni sekund
    Wire.write(decToBcd(minute));              //nastaveni minut
    Wire.write(decToBcd(hour));                //nastaveni hodin
    Wire.write(decToBcd(dayOfWeek));           //nastaveni dne v tydnu
    Wire.write(decToBcd(dayOfMonth));          //nastaveni data
    Wire.write(decToBcd(month));               //nastaveni mesice
    Wire.write(decToBcd(year));                //nastaveni roku
    Wire.endTransmission();                    //konec spojeni
}

//ziskani casu
void readDS3231time(byte *second, byte *minute, byte *hour, byte *
    dayOfWeek, byte *dayOfMonth, byte *month, byte *year){
    Wire.beginTransmission(DS3231_I2C_ADDRESS); //navazani spojeni
    Wire.write(0);            //nastaveni ukazatele na registr sekund
    Wire.endTransmission();                  //konec spojeni
```

```
    Wire.requestFrom(DS3231_I2C_ADDRESS, 7); //pozadavek na 7 bajtu dat
    *second = bcdToDec(Wire.read() & 0x7f);
    *minute = bcdToDec(Wire.read());
    *hour = bcdToDec(Wire.read() & 0x3f);
    *dayOfWeek = bcdToDec(Wire.read());
    *dayOfMonth = bcdToDec(Wire.read());
    *month = bcdToDec(Wire.read());
    *year = bcdToDec(Wire.read());
}
```

3.3.3 Funkce setup

Mezi procedury vykonávané pouze při spuštění Arduina patří především inicializace komunikace a nastavení motorů a režimu pinů:

```
Wire.begin();           //komunikace po I2C
Serial.begin(115200);   //komunikace po seriove lince

//nehraje roli, ale musi byt zavolano
motorX.setSpeed(30);
motorY.setSpeed(30);

//nastaveni rezimu pinu
pinMode(xPin, INPUT);
pinMode(yPin, INPUT);
pinMode(pinCas, INPUT_PULLUP);
pinMode(pinMan, INPUT_PULLUP);
pinMode(hpPin, INPUT);
pinMode(hlPin, INPUT);
pinMode(dpPin, INPUT);
pinMode(dlPin, INPUT);
pinMode(trimr_u, INPUT);
pinMode(trimr_f, INPUT);
```

Arduino nabízí tři pracovní režimy pinů. Režim `OUTPUT` nastaví pin jako výstupní. V uvedeném kódu této možnosti není využito, protože veškeré výstupní piny (např. pro ovládání motorů nebo displeje) jsou nastaveny interně v rámci importovaných knihoven. Dalším režimem je `INPUT`, kdy pin pracuje naopak jako vstupní. Poslední možností je nastavení `INPUT_PULLUP` – s takovým nastavením Arduino připojí daný pin přes interní (pull-up) rezistor k hladině 5 V, čímž je umožněno vyslat signál tlačítkem (nebo spínačem) zapojeným mezi tento pin a hladinu GND (odpadá tedy nutnost použít pro nastavení proudu externí rezistor).

Následuje kontrola nastavení pracovního režimu založená právě na poloze přepínače (resp. tlačítka). Pokud je přepínač v určité poloze sepnut, propojí příslušný pin Arduina s hladinou GND. Takový stav je identifikován jako `LOW` (opakem je stav `HIGH`). Ke zjištění stavu pinů slouží funkce `digitalRead(pin)`. Zvolený pracovní režim je reprezentován čísly 1–3 a uložen do proměnné `rezim`.

```
//nastaveni pracovniho rezimu
if (digitalRead(pinCas) == LOW){
  rezim = 2;
  Serial.println("Rezim casovace");      //vypis na seriovou linku
}else if (digitalRead(pinMan) == LOW){
  rezim = 3;
  Serial.println("Manualni rezim");
}else{
  rezim = 1;
  Serial.println("Automaticky rezim");
}
```

V případě, že se tak nestalo již dříve, je třeba modulu reálného času předat aktuální datum a čas, které budou dále udržovány pomocí knoflíkové baterie. K nastavení času se používá již zmíněná funkce `setDS3231time`. Údaje na řádku 164 je třeba upravit dle skutečnosti. Po nahrání programu do Arduina dojde k nastavení času v okamžiku jeho spuštění. Poté je nutno provést úpravu spočívající v zakomentování řádku s touto funkcí (jak je v kódu uvedeno) a program nahrát znovu. Čas již bude díky baterii odpovídat skutečnosti (odchylka díky vnitřnímu oscilátoru dosahuje jednotek minut ročně).

```
//nastaveni casu (seconds, minutes, hours, day, date, month, year)
//setDS3231time(0,37,16,4,26,04,17);
```

Poté probíhá inicializace displeje, magnetometru a akcelerometru a nastavení potřebných parametrů. Pokud připojení magnetometru nebo akcelerometru selže, program bude provádět další pokusy, dokud se to nepodaří. Zároveň o tom vypíše informaci na displej a na sériovou linku. Funkce `delay` přijímá jako parametr počet milisekund, na které pozastaví vykonávání programu.

```
Serial.println("Nastaveni displeje");
lcd.begin(16, 2);

Serial.println("Spousteni kompasu");
while (!hmc.begin()){
  Serial.println("Kompas nenalezen!");
  lcd.clear();               //vyprazdneni displeje
```

```
  lcd.setCursor(0,0);          //nastaveni kurzoru
  lcd.print("Kompas");          //vypis na displej
  lcd.setCursor(0,1);          //presun kurzoru
  lcd.print("nenalezen!");
  delay(2000);                 //dvousekundova pauza
}
Serial.println("Spousteni akcelerometru");
while(!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G)){
  Serial.println("Akcelerometr nenalezen!");
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Akcelerometr");
  lcd.setCursor(0,1);
  lcd.print("nenalezen!");
  delay(2000);
}

hmc.setRange(HMC5883L_RANGE_1_3GA);           //nastaveni rozsahu
hmc.setMeasurementMode(HMC5883L_CONTINOUS); //nastaveni rezimu
hmc.setDataRate(HMC5883L_DATARATE_15HZ);     //frekvence mereni
hmc.setSamples(HMC5883L_SAMPLES_4);          //pocet vzorku pro vypocet
```

Je-li při spouštění stisknut příslušný spínač, jsou snímače polohy zkalibrovány v poloze nula (azimut i zenit 0°):

```
//proces kalibrace
if (digitalRead(pinKal) == LOW){
  hmc.setOffset(0, 0);     //magnetometr
  mpu.calibrateGyro();     //akcelerometr
}
```

Pro režim časovače je třeba rovněž připojit čtečku SD karet, získat aktuální datum a otevřít odpovídající soubor. Pokud opět něco selže, vypíše se informace na displej a sériovou linku.

```
if (rezim == 2){
  Serial.println("Pripojovani karty");
  while (!SD.begin(CS_PIN)){
    Serial.println("Kartu se nepodarilo pripojit.");
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Kartu nelze");
    lcd.setCursor(0,1);
    lcd.print("pripojit!");
    delay(2000);
```

```
  }
212 readDS3231time(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &
    month, &year);                    //ziskani data a casu
  sprintf(soubor, "%02d-%02d.txt", month, dayOfMonth);
214 file = SD.open(soubor);           //otevreni souboru pro dany den
  Serial.println("Otevirani souboru dat");
216 while (!file){
    Serial.println("Soubor nelze otevrit.");
218   lcd.clear();
    lcd.setCursor(0,0);
220   lcd.print("Soubor nelze");
    lcd.setCursor(0,1);
222   lcd.print("otevrit!");
    delay(2000);
224 }
```

Je-li vše v pořádku, program začne procházet datový soubor a ukládat si do paměti (do dvourozměrného pole nazvaného `pole`) potřebná data. Následně toto pole projde a do proměnných `azimut_n` a `zenit_n` uloží nastavení polohy pro daný čas.

```
  file.seek(0);                   //nastaveni bajtu v souboru
226 while (true) {                //cteni carkami oddelenych hodnot
    n = readField(&file, str, sizeof(str), ",\n");
228   if (n == 0) break;                      //konec souboru
    if (str[n-1] == ',' || str[n-1] == '\n') {
230   //i = sloupcovy index, j = radkovy index
        if (i == 0) j++;                      //novy radek
232     if (i == 3) i = -1;
        i = i + 1;
234     str[n-1] = 0;                         //odstraneni carky
    } else{
236     if (file.available()){
        Serial.print("Chyba!");
238   }else{                                  //konec souboru
        i = 0;
240   }
    }
242   float cislo = atof(str);                //prevod na cislo
    pole[j][i == 0 ? 3 : i - 1] = cislo;    //zapis do pole hodnot
244 }
  file.close();                               //uzavreni souboru
246 cas_h = 100 * hour + minute;             //aktualni cas
  cas_m = minute;
248 den_m = dayOfMonth;
  for (i = 0; i < zaznamu; i++){            //prochazeni pole zaznamu
```

```
250     if (100 * pole[i][0] + pole[i][1] <= cas_h){
          radek = i;                    //vyhledani radku dle aktualniho casu
252     }else{
          break;
254     }
      }
256   azimut_n = pole[radek][3];      //nastaveni pozadovaneho azimutu
      zenit_n = pole[radek][2];       //nastaveni pozadovaneho zenitu
258 }
    i = 0;                            //reset pomocne promenne
```

Pokud je přepínač režimu v poloze manuální, program na to upozorní výpisem na displej:

```
260 if (rezim == 3){              //manualni rezim
      lcd.clear();
262   lcd.setCursor(4,0);
      lcd.print("MANUALNI");
264   lcd.setCursor(5,1);
      lcd.print("REZIM!");
266 }
```

3.3.4 Funkce loop

Po vykonání funkce `setup()` program přejde k funkci `loop()`, kterou bude vykonávat v nekonečné smyčce. Nejprve načte data ze snímačů polohy a z joysticku (čtení hodnot na analogových vstupních pinech se provádí funkcí `analogRead(pin)`):

```
270 //cteni normalizovanych hodnot z akcelerometru
    Vector norm = hmc.readNormalize();
272 Vector acc = mpu.readNormalizeAccel();
    Vector gyr = mpu.readNormalizeGyro();
274
    //cteni hodnot z joysticku
276 xPozice = analogRead(xPin);
    yPozice = 1023 - analogRead(yPin);
```

Následně jsou ze získaných dat vypočítány potřebné úhly, azimut je upraven o deklinaci (dle vztahu 3.3) nastavenou na řádku 285. Úhly jsou současně převedeny z obloukové do stupňové míry.

```
    //vypocet uhlu (rad)
280 azimut = atan2(norm.YAxis, norm.XAxis);
```

```
akc_naklon = -(atan2(acc.XAxis, sqrt(acc.YAxis*acc.YAxis + acc.ZAxis*
    acc.ZAxis))*180.0)/PI;
zenit = -kalmanY.update(akc_naklon, gyr.YAxis);

//deklinace: (deg + (min / 60)) / (180 / PI);
float deklinace = (4 + (32 / 60)) / (180 / PI);
azimut += deklinace;

//korekce uhlu
if (azimut < 0){azimut += 2 * PI;}
if (azimut > 2 * PI){azimut -= 2 * PI;}
azimut = azimut * 180/PI;
```

Dále probíhá vyhodnocení dat z fotorezistorů, jsou počítány průměrné hodnoty pro dvojice fotorezistorů nahoře, dole, vlevo a vpravo a jejich rozdíly. Poté program načítá aktuální nastavení trimrů pro určení tolerance diferencí fotorezistorů (`necitlivost_f`) a tolerance nastavení úhlu v režimu časovače (`necitlivost_u`):

```
//ziskani hodnot z fotorezistoru a vypocet prumernych hodnot
hp = analogRead(hpPin);
hl = analogRead(hlPin);
dl = analogRead(dlPin);
dp = analogRead(dpPin);
nahore = (hp + hl) / 2;
dole = (dp + dl) / 2;
vlevo = (dl + hl) / 2;
vpravo = (hp + dp) / 2;
rozdilX = vlevo - vpravo;
rozdilY = nahore - dole;

//nacteni hodnot trimru
necitlivost_f = map(analogRead(trimr_f), 0, 1023, 1, 256);
necitlivost_u = map(analogRead(trimr_u), 0, 1023, 1, 360);
```

Hlavní část bloku `loop()` je konstrukce `switch(rezim)`, která na základě zvoleného režimu vyhodnocuje potřebu změny polohy zařízení a řídí pohyb motorů.

V automatickém režimu program každou třístou iteraci aktualizuje údaje na displeji: údaje z fotorezistorů a nastavenou hodnotu tolerance. Poté vyhodnocuje diference intenzity osvětlení v jednotlivých osách a pokud přesahují nastavenou toleranci, zahájí otáčení zařízení.

```
switch(rezim){
  case 1: //automaticky
    if (i >= 300){        //aktualizace displeje kazdou tristou iteraci
```

```
312         lcd.clear();
            lcd.setCursor(0,0);
314         lcd.print(hl);          //udaje z hornich fotorezistoru
            lcd.print("/");
316         lcd.print(hp);
            lcd.setCursor(9,0);
318         lcd.print("(~");
            lcd.print(necitlivost_f, 0);  //aktualni nastaveni tolerance
320         lcd.print(")");
            lcd.setCursor(0,1);
322         lcd.print(dl);          //udaje ze spodnich fotorezistoru
            lcd.print("/");
324         lcd.print(dp);
            lcd.setCursor(9,1);
326         lcd.print("(~");
            lcd.print(necitlivost_f, 0);  //opet tolerance
328         lcd.print(")");
            i = 0;                        //restart pocitadla iteraci
330       }
          if (rozdilX > necitlivost_f){   //kontrola proti toleranci
332         motorX.step(-1);              //otoceni motoru
            delay(1);
334       }else if (rozdilX < -necitlivost_f){
            motorX.step(1);               //opacny smer
336         delay(1);
          }
338       if (rozdilY > necitlivost_f){
            motorY.step(-1);
340         delay(1);
          }else if (rozdilY < -necitlivost_f){
342         motorY.step(1);
            delay(1);
344       }
        break;
```

Pro režim časovače je zapotřebí nejprve získat aktuální datum a čas pomocí funkce `readDS3231time`. Poté je porovnáno aktuální datum s platností dříve načteného souboru z SD karty. Pokud data nesouhlasí, Arduino se restartuje, čímž dojde k načtení nového souboru. Jestliže data souhlasí, program projde pole načtených hodnot a vyhledá časově odpovídající nastavení úhlů, které pak uloží do pracovních proměnných. Následuje výpis údajů na displej (opět každou třístou iteraci). Vypisují se informace o skutečných a požadovaných úhlech a nastavená tolerance. V poslední části je řízen pohyb motorů analogicky k předchozímu bloku kódu.

```
346   case 2: //casovac
```

```
  readDS3231time(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &
  month, &year);       //aktualni cas
348  if (den_m != dayOfMonth){        //po prechodu na novy den
    asm volatile ("  jmp 0");       //restart Arduina
350  }
  if (minute != cas_m){             //kontrola kazdou minutu
352    cas_m = minute;
    cas_h = 100 * hour + minute;
354    if (pole[radek + 1][0] != 0){    //kontrola pole dat
      if (100 * pole[radek + 1][0] + pole[radek + 1][1] <= cas_h){
356        radek++;
        azimut_n = pole[radek][3];  //nastaveni pozadovanych hodnot
358        zenit_n = pole[radek][2];
      }
360    }
  }
362  if (i >= 300){                    //kazda trista iterace
    lcd.clear();
364    lcd.setCursor(0,0);
    lcd.print(azimut, 1);           //vypis skutecneho azimutu
366    lcd.print("/");
    lcd.print(zenit, 1);            //vypis skutecneho zenitu
368    lcd.setCursor(12,0);
    lcd.print("~");
370    lcd.print(necitlivost_u, 0);    //tolerance uhlu
    lcd.setCursor(0,1);
372    lcd.print("~");
    lcd.print(azimut_n);            //pozadovany azimut
374    lcd.print("/");
    lcd.print(zenit_n);             //pozadovany zenit
376    i = 0;
  }
378  if (azimut_n > azimut + necitlivost_u){
    motorX.step(-1);
380    delay(1);
  }else if (azimut_n < azimut - necitlivost_u){
382    motorX.step(1);
    delay(1);
384  }
  if (zenit_n > zenit + necitlivost_u){
386    motorY.step(1);
    delay(1);
388  }else if (zenit_n < zenit - necitlivost_u){
    motorY.step(-1);
390    delay(1);
  }
```

```
392   break;
```

V posledním, manuálním režimu se pouze zjišťuje, zda je výchylka joysticku dostatečná, a řídí se pohyb motorů. Posledním příkazem funkce `loop()` je aktualizace počítadla iterací.

```
  case 3: //manualni
394   if (xPozice < 300){          //kontrola vychylky joysticku
        motorX.step(-1);
396     delay(1);
      }else if (xPozice > 700){
398     motorX.step(1);
        delay(1);
400   }
      if (yPozice < 300){
402     motorY.step(1);
        delay(1);
404   }else if (yPozice > 700){
        motorY.step(-1);
406     delay(1);
      }
408 break;
}
410 i++;                           //pripocitani iterace
```

Funkce pro práci s časem byly převzaty z webu [43], funkce `readField` je veřejně dostupná na fóru [44].

3.3.5 Varianty pohonů

Krokové motory

Uvedený kód dokáže ovládat unipolární i bipolární krokové motory pomocí knihovny `Stepper`, která je určena pro dvoufázové řízení bipolárních motorů. Je-li připojen unipolární motor, dokáže jej řídit také, ovšem je třeba zaměnit polohu druhého a třetího vodiče (v tomto případě provedeno softwarově na řádcích 40 a 41).

V případě potřeby řídit motory jinak je třeba knihovnu `Stepper` upravit. Změny způsobu řízení lze docílit nahrazením řádků 254–281 souboru `Stepper.cpp`. Tyto řádky definují sekvenci povelů pro krokování.

Pro zabezpečení jednofázového řízení unipolárního nebo bipolárního motoru je nutno uvedené řádky přepsat následujícím kódem:

```
254 if (this->pin_count == 4) {
```

```
  switch (thisStep) {
256   case 0:  // 1000
        digitalWrite(motor_pin_1, HIGH);
258     digitalWrite(motor_pin_2, LOW);
        digitalWrite(motor_pin_3, LOW);
260     digitalWrite(motor_pin_4, LOW);
    break;
262   case 1:  // 0010
        digitalWrite(motor_pin_1, LOW);
264     digitalWrite(motor_pin_2, LOW);
        digitalWrite(motor_pin_3, HIGH);
266     digitalWrite(motor_pin_4, LOW);
    break;
268   case 2:  // 0100
        digitalWrite(motor_pin_1, LOW);
270     digitalWrite(motor_pin_2, HIGH);
        digitalWrite(motor_pin_3, LOW);
272     digitalWrite(motor_pin_4, LOW);
    break;
274   case 3:  // 0001
        digitalWrite(motor_pin_1, LOW);
276     digitalWrite(motor_pin_2, LOW);
        digitalWrite(motor_pin_3, LOW);
278     digitalWrite(motor_pin_4, HIGH);
    break;
280 }
}
```

Řízení s poloviční krokem lze dosáhnout rozsáhlejší úpravou. Kód k nahrazení:

```
254 if (this->pin_count == 4) {
  switch (thisStep) {
256   case 0:  // 1000
        digitalWrite(motor_pin_1, HIGH);
258     digitalWrite(motor_pin_2, LOW);
        digitalWrite(motor_pin_3, LOW);
260     digitalWrite(motor_pin_4, LOW);
    break;
262   case 1:  // 1010
        digitalWrite(motor_pin_1, HIGH);
264     digitalWrite(motor_pin_2, LOW);
        digitalWrite(motor_pin_3, HIGH);
266     digitalWrite(motor_pin_4, LOW);
    break;
268   case 2:  // 0010
        digitalWrite(motor_pin_1, LOW);
```

```
270         digitalWrite(motor_pin_2, LOW);
            digitalWrite(motor_pin_3, HIGH);
272         digitalWrite(motor_pin_4, LOW);
        break;
274     case 3:   // 0110
            digitalWrite(motor_pin_1, LOW);
276         digitalWrite(motor_pin_2, HIGH);
            digitalWrite(motor_pin_3, HIGH);
278         digitalWrite(motor_pin_4, LOW);
        break;
280     case 4:   // 0100
            digitalWrite(motor_pin_1, LOW);
282         digitalWrite(motor_pin_2, HIGH);
            digitalWrite(motor_pin_3, LOW);
284         digitalWrite(motor_pin_4, LOW);
        break;
286     case 5:   // 0101
            digitalWrite(motor_pin_1, LOW);
288         digitalWrite(motor_pin_2, HIGH);
            digitalWrite(motor_pin_3, LOW);
290         digitalWrite(motor_pin_4, HIGH);
        break;
292     case 6:   // 0001
            digitalWrite(motor_pin_1, LOW);
294         digitalWrite(motor_pin_2, LOW);
            digitalWrite(motor_pin_3, LOW);
296         digitalWrite(motor_pin_4, HIGH);
        break;
298     case 7:   // 1001
            digitalWrite(motor_pin_1, HIGH);
300         digitalWrite(motor_pin_2, LOW);
            digitalWrite(motor_pin_3, LOW);
302         digitalWrite(motor_pin_4, HIGH);
        break;
304   }
}
```

Dále je pro mikrokrokování nutno nahradit řádek 224 uvedeného souboru kódem:

```
224 stepMotor(this->step_number % 8);
```

Stejnosměrné motory

Při použití stejnosměrných motorů je z uvedeného kódu možné odstranit všechny podbarvené řádky mimo těch v bloku `loop()`, ty budou nahrazeny jiným kódem. Do části `setup()` je třeba přidat následující kód, který nastaví příslušné piny jako výstupní a připojí je na hladinu 5 V:

```
pinMode(PinX1, OUTPUT);
pinMode(PinX2, OUTPUT);
pinMode(PinX3, OUTPUT);
digitalWrite(PinX1, HIGH);
digitalWrite(PinX2, HIGH);
digitalWrite(PinX3, HIGH);
```

Úprava kódu v bloku `loop()` spočívá v nahrazení řádků 331–337 tímto kódem:

```
if (rozdilX > necitlivost_f){
  digitalWrite(PinX2, HIGH);          //rozepnuti rele REL2
  digitalWrite(PinX3, HIGH);          //rozepnuti rele REL3
  delay(200);                         //pauza na prestaveni kontaktu
  digitalWrite(PinX1, LOW);           //sepnuti rele REL1
}else if (rozdilX < -necitlivost_f){
  digitalWrite(PinX2, LOW);           //sepnuti rele REL2
  digitalWrite(PinX3, LOW);           //sepnuti rele REL3
  delay(200);                         //pauza na prestaveni kontaktu
  digitalWrite(PinX1, LOW);           //sepnuti rele REL1
}else{                                //zastaveni motoru
  digitalWrite(PinX1, HIGH);          //rozepnuti rele REL1
  digitalWrite(PinX2, HIGH);          //rozepnuti rele REL2
  digitalWrite(PinX3, HIGH);          //rozepnuti rele REL3
}
```

Tyto úpravy předpokládají zapojení dle schématu na obr. 3.17.

Obdobné úpravy je samozřejmě třeba provést i v částech kódu vztahujících se ke zbylým dvěma pracovním režimům a k druhému z dvojice motorů.

Servomotory

Využití servomotorů vyžaduje importovat namísto knihovny `Stepper` knihovnu `Servo`, řádek 2 tedy bude nahrazen výrazem:

```
#include <Servo.h>
```

Změny dozná pochopitelně i nastavení motorů na řádcích 40 a 41:

```
40 Servo motorX;
   Servo motorY;
```

Zároveň je nutno definovat nové proměnné ještě před blokem `setup()`:

```
int naklon = 0;           //vychozi uhel naklonu
int natoceni = 0;         //vychozi uhel natoceni
int naklon_min = 0;       //minimalni uhel naklonu
int naklon_max = 90;      //maximalni uhel naklonu
int natoceni_min = 0;     //minimalni uhel natoceni
int natoceni_max = 180;   //maximalni uhel natoceni
```

Řádky 136 a 137 ve funkci `setup()` budou nahrazeny kódem:

```
136 motorX.attach(PinX0);     //pripojeni serva k pinu PinX0
    motorY.attach(PinY0);     //pripojeni serva k pinu PinY0
```

Pohyb servomotorů je řízen nastavením konkrétního požadovaného úhlu natočení, který musí být ve stanovených mezích, aby nedošlo k mechanickému poškození motoru.

Úprava kódu v sekci `loop()` spočívá v následující úpravě řádků 331–337:

```
if (rozdilX > necitlivost_f){
332   if (natoceni < natoceni_max){       //kontrola limitu
        natoceni = natoceni + 1;          //zvetseni uhlu natoceni
334     motorX.write(natoceni);           //pokyn k pohybu motoru
        delay(1000);                      //pauza
336   }
}else if (rozdilX < -necitlivost_f){
338   if (natoceni > natoceni_min){       //kontrola limitu
        natoceni = natoceni - 1;          //zmenseni uhlu natoceni
340     motorX.write(natoceni);           //pokyn k pohybu motoru
        delay(1000);                      //pauza
342   }
}
```

Analogickým způsobem je zapotřebí upravit kód pro ostatní pracovní režimy a pro motor náklonu.

3.3.6 Volitelné příslušenství

Koncové spínače

Připojení koncových spínačů je velice jednoduché – jeden kontakt spínače je připojen k hladině GND, druhý kontakt ke vstupnímu pinu, který je třeba v programu nejprve definovat:

```
int konecL = 29;        //pin 29
int konecP = 31;        //pin 31
```

Dále je nutné v bloku `setup()` tyto piny definovat jako vstupní a připojené přes interní rezistory k hladině 5 V:

```
pinMode(konecL, INPUT_PULLUP);
pinMode(konecP, INPUT_PULLUP);
```

Řízení pohybu motorů v části `loop()` potom nejprve ověřuje, jestli se zařízení již nenachází v krajní poloze. Pokud ano, pohyb je umožněn jen v opačném směru. Odpovídající náhrada řádků 331–337 tedy spočívá v doplnění podmínek:

```
if (rozdilX > necitlivost_f){
  if (digitalRead(konecP) != LOW){     //kontrola koncoveho spinace
    motorX.step(-1)
    delay(1);
  }
}else if (rozdilX < -necitlivost_f){
  if (digitalRead(konecL) != LOW){     //kontrola koncoveho spinace
    motorX.step(1);
    delay(1);
  }
}
```

Podobnou úpravu je třeba opět provést i pro ostatní pracovní režimy. Pokud je koncovými spínači vybaveno i otáčení v druhé ose, připojí se tyto v souladu se schématem na obr. 3.15 k pinům 33 a 35.

Potenciometry

Připojení potenciometru pro snímání úhlu natočení nebo náklonu vychází ze schématu na obr. 3.13. Jako vstup poslouží pin A8 (případně i A9). Dále je třeba definovat proměnné `natoceni`, `natoceni_min` a `natoceni_max` stejně jako při použití serva:

```
int potX = A8;              //potenciometr natoceni na pinu A8
int natoceni = 0;           //promenna pro uhel natoceni
int natoceni_min = 0;       //minimalni uhel natoceni
int natoceni_max = 170;     //maximalni uhel natoceni
```

Nastavení pinu jako vstupního v části `setup()`:

```
pinMode(potX, INPUT);
```

V oddíle `loop()` je třeba na začátek (např. za řádek 278) vložit kód pro zjištění aktuálního natočení, převod na úhel a jeho uložení do příslušné proměnné:

```
natoceni = map(analogRead(potX), 70, 920, 0, 220);
```

K řízení motorů potom přibudou stejné podmínky jako v případě užití servomotorů:

```
if (rozdilX > necitlivost_f){
  if (natoceni < natoceni_max){        //kontrola limitu
    motorX.step(-1);
    delay(1);
  }
}else if (rozdilX < -necitlivost_f){
  if (natoceni > natoceni_min){        //kontrola limitu
    motorX.step(1);
    delay(1);
  }
}
```

3.3.7 Spínání trackeru

Arduino ve verzi Mega 2560 nedokáže samo sebe vypnout a zapnout. Proto lze pro spínání trackeru využít běžný soumrakový spínač a pomocí něj připojovat a odpojovat napájecí napětí Arduina. Další možností je využití časového spínače. Ten lze naprogramovat pomocí údajů o časech východu a západu slunce, které jsou umístěny na přiložené CD.

Tracker je zároveň možné ponechat vypnutý a zapínat např. v desetiminutových intervalech vždy jen na krátkou dobu, kdy dojde k vyhodnocení polohy a případné úpravě natočení solárního zařízení. K tomuto účelu lze použít např. časové relé nebo obvod s časovačem 555.

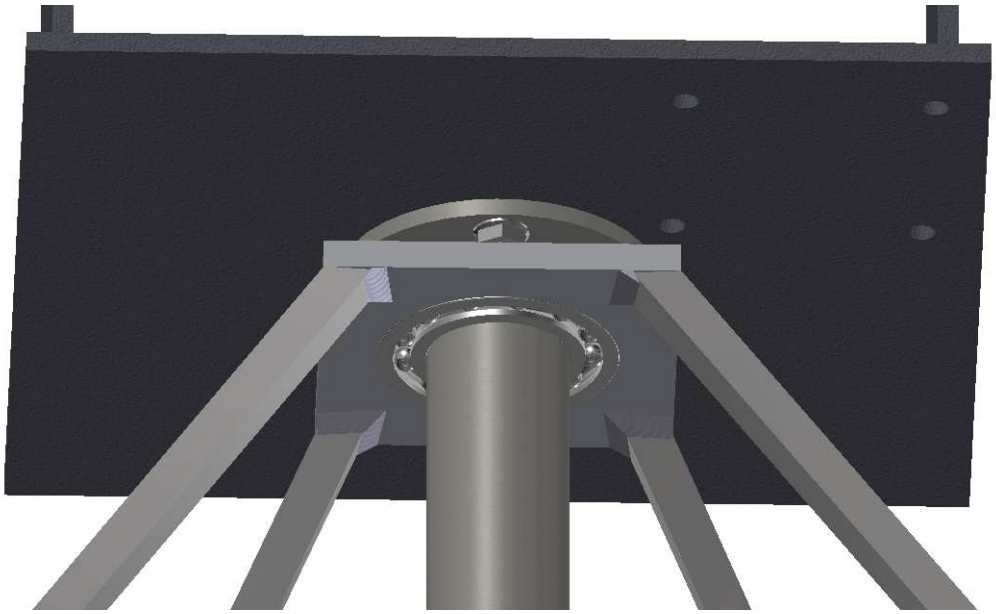
3.4 Pohyblivý stojan solárního zařízení

Výchozím zařízením dle zadání práce je diskový solární koncentrátor vyrobený ze satelitní paraboly o průměru přibližně 60 cm. Konkávní plocha paraboly je pokryta odraznou hliníkovou vrstvou.

Návrh pohyblivého stojanu pro tento koncentrátor předkládá obr. 3.18. Základem je ocelová trubka o průměru 40 mm pohyblivě připojená pomocí dvou ložisek (axiálního dole a radiálního nahoře – detail je zachycen na obr. 3.19), resp. ložiskových domků k železné desce o rozměrech 500 × 500 mm. K horní části trubky je přivařena příruba, ke které je přišroubována horní železná deska o rozměrech 270 × 250 mm. Vertikální pohyb zajišťuje trubka o průměru 20 mm uchycená pomocí dvou radiálních kuličkových ložisek.

Obr. 3.18: Model pohyblivého stojanu pro malý koncentrátor

Zdroj: autor



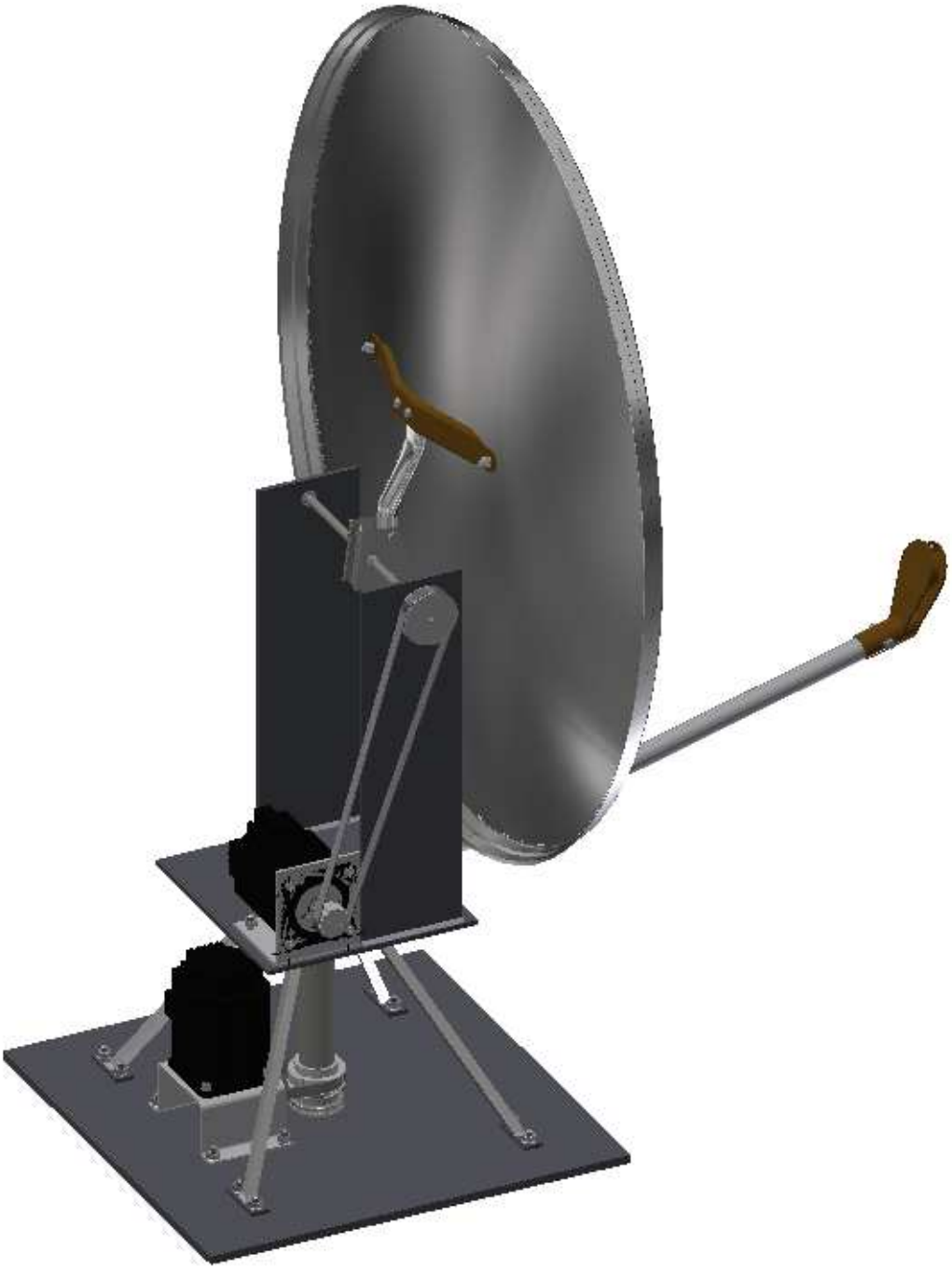
Obr. 3.19: Detail ložiska pohyblivého stojanu

Zdroj: autor

Návrh kompletního zařízení (stojanu včetně paraboly a pohonů) znázorňuje obr. 3.20.

Vzhledem k velikosti navrženého stojanu a hmotnosti koncentrátoru navrhuji pro pohon použít stejnosměrné motory s planetovou převodovkou s výstupními otáčkami max. 2 ot./min a momentem alespoň 1 Nm. Těmto požadavkům vyhovují např. motory série PG421 [45].

Přenos momentu může být proveden klínovým, drážkovým nebo ozubeným řemenem, ozubenými koly nebo přímým připojením motoru k hřídeli. Pro jednoduchost navrhuji využití drážkového řemene.



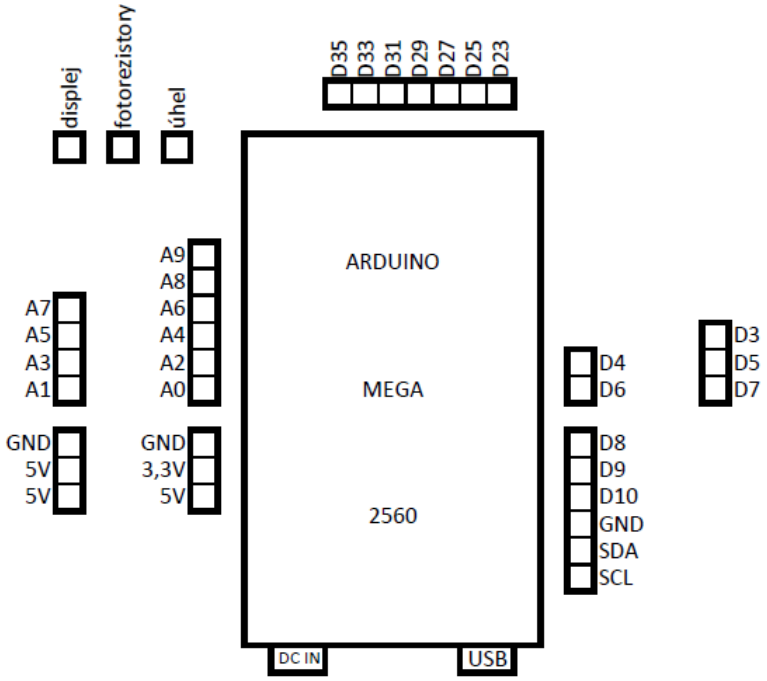
Obr. 3.20: Model kompletní sestavy koncentrátoru včetně pohonu

Zdroj: autor

4 KONSTRUKCE, OVĚŘENÍ FUNKČNOSTI A ANALÝZA

4.1 Konstrukce trackeru

Původní návrh desky plošných spojů vizualizovaný na obr. 3.16 předpokládá poměrně přesnou oboustrannou desku s prokovenými otvory. Výroba takové desky vyžaduje specializované vybavení (což se odráží i na ceně služby), proto byla vyrobena deska modifikovaná, která oproti předloženému návrhu neobsahuje rezervní svorkovnice a namísto svorkovnic pro displej a snímače polohy je u těchto komponent využito pevně připojených vodičů. Tyto úpravy umožnily vyrobit desku jako jednostrannou z univerzálního polotovaru (laboratorní plošný spoj). Rozložení jednotlivých svorkovnic a popis svorek jsou uvedeny na obr. 4.1 a zároveň přiloženy k desce samotné.



A0 joystick X
A1 joystick Y
A2 fotorezistor HP
A3 fotorezistor HL
A4 fotorezistor DL
A5 fotorezistor DP
A6
A7
A8 potenciometr natočení
A9 potenciometr náklonu

D3 motor X, 0
D4 motor X, 1
D5 motor X, 2
D6 motor X, 3
D7 motor Y, 0
D8 motor Y, 1
D9 motor Y, 2
D10 motor Y, 3

D23 spínač: časovač
D25 spínač: manuální
D27 kalibrace
D29 koncový spínač č. 1
D31 koncový spínač č. 2
D33 koncový spínač č. 3
D35 koncový spínač č. 4

Obr. 4.1: Rozložení svorkovnic a popis svorek zjednodušené desky

Zdroj: autor

Deska byla osazena potřebnými komponentami a spolu s nimi uložena do montážní krabice o vnějších rozměrech 18 × 23 × 9 cm s krytím IP 65. Snímek výsledného zařízení je zachycen na obr. 4.2.

Obr. 4.2: Hotový tracker s krytem

Zdroj: autor

4.2 Konstrukce pohyblivého stojanu

Konstrukci malého stojanu dle uvedeného návrhu (obr. 3.18) nebylo možné realizovat z důvodu náročného obrábění (a s tím souvisejících nároků na strojní vybavení) a cenově nedostupných komponent (zejména ložisek). Byl však zhotoven stojan obdobné konstrukce opatřený dvojicí stejnosměrných (stěračových) motorů s momentem 1,2 Nm. Přenos síly je zajištěn čelními ozubenými koly vyrobenými z akrylátu, převodový poměr je u obou motorů přibližně 1:11. Stojan je vybaven potenciometrem pro snímání náklonu a dvěma koncovými spínači pro kontrolu rozsahu úhlu natočení.

Hotový stojan s připevněným koncentrátorem předkládají obrázky 4.3 a 4.4.

Obr. 4.3: Pohyblivý stojan včetně koncentrátoru (přední pohled)

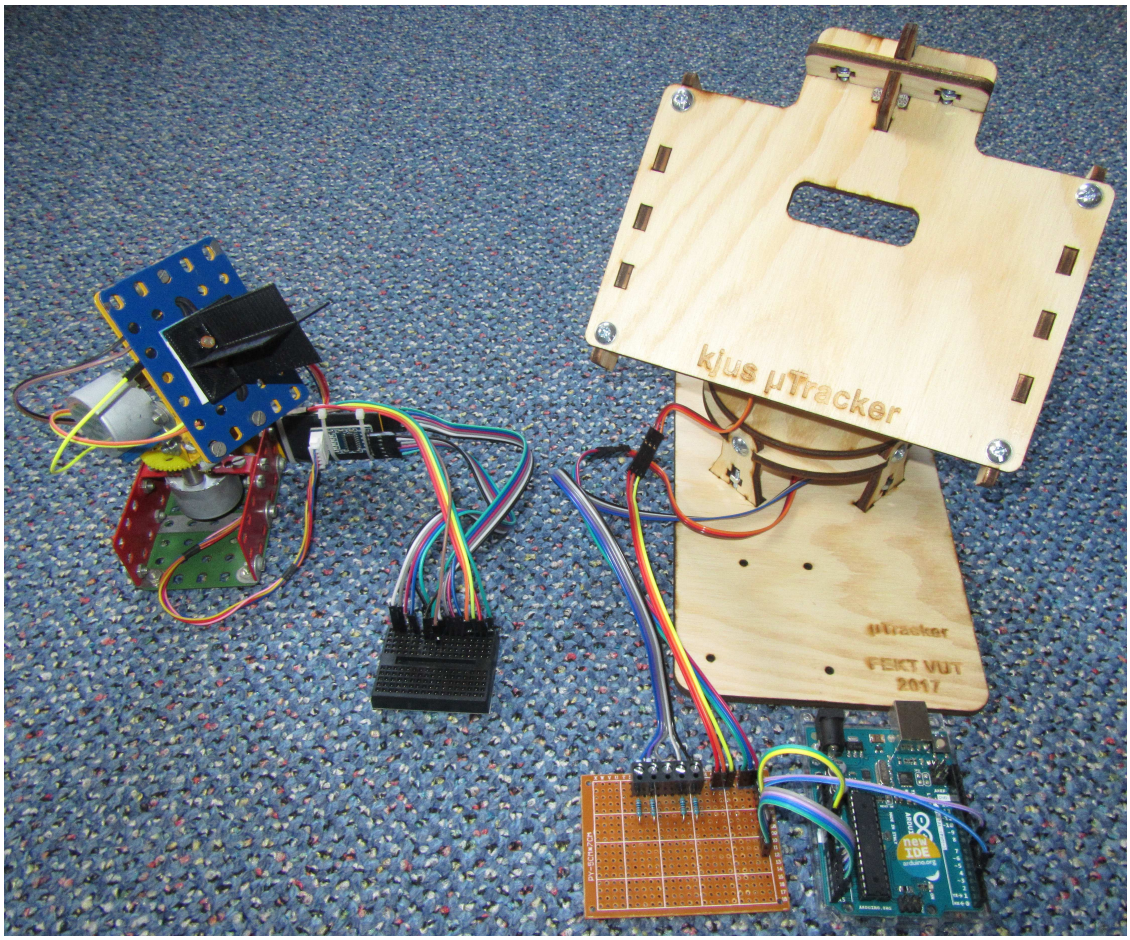
Zdroj: autor

Obr. 4.4: Pohyblivý stojan včetně koncentrátoru (boční pohled)

Zdroj: autor

Po montáži koncentrátoru s malým tepelným výměníkem došlo při otáčení k vylomení několika zubů umělohmotných ozubených kol, akrylát navíc neodolal účinkům slunečního záření a prohnul se, stojan této konstrukce tedy bylo možné využít pouze při vývoji trackeru.

Pro účely prezentace trackeru byly vytvořeny dvě miniatury stojanů s funkcí sledování slunce zachycené na obr. 4.5. První z nich využívá kovový stojan, který uvádí v pohyb dvojice krokových motorů. Druhý stojan[1] je dřevěný a k pohybu slouží dva modelářské servomotory.



Obr. 4.5: Prezentační model sledovacího zařízení s krokoými motory (vlevo) a se servomotory (vpravo)

Zdroj: autor

[1]Software použitý k výrobě druhého stojanu poskytl Joshua Zimmerman [46].

4.3 Ověření funkčnosti trackeru

Ověření jednotlivých funkcí trackeru bylo prováděno průběžně při jejich implementaci na sestrojeném stojanu s talířovým koncentrátorem. Testování odhalilo, že magnetometr HMC5883L, a především akcelerometr MPU6050 jsou v reálných podmínkách (mimo laboratoř) natolik ovlivněny elektrosmogem a dalšími vnějšími vlivy, že se stávají pro danou aplikaci nepoužitelnými. Zbylé komponenty však při testování pracovaly bezchybně a tracker byl shledán funkčním s tím, že přesnou polohu je v případě potřeby nutno snímat pomocí potenciometrů.

Vzhledem k tomu, že sestrojený stojan není schopen dlouhodobějšího provozu bez nutnosti zásahu a tracker bylo třeba otestovat na reálném, funkčním (a zároveň energeticky významném) zařízení, byl tracker otestován na malé soukromé fotovoltaické elektrárně poblíž Košic. Tato elektrárna se skládá z pohyblivého stojanu nesoucího čtyři fotovoltaické panely o výkonu 280 Wp (je však dimenzována na osm takových panelů). Pohyb je zajištěn malými stejnosměrnými motory s trojstupňovými planetovými převodovkami a standardně je řízen integrovanými obvody a tranzistorovými H-můstky.

Přibližně hodinu trvající testování trackeru na tomto zařízení prokázalo schopnost trackeru řídit v automatickém režimu pohyb motorů tak, aby zařízení sledovalo slunce na obloze. Dále byla vyzkoušena funkčnost manuálního ovládání včetně reakcí na dosažení mezních poloh (pomocí koncových spínačů). Rovněž bylo zjištěno, že zařízení nemusí být před východem slunce orientováno na východ, aby tracking správně fungoval, kód byl proto aktualizován a tato funkce byla odstraněna. Nepodařilo se vyzkoušet fungování v režimu časovače, neboť stojan není osazen potenciometry. Tento režim byl však již dříve úspěšně otestován na vlastním stojanu s talířovým koncentrátorem.

Po přerušení a následném obnovení napájecího napětí nedošlo k žádnému problému nebo neočekávané akci. K potížím nedošlo ani po pokropení krabice trackeru vodou ze zahradní hadice (simulace deště).

Tracker byl napájen napětím 12 V a odebíral proud 20–70 mA, což znamená, že jeho příkon se pohyboval v rozmezí 0,2–0,8 W. Při napětí 5 V (USB) se hodnoty[2] snížily prakticky na polovinu: 30 mA; 0,15 W.

4.4 Ekonomická analýza

Pro vyhodnocení účinnosti sledování slunce za použití vyrobeného trackeru by bylo zapotřebí mít k dispozici dvě naprosto identická solární zařízení ve stejný čas na stejném místě, kdy jedno by bylo ovládáno trackerem, druhé by bylo ve fixní poloze.

[2]Hodnoty byly v obou případech změřeny multimetrem UNI-T UT203.

Určující by potom byl rozdíl v energetickém zisku obou zařízení. Pro absenci druhého zařízení nemohlo být toto vyhodnocení provedeno. Obecně však lze očekávat, že rozdíl v intenzitě slunečního záření bude s trackováním v automatickém režimu přibližně odpovídat hodnotám prezentovaným na obr. 1.13, resp. v tab. 1.2.

Ze zkušeností majitele zmíněné fotovoltaické elektrárny lze při dané konfiguraci díky dvouosému trackování dosáhnout energetického zisku až 5,6 kWh/den. Před úpravou stojanu na pohyblivý bylo při stejné konfiguraci údajně dosaženo maximální hodnoty 4,5 kWh/den, což znamená nárůst o přibližně 24 % díky sledování[3]. Tyto údaje však nebylo možné ověřit, neboť by bylo třeba provést dlouhodobé měření jak s trackerem, tak bez něj.

Energie potřebná pro zajištění pohybu je určena především použitými motory, spotřebu samotného trackeru lze v tomto případě zanedbat. Při napětí 12 V byl proudový odběr[4] motoru pro natáčení 2,0 A, motoru pro náklon potom 2,4 A. Motory jsou (při rozumném nastavení tolerance trimrem) v pohybu vždy jednotky, nanejvýš desítky vteřin, následně jsou několik minut v klidu až do další změny orientace zařízení. Proto je možné z hlediska nákladů na provoz zanedbat i spotřebu motorů. Je nicméně nutné zajistit zdroj schopný dodat potřebný proud pro otáčení.

Při osazení stojanu osmi panely by celkový instalovaný výkon dosáhl hodnoty 2,24 kWp. S takovým výkonem by bylo bez trackování dle kalkulátoru americké Národní laboratoře pro obnovitelné zdroje (odkud pochází i dříve zmiňovaný algoritmus pro výpočet polohy slunce) možné v dané lokalitě (Košice) ročně vyrobit průměrně 2,25 MWh energie [47]. Pokud by tracker skutečně zvýšil zisk o 24 %, vyrobená energie by dosáhla hodnoty 2,79 MWh/rok. Tento údaj se příliš neliší od výpočtu stejného kalkulátoru při změně zadání z fixní instalace na dvouosé natáčení (2,75 MWh/rok) [47].

Podle údajů o průměrných cenách elektřiny pro domácnosti z roku 2015 by tento nárůst energie znamenal úsporu přibližně 82 € ročně na Slovensku (0,152 €/kWh) a 70 € ročně v České republice (0,129 €/kWh) [48]. Při průměrné ceně energie v EU 0,211 €/kWh by se pak jednalo o úsporu 114 €/rok [48].

4.4.1 Výrobní náklady

Náklady na zhotovení trackeru činí 6020 Kč, položkový rozpis je uveden v tab. 4.1. Uvedené ceny zahrnují DPH a byly získány pouze u českých dodavatelů. Při nákupu v zahraničních obchodech by celková cena mohla být nižší o přibližně 40–70 %.

[3]Rozdíl oproti hodnotám znázorněným na obr. 1.13 a v tab. 1.2 je způsoben především výrazně odlišnou zeměpisnou polohou.

[4]Hodnoty byly opět změřeny multimetrem UNI-T UT203.

Tab. 4.1: Cenová kalkulace trackeru

Položka	**Cena** (Kč)
výroba DPS	1600
komponenty trackeru	
Arduino Mega 2560	1100
HMC5883L	120
MPU6050	120
čtečka karet + karta	280
DS3231 + baterie	110
joystick	80
displej	100
spínače	160
ostatní součástky	500
čidlo slunečního svitu	
3D tisk	450
fotorezistory	200
průhledná kopule	460
montážní krabice	490
kryt trackeru	250
Celkem	**6020**

Zdroj: autor

Náklady na výrobu pohyblivého stojanu (nebo úpravu stojanu na pohyblivý) jsou závislé na konkrétním solárním zařízení, nelze je tedy paušálně vyčíslit. Náklady na stojan navržený v části 3.4 (obr. 3.18) však činí přibližně 1900 Kč včetně DPH (z toho více než polovinu částky činí ložiska), motory a převody potom přibližně 7000 Kč.

Podstatného snížení nákladů však lze dosáhnout zakoupením již použitých součástek (motory z autovrakoviště, hutní materiál, ložiska a převodovky z kovošrotu). V takovém případě mohou být náklady na pohyblivý stojan sníženy až o 90 %. Pohyblivý stojan zmíněné fotovoltaické elektrárny byl dle majitele tímto způsobem zhotoven za cenu přibližně 170 €.

V uvedených cenách není zahrnuta mechanická práce/pronájem strojů – lze očekávat částku v řádu stovek až jednotek tisíc Kč.

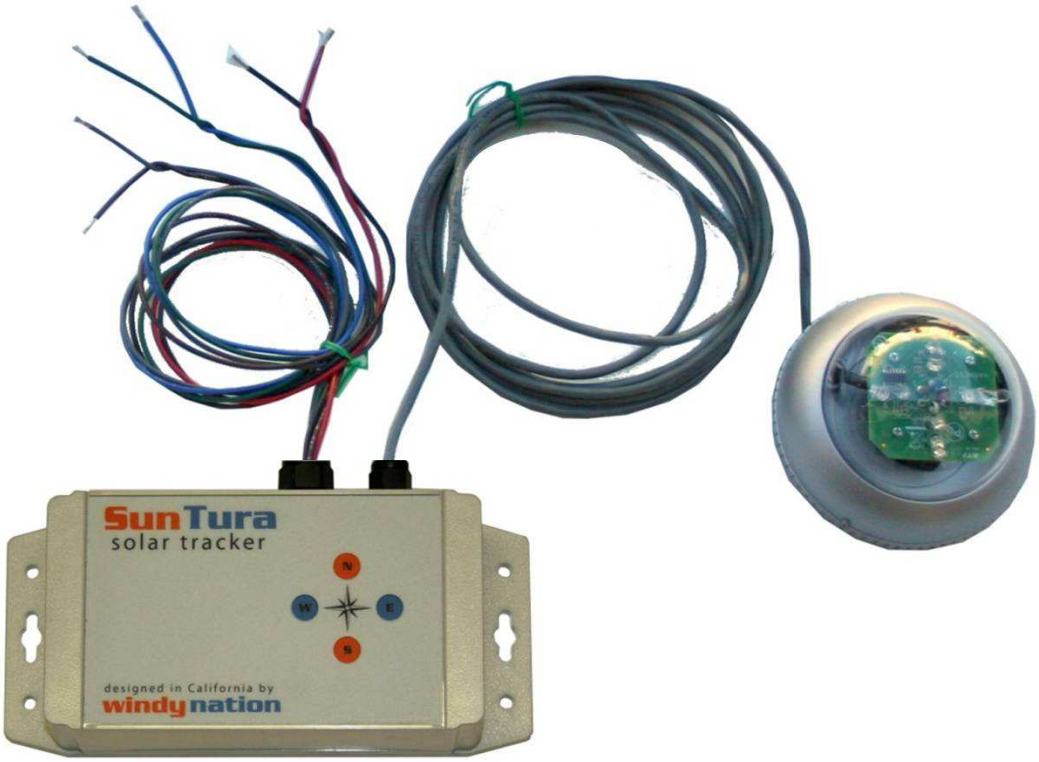
4.4.2 Komerční systémy

U cenového srovnání navrženého trackeru s komerčně dostupnými řešeními je nutno poznamenat, že uvedená kalkulace je spočítána pro jeden exemplář, při hromadné výrobě by mohla být cena dále snížena (především náklady na výrobu DPS). Současně ale v kalkulaci nejsou zahrnuty náklady na vývoj a sestavení trackeru.

Samostatné trackery patrně na českém trhu k dispozici nejsou, zahraniční obchody však několik výrobků nabízí. Pravděpodobně nejlevnějším trackerem na trhu je polský výrobek v ceně přibližně 680 Kč, který však slouží pouze k jednoosému sledování [49].

Dalším jednoosým sledovačem je výrobek americké firmy MTM Scientific v ceně 1285 Kč, který je dodáván jako polotovar (pouze DPS, bez krabice, bez čidla slunečního svitu) [50].

Dvouosý tracker americké výroby SunTura lze včetně optického čidla zakoupit přibližně za 6360 Kč [51]. Tento tracker (zachycený na obr. 4.6) oproti výše uvedeným modelům umožňuje i manuální režim ovládání.



Obr. 4.6: Tracker SunTura s čidlem slunečního svitu

Zdroj: [51]

Všechna výše uvedená zařízení vyhodnocují polohu slunce na základě intenzity slunečního svitu a všechna umožňují pouze přímé připojení stejnosměrných motorů (resp. lineárních aktuátorů).

Český trh však nabízí hotová řešení ve formě fotovoltaické elektrárny, kde součástí dodávky je i tracker a pohyblivý stojan včetně motorů. V této kategorii lze zakoupit např. systém Sunmaster Trekr 5 o celkovém instalovaném výkonu 5,6 kWp (24 panelů). Cena tohoto řešení je 833 000 Kč [52]. Využívá se zde opět čidel intenzity slunečního záření.

Obdobným řešením je produkt TERMS Tracker TT50 s instalovaným výkonem 10,08 kWp (42 panelů). Ten naopak využívá datového souboru s předem vypočítanými směry natočení. Lze jej pořídit za cenu 1 129 000 Kč [53].

V zahraničí lze nalézt alternativu v podobě trackeru a pohyblivého stojanu, ovšem bez fotovoltaických panelů (nebo jiných solárních zařízení). Slovinský dodavatel SAT CONTROL nabízí řešení pro čtyři panely (1 kWp) za cenu 19 500 Kč. Cena největší varianty, zařízení pro 15 panelů (3,75 kWp), je potom 113 000 Kč [54]. Tyto trackery rovněž využívají předem vypočítané polohy slunce.

5 ZÁVĚR

Cílem této práce bylo zdokumentovat současně dostupné technologie solárních zařízení a možnosti sledování slunce, navrhnout automatický sledovací systém založený na platformě Arduino a vytvořit funkční model sestavy solárního zařízení a trackeru.

V souladu s uvedenými cíli byl zdokumentován současný stav techniky v oblasti aktivního zpracování energie slunečního záření a byly zanalyzovány technologie sledování slunce a jejich aplikace; podrobněji byla představena metoda teoretického výpočtu polohy slunce včetně vhodného algoritmu. Na základě získaných poznatků byl následně vypracován návrh automatického solárního trackeru disponujícího velkou měrou variability kompatibilních solárních zařízení.

Navržený sledovač umožňuje pohyb ve dvou osách s možností připojení různých druhů pohonů zahrnujících krokové motory, stejnosměrné motory a servomotory. Pro sledování slunce je možnost volit mezi implementací uvedeného algoritmu výpočtu a automatickým vyhodnocováním intenzity osvětlení. Návrh trackeru využívá platformu Arduino a klade důraz na rozšiřitelnost systému jednak možností připojit různé doplňkové příslušenství, jednak dostatkem vstupně-výstupních konektorů pro případné budoucí doplnění o další komponenty a funkce.

V rámci řešení práce byl zpracován i návrh pohyblivého stojanu pro diskový solární koncentrátor, jeho konstrukce však stejně jako konstrukce navrženého trackeru narazila na omezení vyplývající z nedostatečného strojního vybavení, proto bylo při sestavení stojanu a trackeru přistoupeno k několika zjednodušením. Stojan byl následně využit pro testování zhotoveného trackeru, který byl poté důkladněji otestován na reálném zařízení – fotovoltaické elektrárně – a byl shledán funkčním.

Pro účely prezentace byly rovněž vytvořeny dva zmenšené modely sledovacích zařízení s krokovými motory a servomotory.

V závěrečné části práce je uvedena ekonomická analýza provozu trackeru a jeho výrobních nákladů, je nabídnuto i porovnání s komerčními systémy.

Pro případnou navazující práci navrhuji zaměření na zvýšení spolehlivosti snímačů polohy (nebo využití zcela jiného systému) nebo na návrh a konstrukci pohyblivého stojanu solárních zařízení z mechanického hlediska, který by po osazení trackerem představoval komplexní řešení problému zvýšení účinnosti solárních zařízení.

LITERATURA

[1] MASTNÝ, Petr, Jan MORÁVEK a Jiří PITRON. *Malé zdroje elektrické energie* [online]. Brno, 2015 [cit. 2016-12-25]. Dostupné z: e-learningu VUT

[2] Sun Fact Sheet. *NASA Space Science Data Coordinated Archive* [online]. 2016 [cit. 2016-12-25]. Dostupné z: http://nssdc.gsfc.nasa.gov/planetary/factsheet/sunfact.html

[3] CODDINGTON, O., J. L. LEAN, P. PILEWSKIE, M. SNOW a D. LINDHOLM. A Solar Irradiance Climate Data Record. In: *Bulletin of the American Meteorological Society* [online]. Boston: American Meteorological Society, 2016, **97**(7), s. 1265-1282 [cit. 2017-05-17]. DOI: 10.1175/BAMS-D-14-00265.1. ISSN 0003-0007. Dostupné z: http://journals.ametsoc.org/doi/10.1175/BAMS-D-14-00265.1

[4] QUASCHNING, Volker. *Obnovitelné zdroje energií*. Praha: Grada, 2010. Stavitel. ISBN 978-80-247-3250-3.

[5] REMMERS, Karl-Heinz. *Velká solární zařízení: Úvod k navrhování a provozu*. Brno: ERA, 2007. Technická knihovna (ERA). ISBN 978-80-7366-110-6.

[6] Termické solární kolektory (panely). *Solární-energie.info* [online]. Praha [cit. 2016-12-11]. Dostupné z: http://www.solarni-energie.info/termicke-solarni-panely-kolektory.php

[7] SCHLAICH, Jörg, Rudolf BERGERMANN, Wolfgang SCHIEL a Gerhard WEINREBE. Design of Commercial Solar Updraft Tower Systems: Utilization of Solar Induced Convective Flows for Power Generation. *Journal of Solar Energy Engineering*. 2005, **127**(1), 117-124. DOI: 10.1115/1.1823493. ISSN 01996231. Dostupné také z: http://SolarEnergyEngineering.asmedigitalcollection.asme.org/article.aspx?articleid=1457227

[8] *Energy & Resources Predictions 2012* [online]. Deloitte Global Services Limited, c2011 [cit. 2016-12-25]. Dostupné z: https://www2.deloitte.com/content/dam/Deloitte/ie/Documents/Energy%20Resources/EnergyandResourcespredictions2012_DeloitteIreland_EnergyResources.pdf

[9] MILLS, David R. a Graham L. MORRISON. Compact Linear Fresnel Reflector solar thermal powerplants. *Solar Energy*. 2000, **68**(3), 263-283.

DOI: 10.1016/S0038-092X(99)00068-7. ISSN 0038092x. Dostupné také z: http://linkinghub.elsevier.com/retrieve/pii/S0038092X99000687

[10] DEY, C. J. Heat transfer aspects of an elevated linear absorber. *Solar Energy*. 2004, **76**(1-3), 243-249. DOI: 10.1016/j.solener.2003.08.030. ISSN 0038092x. Dostupné také z: http://linkinghub.elsevier.com/retrieve/pii/S0038092X03002962

[11] Understanding solar concentrators. *Appropedia: the sustainability wiki* [online]. 2013 [cit. 2016-12-26]. Dostupné z: http://www.appropedia.org/Understanding_solar_concentrators

[12] MANCINI, Thomas, Peter HELLER, Barry BUTLER, et al. Dish-Stirling Systems: An Overview of Development and Status. *Journal of Solar Energy Engineering*. 2003, **125**(2), 135-151. DOI: 10.1115/1.1562634. ISSN 01996231. Dostupné také z: http://SolarEnergyEngineering.asmedigitalcollection.asme.org/article.aspx?articleid=1456732

[13] ŠIMONOVÁ, Lucie. *Distribuční soustava Kypru: realizovatelnost obnovitelných zdrojů a přenos energie* [online]. Brno, 2011 [cit. 2016-12-25]. Dostupné z: webu VUT. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Vedoucí práce Jan Dolenský.

[14] XU, Chao, Zhifeng WANG, Xin LI a Feihu SUN. Energy and exergy analysis of solar power tower plants. *Applied Thermal Engineering*. 2011, **31**(17-18), 3904-3913. DOI: 10.1016/j.applthermaleng.2011.07.038. ISSN 13594311. Dostupné také z: http://linkinghub.elsevier.com/retrieve/pii/S135943111100398X

[15] Neuer Weltrekord in der Konzentratorphotovoltaik: 36,7 Prozent für Solarmodul aus höchst effizienten Mehrfachsolarzellen. *Fraunhofer-Institut für Solare Energiesysteme ISE* [online]. Freiburg, 2014 [cit. 2016-12-26]. Dostupné z: https://www.ise.fraunhofer.de/de/presse-und-medien/presseinformationen/presseinformationen-2014/neuer-weltrekord-in-der-konzentratorphotovoltaik

[16] MOUSAZADEH, Hossein, Alireza KEYHANI, Arzhang JAVADI, Hossein MOBLI, Karen ABRINIA a Ahmad SHARIFI. A review of principle and sun-tracking methods for maximizing solar systems output. *Renewable and Sustainable Energy Reviews*. 2009, **13**(8), 1800-1818. DOI: 10.1016/j.rser.2009.01.022. ISSN 13640321. Dostupné také z: http://linkinghub.elsevier.com/retrieve/pii/S1364032109000318

[17] VOROBIEV, Pavel Yu., Jesús GONZÁLEZ-HERNÁNDEZ a Yuri V. VOROBIEV. Optimization of the solar energy collection in tracking and non-tracking photovoltaic solar system. In: *(ICEEE). 1st International Conference on Electrical and Electronics Engineering, 2004*. IEEE, 2004, s. 310-314. DOI: 10.1109/ICEEE.2004.1433900. ISBN 0-7803-8531-4. Dostupné také z: http://ieeexplore.ieee.org/document/1433900/

[18] CLIFFORD, M. J. a D. EASTWOOD. Design of a novel passive solar tracker. *Solar Energy*. 2004, **77**(3), 269-280. DOI: 10.1016/j.solener.2004.06.009. ISSN 0038092x. Dostupné také z: http://linkinghub.elsevier.com/retrieve/pii/S0038092X04001483

[19] POULEK, Vladislav Testing the new solar tracker with shape memory alloy actors. In: *Proceedings of 1994 IEEE 1st World Conference on Photovoltaic Energy Conversion - WCPEC (A Joint Conference of PVSC, PVSEC and PSEC)*. IEEE, 1994, s. 1131-1133. DOI: 10.1109/WCPEC.1994.520161. ISBN 0-7803-1460-3. Dostupné také z: http://ieeexplore.ieee.org/document/520161/

[20] Inspira's CPV Sun Tracking. *Concentrator photovoltaics*. Berlin: Springer, c2007, s. 221-251. ISBN 3540687963.

[21] ABDALLAH, Salah a Salem NIJMEH. Two axes sun tracking system with PLC control. *Energy Conversion and Management*. 2004, **45**(11-12), 1931-1939. DOI: 10.1016/j.enconman.2003.10.007. ISSN 01968904. Dostupné také z: http://linkinghub.elsevier.com/retrieve/pii/S0196890403003066

[22] POULEK, Vladislav a Martin LIBRA. A very simple solar tracker for space and terrestrial applications. *Solar Energy Materials and Solar Cells*. 2000, **60**(2), 99-103. DOI: 10.1016/S0927-0248(99)00071-9. ISSN 09270248. Dostupné také z: http://linkinghub.elsevier.com/retrieve/pii/S0927024899000719

[23] WALRAVEN, Robert. Calculating the position of the sun. *Solar Energy*. 1978, **20**(5), 393-397. DOI: 10.1016/0038-092X(78)90155-X. ISSN 0038092x. Dostupné také z: http://linkinghub.elsevier.com/retrieve/pii/0038092X7890155X

[24] ANDRLE, Pavel. *Základy nebeské mechaniky*. Praha: Academia, 1971.

[25] KOPERNÍK, Mikuláš. *Obehy nebeských sfér*. Bratislava: Veda, 1974. Dejiny vied a techniky.

[26] MARKVART, Tomas, ed. *Solar Electricity*. 2. ed. Chichester: John Wiley, 2008. ISBN 978-047-1988-533.

[27] JENKINS, Alejandro. The Sun's position in the sky. *European Journal of Physics*. 2013, **34**(3), 633-652. DOI: 10.1088/0143-0807/34/3/633. ISSN 0143-0807. Dostupné také z: http://stacks.iop.org/0143-0807/34/i=3/a=633?key=crossref.f86660dff90ed959d2fe719abd1ccdbd

[28] JACOBSON, Mark Z. *Fundamentals of atmospheric modeling*. 2nd ed. Cambridge: Cambridge University Press, 2005. ISBN 05-215-4865-9.

[29] REDA, Ibrahim a Afshin ANDREAS. Solar Position Algorithm for Solar Radiation Applications. *Solar Energy*. 2004, **76**(5), 577-589. DOI: 10.1016/j.solener.2003.12.003. ISSN 0038092x. Dostupné také z: http://linkinghub.elsevier.com/retrieve/pii/S0038092X0300450X

[30] KLECZEK, Josip. *Velká encyklopedie vesmíru*. Praha: Academia, 2002s.,48s.barev.obr.příl. ISBN 80-200-0906-X.

[31] IERS Bulletin-A. *IERS Rapid Service/Prediction Center for Earth Orientation* [online]. Washington, D.C., 2016 [cit. 2017-01-03]. Dostupné z: http://maia.usno.navy.mil/ser7/ser7.dat

[32] ŠUTA, Václav a Jan BERNKOPF. Knihovna pro krokový motor. *Ústav radioelektroniky* [online]. Brno, 2009 [cit. 2017-04-19]. Dostupné z: http://www.urel.feec.vutbr.cz/MIA/2009/suta/

[33] ŘEZÁČ, Kamil. Krokové motory: princip funkce, metody řízení. *Robotika.cz* [online]. 2002 [cit. 2017-04-19]. Dostupné z: https://robotika.cz/articles/steppers/cs

[34] TKOTZ, Klaus. *Příručka pro elektrotechnika*. 2. dopl. vyd. Praha: Europa-Sobotáles, 2006, 623 s. ISBN 80-867-0613-3.

[35] Řízení otáček ss elektromotoru. In: *MediaWiki SPŠ a VOŠ Písek* [online]. Písek: SPŠ a VOŠ Písek, 2010 [cit. 2017-04-19]. Dostupné z: http://wiki.sps-pi.cz/index.php/%C5%98%C3%ADzen%C3%AD_ot%C3%A1%C4%8Dek_ss_elektromotoru

[36] HUGHES, Austin a Bill DRURY. *Electric motors and drives: fundamentals, types and applications*. 4th ed. Amsterdam: Elsevier, 2013. ISBN 978-0-08-098332-5.

[37] How to get the PWM frequency and duration of each pulse? In: *Electrical Engineering Stack Exchange* [online]. New York, NY, 2017 [cit. 2017-04-20]. Dostupné z: https://electronics.stackexchange.com/questions/129961/how-to-get-the-pwm-frequency-and-duration-of-each-pulse

[38] Magnetic Declination: Find the magnetic declination at your location. *Magnetic-Declination.com* [online]. 2015 [cit. 2017-04-21]. Dostupné z: http://magnetic-declination.com/

[39] JARZĘBSKI, Korneliusz. Kompensacja przechyleń kompasów cyfrowych. */dev/Jarzębski: Santyago back to root* [online]. Bytom, 2015 [cit. 2017-04-21]. Dostupné z: http://www.jarzebski.pl/arduino/rozwiazania-i-algorytmy/kompensacja-przechylen-kompasow-cyfrowych.html

[40] RICHTER, Miloslav. Kalmanův filtr. *Materiály k vyučovaným předmětům* [online]. Brno, 2016-02-05 [cit. 2017-04-21]. Dostupné z: http://www.uamt.feec.vutbr.cz/~richter/vyuka/0910_mpov/tmp/kalman_filter.html.cs

[41] JARZĘBSKI, Korneliusz. Odczyty Pitch & Roll oraz filtr Kalmana. In: */dev/Jarzębski: Santyago back to root* [online]. Bytom, 2015 [cit. 2017-04-21]. Dostupné z: http://www.jarzebski.pl/media/zoom/publish/2014/10/mpu6050-processing-2.png

[42] Jarzebski (Korneliusz Jarzębski) / Repositories. *GitHub: The world's leading software development platform* [online]. San Francisco, 2017 [cit. 2017-04-21]. Dostupné z: https://github.com/jarzebski?tab=repositories

[43] BOXALL, John. Tutorial – Using DS1307 and DS3231 Real-time Clock Modules with Arduino. *Tronixstuff* [online]. 2014 [cit. 2017-05-02]. Dostupné z: http://tronixstuff.com/2014/12/01/tutorial-using-ds1307-and-ds3231-real-time-clock-modules-with-arduino/

[44] A Simple Function for Reading CSV Text Files. In: *Arduino Forum* [online]. 2015 [cit. 2017-05-02]. Dostupné z: https://forum.arduino.cc/index.php?topic=340849.0

[45] DC motor série PG421 s planetovou převodovkou. *StejnosměrnéMotory.cz: prodej malých stejnosměrných motorů s převodovkou* [online]. Brno, 2017 [cit. 2017-05-04]. Dostupné z: http://www.dcmotory.cz/dc-motory-planetova-prevodovka/serie-pg421.html

[46] *Brown Dog Gadgets* [online]. Brookfield, 2017 [cit. 2017-05-09]. Dostupné z: https://www.browndoggadgets.com/

[47] *PVWatts Calculator* [online]. Golden, Colorado: National Renewable Energy Laboratory, 2016 [cit. 2017-05-10]. Dostupné z: http://pvwatts.nrel.gov/

[48] *Energy, transport, and environment indicators: 2016 edition* [online]. Luxembourg: Publications Office of the European Union, c2016 [cit. 2017-05-10]. Theme 7–Transport. ISBN 978-92-79-60137-8. Dostupné z: http://ec.europa.eu/eurostat/en/web/products-statistical-books/-/KS-DK-16-001

[49] Solar tracker śledzenie słońca - bateria słoneczna: solar tracker - sterownik - śledzenie słońca w osi wschód-zachód. *Cosmogate: sklep z gadżetami elektronicznymi* [online]. Przeźmierowo, 2003 [cit. 2017-05-10]. Dostupné z: http://cosmogate.pl/product_info.php/products_id/394

[50] Solar Tracker Circuit Kit. *MTM Scientific, Inc.* [online]. Clinton, Michigan [cit. 2017-05-10]. Dostupné z: http://www.mtmscientific.com/solartracker.html

[51] SunTura Dual-Axis Solar Tracker Kit. *Windy Nation* [online]. Ventura, California [cit. 2017-05-10]. Dostupné z: http://www.windynation.com/Solar-Trackers/SunTura-Dual-Axis-Solar-Tracker-Kit/-/238

[52] Sunmaster Trekr 5. *Konex: solární panely, solární systémy, fotovoltaické panely* [online]. Šumperk [cit. 2017-05-10]. Dostupné z: http://www.solarobchod.cz/cz/solarni-sestavy-k-svepomocne-montazi/solarni-fotovoltaicke-systemy-k-svepomocne-montazi/sunmaster-trekr-5/

[53] TERMS Tracker TT50 - PANASONIC 10kWp. *Eshop.TERMS.eu* [online]. České Budějovice, 2013 [cit. 2017-05-10]. Dostupné z: http://eshop.terms.eu/cz/e-shop/966450/c59481-terms-tracker/terms-tracker-tt50-panasonic-10kwp.html

[54] Solar Tracker 2-axis ST44M3V15P w backstr. for 15 pan., 3.75kWp / 0153 / ST44M3V15P, without concrete block. *SAT CONTROL: Solar Tracker & Sun Tracker & Actuator for Solar Tracker* [online]. Cerklje [cit. 2017-05-10]. Dostupné z: http://www.solar-motors.com/gb/solar-tracker-2-axis-st44m3v15p-w-backstr-for-15-pan-3-75kwp-0153-st44m3v15p-without-concrete-block-i232.shtml

SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

ABS	akrylonitrilbutadienstyren
AC	Alternating Current
BCD	Binary Coded Decimal; dvojkově reprezentované dekadické číslo
CD	Compact Disc; kompaktní disk
DC	Direct Current
DMP	Digital Motion Processor
DPH	daň z přidané hodnoty
DPS	deska plošných spojů
ENTSO-E	European Network of Transmission System Operators for Electricity; Evropská síť provozovatelů přenosových elektrizačních soustav
EU	Evropská unie
GPS	Global Positioning System
HSV	Hue, Saturation, Value; odstín, sytost, jas
I^2C	Inter-Integrated Circuit
IP	Ingress Protection; stupeň krytí
LCD	Liquid Crystal Display; displej z tekutých krystalů
OZE	obnovitelné zdroje energie
PE	polyetylén
PLC	Programmable Logic Controller; programovatelný logický automat
PP	polypropylén
PWM	Pulse Width Modulation; pulzně šířková modulace
SCL	Synchronous Clock
SD	Secure Digital
SDA	Synchronous Data
SMA	Shape-memory alloy; slitina s tvarovou pamětí
SPA	Solar Position Algorithm
SPI	Serial Peripheral Interface
TT	Terrestrial Time; terestrický čas
TUV	teplá užitková voda
USB	Universal Serial Bus; univerzální sériová sběrnice
UT	Universal Time; světový čas
UTC	Coordinated Universal Time; koordinovaný světový čas
UV	ultra violet; ultrafialová
Wi-Fi	wireless fidelity

Ostatní symboly a zkratky se vyskytují v textu s jejich okamžitým vysvětlením.

SEZNAM PŘÍLOH

A Data pro výpočet polohy slunce dle SPA **105**

B Svorkovnice na desce plošných spojů **113**

C Základní zdrojový kód pro Arduino **119**

D Řízení sledovače pomocí zpracování obrazu **129**

E Obsah přiloženého CD **131**

A DATA PRO VÝPOČET POLOHY SLUNCE DLE SPA

Tab. A.1: Data pro výpočet heliocentrických souřadnic Země

Výraz	Iterace	A	B	C
L_0	0	175 347 046	0	0
	1	3 341 656	4,669 257	6283,076
	2	34 894	4,6261	12 566,15
	3	3497	2,7441	5753,385
	4	3418	2,8289	3,5231
	5	3136	3,6277	77 713,77
	6	2676	4,4181	7860,419
	7	2343	6,1352	3930,21
	8	1324	0,7425	11 506,77
	9	1273	2,0371	529,691
	10	1199	1,1096	1577,344
	11	990	5,233	5884,927
	12	902	2,045	26,298
	13	857	3,508	398,149
	14	780	1,179	5223,694
	15	753	2,533	5507,553
	16	505	4,583	18 849,23
	17	492	4,205	775,523
	18	357	2,92	0,067
	19	317	5,849	11 790,63
	20	284	1,899	796,298
	21	271	0,315	10 977,08
	22	243	0,345	5486,778
	23	206	4,806	2544,314
	24	205	1,869	5573,143
	25	202	2,458	6069,777
	26	156	0,833	213,299
	27	132	3,411	2942,463
	28	126	1,083	20,775
	29	115	0,645	0,98
	30	103	0,636	4694,003
	31	102	0,976	15 720,84

	32	102	4,267	7,114
	33	99	6,21	2146,17
	34	98	0,68	155,42
	35	86	5,98	161 000,7
	36	85	1,3	6275,96
	37	85	3,67	71 430,7
	38	80	1,81	17 260,15
	39	79	3,04	12 036,46
	40	75	1,76	5088,63
	41	74	3,5	3154,69
	42	74	4,68	801,82
	43	70	0,83	9437,76
	44	62	3,98	8827,39
	45	61	1,82	7084,9
	46	57	2,78	6286,6
	47	56	4,39	14 143,5
	48	56	3,47	6279,55
	49	52	0,19	12 139,55
	50	52	1,33	1748,02
	51	51	0,28	5856,48
	52	49	0,49	1194,45
	53	41	5,37	8429,24
	54	41	2,4	19 651,05
	55	39	6,17	10 447,39
	56	37	6,04	10 213,29
	57	37	2,57	1059,38
	58	36	1,71	2352,87
	59	36	1,78	6812,77
	60	33	0,59	17 789,85
	61	30	0,44	83 996,85
	62	30	2,74	1349,87
	63	25	3,16	4690,48
L_1	0	628 331 966 747	0	0
	1	206 059	2,678 235	6283,076
	2	4303	2,6351	12 566,15
	3	425	1,59	3,523
	4	119	5,796	26,298
	5	109	2,966	1577,344

	6	93	2,59	18 849,23
	7	72	1,14	529,69
	8	68	1,87	398,15
	9	67	4,41	5507,55
	10	59	2,89	5223,69
	11	56	2,17	155,42
	12	45	0,4	796,3
	13	36	0,47	775,52
	14	29	2,65	7,11
	15	21	5,34	0,98
	16	19	1,85	5486,78
	17	19	4,97	213,3
	18	17	2,99	6275,96
	19	16	0,03	2544,31
	20	16	1,43	2146,17
	21	15	1,21	10 977,08
	22	12	2,83	1748,02
	23	12	3,26	5088,63
	24	12	5,27	1194,45
	25	12	2,08	4694
	26	11	0,77	553,57
	27	10	1,3	6286,6
	28	10	4,24	1349,87
	29	9	2,7	242,73
	30	9	5,64	951,72
	31	8	5,3	2352,87
	32	6	2,65	9437,76
	33	6	4,67	4690,48
L_2	0	52 919	0	0
	1	8720	1,0721	6283,076
	2	309	0,867	12 566,15
	3	27	0,05	3,52
	4	16	5,19	26,3
	5	16	3,68	155,42
	6	10	0,76	18 849,23
	7	9	2,06	77 713,77
	8	7	0,83	775,52
	9	5	4,66	1577,34

	10	4	1,03	7,11
	11	4	3,44	5573,14
	12	3	5,14	796,3
	13	3	6,05	5507,55
	14	3	1,19	242,73
	15	3	6,12	529,69
	16	3	0,31	398,15
	17	3	2,28	553,57
	18	2	4,38	5223,69
	19	2	3,75	0,98
L_3	0	289	5,844	6283,076
	1	35	0	0
	2	17	5,49	12 566,15
	3	3	5,2	155,42
	4	1	4,72	3,52
	5	1	5,3	18 849,23
	6	1	5,97	242,73
L_4	0	114	3,142	0
	1	8	4,13	6283,08
	2	1	3,84	12 566,15
L_5	0	1	3,14	0
B_0	0	280	3,199	84 334,66
	1	102	5,422	5507,553
	2	80	3,88	5223,69
	3	44	3,7	2352,87
	4	32	4	1577,34
B_1	0	9	3,9	5507,55
	1	6	1,73	5223,69
R_0	0	100 013 989	0	0
	1	1 670 700	3,098 464	6283,076
	2	13 956	3,055 25	12 566,15
	3	3084	5,1985	77 713,77
	4	1628	1,1739	5753,385
	5	1576	2,8469	7860,419
	6	925	5,453	11 506,77
	7	542	4,564	3930,21
	8	472	3,661	5884,927

	9	346	0,964	5507,553
	10	329	5,9	5223,694
	11	307	0,299	5573,143
	12	243	4,273	11 790,63
	13	212	5,847	1577,344
	14	186	5,022	10 977,08
	15	175	3,012	18 849,23
	16	110	5,055	5486,778
	17	98	0,89	6069,78
	18	86	5,69	15 720,84
	19	86	1,27	161 000,7
	20	65	0,27	17 260,15
	21	63	0,92	529,69
	22	57	2,01	83 996,85
	23	56	5,24	71 430,7
	24	49	3,25	2544,31
	25	47	2,58	775,52
	26	45	5,54	9437,76
	27	43	6,01	6275,96
	28	39	5,36	4694
	29	38	2,39	8827,39
	30	37	0,83	19 651,05
	31	37	4,9	12 139,55
	32	36	1,67	12 036,46
	33	35	1,84	2942,46
	34	33	0,24	7084,9
	35	32	0,18	5088,63
	36	32	1,78	398,15
	37	28	1,21	6286,6
	38	28	1,9	6279,55
	39	26	4,59	10 447,39
R_1	0	103 019	1,107 49	6283,076
	1	1721	1,0644	12 566,15
	2	702	3,142	0
	3	32	1,02	18 849,23
	4	31	2,84	5507,55
	5	25	1,32	5223,69
	6	18	1,42	1577,34

	7	10	5,91	10 977,08
	8	9	1,42	6275,96
	9	9	0,27	5486,78
R_2	0	4359	5,7846	6283,076
	1	124	5,579	12 566,15
	2	12	3,14	0
	3	9	3,63	77 713,77
	4	6	1,87	5573,14
	5	3	5,47	18 849,23
R_3	0	145	4,273	6283,076
	1	7	3,92	12 566,15
R_4	0	4	2,56	6283,08

Tab. A.2: Data pro výpočet nutace

Iterace	$\mathbf{Y_0}$	$\mathbf{Y_1}$	$\mathbf{Y_2}$	$\mathbf{Y_3}$	$\mathbf{Y_4}$	**a**	**b**	**c**	**d**
1	0	0	0	0	1	−171 996	−174,2	92 025	8,9
2	−2	0	0	2	2	−13 187	−1,6	5736	−3,1
3	0	0	0	2	2	−2274	−0,2	977	−0,5
4	0	0	0	0	2	2062	0,2	−895	0,5
5	0	1	0	0	0	1426	−3,4	54	−0,1
6	0	0	1	0	0	712	0,1	−	0
7	−2	1	0	2	2	−517	1,2	224	−0,6
8	0	0	0	2	1	−386	−0,4	200	0
9	0	0	1	2	2	−301	0	129	−0,1
10	−2	−1	0	2	2	217	−0,5	−95	0,3
11	−2	0	1	0	0	−158	0	0	0
12	−2	0	0	2	1	129	0,1	−70	0
13	0	0	−1	2	2	123	0	−53	0
14	2	0	0	0	0	63	0	0	0
15	0	0	1	0	1	63	0,1	−33	0
16	2	0	−1	2	2	−59	0	26	0
17	0	0	−1	0	1	−58	−0,1	32	0
18	0	0	1	2	1	−51	0	27	0
19	−2	0	2	0	0	48	0	0	0
20	0	0	−2	2	1	46	0	−24	0
21	2	0	0	2	2	−38	0	16	0

22	0	0	2	2	2	−31	0	13	0
23	0	0	2	0	0	29	0	0	0
24	−2	0	1	2	2	29	0	−12	0
25	0	0	0	2	0	26	0	0	0
26	−2	0	0	2	0	−22	0	0	0
27	0	0	−1	2	1	21	0	−10	0
28	0	2	0	0	0	17	−0, 1	0	0
29	2	0	−1	0	1	16	0	−8	0
30	−2	2	0	2	2	−16	0,1	7	0
31	0	1	0	0	1	−15	0	9	0
32	−2	0	1	0	1	−13	0	7	0
33	0	−1	0	0	1	−12	0	6	0
34	0	0	2	−2	0	11	0	0	0
35	2	0	−1	2	1	−10	0	5	0
36	2	0	1	2	2	−8	0	3	0
37	0	1	0	2	2	7	0	−3	0
38	−2	1	1	0	0	−7	0	0	0
39	0	−1	0	2	2	−7	0	3	0
40	2	0	0	2	1	−7	0	3	0
41	2	0	1	0	0	6	0	0	0
42	−2	0	2	2	2	6	0	−3	0
43	−2	0	1	2	1	6	0	−3	0
44	2	0	−2	0	1	−6	0	3	0
45	2	0	0	0	1	−6	0	3	0
46	0	−1	1	0	0	5	0	0	0
47	−2	−1	0	2	1	−5	0	3	0
48	−2	0	0	0	1	−5	0	3	0
49	0	0	2	2	1	−5	0	3	0
50	−2	0	2	0	1	4	0	0	0
51	−2	1	0	2	1	4	0	0	0
52	0	0	1	−2	0	4	0	0	0
53	−1	0	1	0	0	−4	0	0	0
54	−2	1	0	0	0	−4	0	0	0
55	1	0	0	0	0	−4	0	0	0
56	0	0	1	2	0	3	0	0	0
57	0	0	−2	2	2	−3	0	0	0
58	−1	−1	1	0	0	−3	0	0	0
59	0	1	1	0	0	−3	0	0	0
60	0	−1	1	2	2	−3	0	0	0

61		2	−1	−1	2	2	−3	0	0	0
62		0	0	3	2	2	−3	0	0	0
63		2	−1	0	2	2	−3	0	0	0

B SVORKOVNICE NA DESCE PLOŠNÝCH SPOJŮ

Tab. B.1: Popis jednotlivých svorkovnic na navržené desce plošných spojů

Č.	Pin Arduina	Pin komponenty	Použití
MOTOR_X			
1	5V	—	pomocné napájení
2	GND	—	
3	D6	—	ovládání motoru
4	D5	—	
5	D4	—	
6	D3	—	
MOTOR_Y			
1	5V	—	pomocné napájení
2	GND	—	
3	D10	—	ovládání motoru
4	D9	—	
5	D8	—	
6	D7	—	
KONCOVE_SPINACE			
1	D29	—	koncový spínač č. 1
2	GND	—	
3	D31	—	koncový spínač č. 2
4	GND	—	
5	D33	—	koncový spínač č. 3
6	GND	—	
7	D35	—	koncový spínač č. 4
8	GND	—	
POTENCIOMETRY			
1	5V	—	potenciometr pro snímání natočení
2	A8	—	
3	GND	—	
4	5V	—	potenciometr pro snímání náklonu
5	A9	—	
6	GND	—	

FOTOREZISTORY			
1	5V	—	spojené vývody fotorezistorů
2	A5	—	fotorezistor DP
3	A4	—	fotorezistor DL
4	A3	—	fotorezistor HL
5	A2	—	fotorezistor HP
DS3231			
1	—	32K	—
2	—	SQW	—
3	SCL	SCL	I^2C
4	SDA	SDA	
5	3.3V	VCC	napájení
6	GND	GND	
SD			
1	GND	GND	napájení
2	3.3V	3.3V	
3	5V	5V	
4	D2	CS	SPI
5	D51	MOSI	
6	D52	SCK	
7	D50	MISO	
8	—	C.DET	—
MPU6050			
1	—	INT	—
2	5V	AD0	určení adresy akcelerometru
3	—	XCL	—
4	—	XDA	—
5	SDA	SDA	I^2C
6	SCL	SCL	
7	GND	GND	napájení
8	5V	VCC	

HMC5883L			
1	3.3V	VCC	napájení
2	GND	GND	
3	SCL	SCL	I^2C
4	SDA	SDA	
5	—	DRDY	—
REZIM			
1	GND	—	přepínač režimu
2	D23	—	
3	D25	—	
4	GND	—	spínač kalibrace
5	D27	—	
DISPLEJ1			
1	GND	VSS	napájení displeje
2	5V	VDD	
3	(TR_D)	V0	nastavení kontrastu
4	D39	RS	řízení řadiče
5	GND	RW	
6	D41	E	
DISPLEJ2			
1	D43	D4	data
2	D45	D5	
3	D47	D6	
4	D49	D7	
5	(R_D)	A	napájení podsvícení
6	GND	K	
JOYSTICK			
1	5V	+5V	napájení
2	GND	GND	
3	A0	VRx	osa x
4	A1	VRy	osa y

V_A			
1	A15	—	volné analogové vstupy
2	A14	—	
3	A13	—	
4	A12	—	
5	A11	—	
6	A10	—	
V_D1			
1	D48	—	volné digitální vstupy/výstupy
2	D46	—	
3	D44	—	
4	D42	—	
5	D40	—	
6	D38	—	
7	D36	—	
8	D34	—	
9	D32	—	
10	D30	—	
V_D2			
1	D28	—	volné digitální vstupy/výstupy
2	D26	—	
3	D24	—	
4	D22	—	
5	D37	—	
5V			
1	5V	—	napájení 5 V (rezerva)
2			
3			
4			
3.3V			
1	3.3V	—	napájení 3,3 V (rezerva)
2			
3			
4			

GND			
1	GND	—	napájení GND (rezerva)
2			
3			
4			
SCL_SDA			
1	SCL	—	I^2C (rezerva)
2			
3	SDA	—	
4			

C ZÁKLADNÍ ZDROJOVÝ KÓD PRO ARDUINO

```
#include <Wire.h>                //komunikace po I2C
#include <Stepper.h>             //krokove motory
#include <HMC5883L.h>            //magnetometr
#include <SPI.h>                 //komunikace po SPI
#include <SD.h>                  //ctecka SD karet
#include <LiquidCrystal.h>       //LCD
#include <MPU6050.h>             //akcelerometr
#include <KalmanFilter.h>        //Kalmanuv filtr
#define DS3231_I2C_ADDRESS 0x68  //nastaveni adresy casoveho modulu
#define CS_PIN 2                 //nastaveni datoveho pinu ctecky

int xPin = A0;       //osa x joysticku
int yPin = A1;       //osa y joysticku
int hpPin = A2;      //fotorezistor HP
int hlPin = A3;      //fotorezistor HL
int dlPin = A4;      //fotorezistor DL
int dpPin = A5;      //fotorezistor DP
int pinCas = 23;     //prepinac rezimu v poloze casovace
int pinMan = 25;     //prepinac rezimu v poloze manualni
int pinKal = 27;     //spinac kalibrace
int PinX0 = 3;       //piny motoru nataceni (osa x)
int PinX1 = 4;
int PinX2 = 5;
int PinX3 = 6;
int PinY0 = 7;       //piny motoru naklonu (osa y)
int PinY1 = 8;
int PinY2 = 9;
int PinY3 = 10;
int trimr_u = A6;    //trimr tolerance uhlu
int trimr_f = A7;    //trimr tolerance fotorezistoru

const byte zaznamu = 200;          //zaznamy na karte pro jeden den
float necitlivost_f = 0.08 * 256; //tolerance fotorezistoru
float necitlivost_u = 5;           //tolerance uhlu

//nastaveni pinu pro displej
LiquidCrystal lcd(39, 41, 43, 45, 47, 49);

//nastaveni motoru
Stepper motorX(200, PinX0, PinX2, PinX1, PinX3);
Stepper motorY(200, PinY0, PinY2, PinY1, PinY3);

//nastaveni snimacu polohy a Kalmanova filtru
HMC5883L hmc;
```

```
MPU6050 mpu;
KalmanFilter kalmanX(0.001, 0.003, 0.03);
KalmanFilter kalmanY(0.001, 0.003, 0.03);

float akc_naklon = 0;    //naklon dle akcelerometru
float azimut = 0;        //skutecny azimut
float zenit = 0;         //skutecny zenit
float azimut_n = 0;      //pozadovany azimut
float zenit_n = 0;       //pozadovany zenit
int xPozice = 0;         //vychylka joysticku v ose x
int yPozice = 0;         //vychylka joysticku v ose y
byte rezim = 1;    //rezim: 1 = automaticky, 2 = casovac, 3 = manualni
int hp = 0;              //hodnota fotorezistoru HP
int hl = 0;              //hodnota fotorezistoru HL
int dp = 0;              //hodnota fotorezistoru DP
int dl = 0;              //hodnota fotorezistoru DL
int nahore = 0;          //osvetleni horni casti
int dole = 0;            //osvetleni dolni casti
int vlevo = 0;           //osvetleni leve casti
int vpravo = 0;          //osvetleni prave casti
int rozdilX = 0;         //rozdil osvetleni v ose x
int rozdilY = 0;         //rozdil osvetleni v ose y
int i = 0;               //pomocna promenna
int j = -1;              //pomocna promenna

//promenne casu
byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
int cas_h, cas_m, den_m;
float pole[zaznamu][4]; //pole dat z SD karty
byte cislo = 0;          //pomocna promenna
byte radek = 0;          //pomocna promenna
char soubor[9] = "";     //nazev souboru na karte
File file;               //ukazatel souboru
size_t n;                //pomocna promenna
char str[20];            //pomocna promenna

//prace se soubory
size_t readField(File* file, char* str, size_t size, char* delim) {
  char ch;
  size_t n = 0;
  while ((n + 1) < size && file->read(&ch, 1) == 1) {
    if (ch == '\r') {
      continue;
    }
    str[n++] = ch;
    if (strchr(delim, ch)) {
        break;
```

```
    }
  }
  str[n] = '\0';
  return n;
}

//prevod mezi soustavami
byte decToBcd(byte val){ return( (val/10*16) + (val%10) ); }
byte bcdToDec(byte val){ return( (val/16*10) + (val%16) ); }

//nastaveni casu
void setDS3231time(byte second, byte minute, byte hour, byte dayOfWeek,
     byte dayOfMonth, byte month, byte year){
    Wire.beginTransmission(DS3231_I2C_ADDRESS); //navazani spojeni
    Wire.write(0);              //nastaveni ukazatele na registr sekund
    Wire.write(decToBcd(second));                //nastaveni sekund
    Wire.write(decToBcd(minute));                //nastaveni minut
    Wire.write(decToBcd(hour));                  //nastaveni hodin
    Wire.write(decToBcd(dayOfWeek));             //nastaveni dne v tydnu
    Wire.write(decToBcd(dayOfMonth));            //nastaveni data
    Wire.write(decToBcd(month));                 //nastaveni mesice
    Wire.write(decToBcd(year));                  //nastaveni roku
    Wire.endTransmission();                      //konec spojeni
}

//ziskani casu
void readDS3231time(byte *second, byte *minute, byte *hour, byte *
    dayOfWeek, byte *dayOfMonth, byte *month, byte *year){
    Wire.beginTransmission(DS3231_I2C_ADDRESS); //navazani spojeni
    Wire.write(0);              //nastaveni ukazatele na registr sekund
    Wire.endTransmission();                    //konec spojeni
    Wire.requestFrom(DS3231_I2C_ADDRESS, 7); //pozadavek na 7 bajtu dat
    *second = bcdToDec(Wire.read() & 0x7f);
    *minute = bcdToDec(Wire.read());
    *hour = bcdToDec(Wire.read() & 0x3f);
    *dayOfWeek = bcdToDec(Wire.read());
    *dayOfMonth = bcdToDec(Wire.read());
    *month = bcdToDec(Wire.read());
    *year = bcdToDec(Wire.read());
}

void setup() {
  Wire.begin();             //komunikace po I2C
  Serial.begin(115200);     //komunikace po seriove lince

  //nehraje roli, ale musi byt zavolano
  motorX.setSpeed(30);
```

```
137   motorY.setSpeed(30);

139   //nastaveni rezimu pinu
      pinMode(xPin, INPUT);
141   pinMode(yPin, INPUT);
      pinMode(pinCas, INPUT_PULLUP);
143   pinMode(pinMan, INPUT_PULLUP);
      pinMode(hpPin, INPUT);
145   pinMode(hlPin, INPUT);
      pinMode(dpPin, INPUT);
147   pinMode(dlPin, INPUT);
      pinMode(trimr_u, INPUT);
149   pinMode(trimr_f, INPUT);

151   //nastaveni pracovniho rezimu
      if (digitalRead(pinCas) == LOW){
153     rezim = 2;
        Serial.println("Rezim casovace");      //vypis na seriovou linku
155   }else if (digitalRead(pinMan) == LOW){
        rezim = 3;
157     Serial.println("Manualni rezim");
      }else{
159     rezim = 1;
        Serial.println("Automaticky rezim");
161   }

163   //nastaveni casu (seconds, minutes, hours, day, date, month, year)
      //setDS3231time(0,37,16,4,26,04,17);
165
      Serial.println("Nastaveni displeje");
167   lcd.begin(16, 2);

169   Serial.println("Spousteni kompasu");
      while (!hmc.begin()){
171     Serial.println("Kompas nenalezen!");
        lcd.clear();                //vyprazdneni displeje
173     lcd.setCursor(0,0);         //nastaveni kurzoru
        lcd.print("Kompas");        //vypis na displej
175     lcd.setCursor(0,1);         //presun kurzoru
        lcd.print("nenalezen!");
177     delay(2000);                //dvousekundova pauza
      }
179   Serial.println("Spousteni akcelerometru");
      while(!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G)){
181     Serial.println("Akcelerometr nenalezen!");
        lcd.clear();
183     lcd.setCursor(0,0);
```

```
    lcd.print("Akcelerometr");
185   lcd.setCursor(0,1);
    lcd.print("nenalezen!");
187   delay(2000);
  }
189
  hmc.setRange(HMC5883L_RANGE_1_3GA);           //nastaveni rozsahu
191 hmc.setMeasurementMode(HMC5883L_CONTINOUS); //nastaveni rezimu
  hmc.setDataRate(HMC5883L_DATARATE_15HZ);     //frekvence mereni
193 hmc.setSamples(HMC5883L_SAMPLES_4);        //pocet vzorku pro vypocet

195 //proces kalibrace
  if (digitalRead(pinKal) == LOW){
197   hmc.setOffset(0, 0);     //magnetometr
    mpu.calibrateGyro();     //akcelerometr
199 }

201 if (rezim == 2){
    Serial.println("Pripojovani karty");
203   while (!SD.begin(CS_PIN)){
      Serial.println("Kartu se nepodarilo pripojit.");
205     lcd.clear();
      lcd.setCursor(0,0);
207     lcd.print("Kartu nelze");
      lcd.setCursor(0,1);
209     lcd.print("pripojit!");
      delay(2000);
211   }
    readDS3231time(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &
   month, &year);                    //ziskani data a casu
213   sprintf(soubor, "%02d-%02d.txt", month, dayOfMonth);
    file = SD.open(soubor);           //otevreni souboru pro dany den
215   Serial.println("Otevirani souboru dat");
    while (!file){
217     Serial.println("Soubor nelze otevrit.");
      lcd.clear();
219     lcd.setCursor(0,0);
      lcd.print("Soubor nelze");
221     lcd.setCursor(0,1);
      lcd.print("otevrit!");
223     delay(2000);
    }
225   file.seek(0);               //nastaveni bajtu v souboru
    while (true) {              //cteni carkami oddelenych hodnot
227     n = readField(&file, str, sizeof(str), ",\n");
      if (n == 0) break;                    //konec souboru
229     if (str[n-1] == ',' || str[n-1] == '\n') {
```

```
      //i = sloupcovy index, j = radkovy index
231         if (i == 0) j++;                    //novy radek
            if (i == 3) i = -1;
233         i = i + 1;
          str[n-1] = 0;                         //odstraneni carky
235     } else{
          if (file.available()){
237         Serial.print("Chyba!");
          }else{                                //konec souboru
239         i = 0;
          }
241     }
        float cislo = atof(str);                //prevod na cislo
243     pole[j][i == 0 ? 3 : i - 1] = cislo;    //zapis do pole hodnot
    }
245 file.close();                                //uzavreni souboru
    cas_h = 100 * hour + minute;                //aktualni cas
247 cas_m = minute;
    den_m = dayOfMonth;
249 for (i = 0; i < zaznamu; i++){              //prochazeni pole zaznamu
      if (100 * pole[i][0] + pole[i][1] <= cas_h){
251     radek = i;                  //vyhledani radku dle aktualniho casu
      }else{
253     break;
      }
255 }
    azimut_n = pole[radek][3];      //nastaveni pozadovaneho azimutu
257 zenit_n = pole[radek][2];       //nastaveni pozadovaneho zenitu
  }
259 i = 0;                          //reset pomocne promenne
  if (rezim == 3){          //manualni rezim
261 lcd.clear();
    lcd.setCursor(4,0);
263 lcd.print("MANUALNI");
    lcd.setCursor(5,1);
265 lcd.print("REZIM!");
  }
267 }

269 void loop() {
  //cteni normalizovanych hodnot z akcelerometru
271 Vector norm = hmc.readNormalize();
  Vector acc = mpu.readNormalizeAccel();
273 Vector gyr = mpu.readNormalizeGyro();

275 //cteni hodnot z joysticku
  xPozice = analogRead(xPin);
```

```
277   yPozice = 1023 - analogRead(yPin);

279   //vypocet uhlu (rad)
      azimut = atan2(norm.YAxis, norm.XAxis);
281   akc_naklon = -(atan2(acc.XAxis, sqrt(acc.YAxis*acc.YAxis + acc.ZAxis*
        acc.ZAxis))*180.0)/PI;
      zenit = -kalmanY.update(akc_naklon, gyr.YAxis);
283
      //deklinace: (deg + (min / 60)) / (180 / PI);
285   float deklinace = (4 + (32 / 60)) / (180 / PI);
      azimut += deklinace;
287
      //korekce uhlu
289   if (azimut < 0){azimut += 2 * PI;}
      if (azimut > 2 * PI){azimut -= 2 * PI;}
291   azimut = azimut * 180/PI;

293   //ziskani hodnot z fotorezistoru a vypocet prumernych hodnot
      hp = analogRead(hpPin);
295   hl = analogRead(hlPin);
      dl = analogRead(dlPin);
297   dp = analogRead(dpPin);
      nahore = (hp + hl) / 2;
299   dole = (dp + dl) / 2;
      vlevo = (dl + hl) / 2;
301   vpravo = (hp + dp) / 2;
      rozdilX = vlevo - vpravo;
303   rozdilY = nahore - dole;

305   //nacteni hodnot trimru
      necitlivost_f = map(analogRead(trimr_f), 0, 1023, 1, 256);
307   necitlivost_u = map(analogRead(trimr_u), 0, 1023, 1, 360);

309   switch(rezim){
        case 1: //automaticky
311       if (i >= 300){      //aktualizace displeje kazdou tristou iteraci
            lcd.clear();
313         lcd.setCursor(0,0);
            lcd.print(hl);        //udaje z hornich fotorezistoru
315         lcd.print("/");
            lcd.print(hp);
317         lcd.setCursor(9,0);
            lcd.print("(~");
319         lcd.print(necitlivost_f, 0);  //aktualni nastaveni tolerance
            lcd.print(")");
321         lcd.setCursor(0,1);
            lcd.print(dl);        //udaje ze spodnich fotorezistoru
```

```
323         lcd.print("/");
            lcd.print(dp);
325         lcd.setCursor(9,1);
            lcd.print("(~");
327         lcd.print(necitlivost_f, 0);  //opet tolerance
            lcd.print(")");
329         i = 0;                        //restart pocitadla iteraci
          }
331       if (rozdilX > necitlivost_f){   //kontrola proti toleranci
            motorX.step(-1);              //otoceni motoru
333         delay(1);
          }else if (rozdilX < -necitlivost_f){
335         motorX.step(1);               //opacny smer
            delay(1);
337       }
          if (rozdilY > necitlivost_f){
339         motorY.step(-1);
            delay(1);
341       }else if (rozdilY < -necitlivost_f){
            motorY.step(1);
343         delay(1);
          }
345     break;
        case 2: //casovac
347       readDS3231time(&second, &minute, &hour, &dayOfWeek, &dayOfMonth,
    &month, &year);       //aktualni cas
          if (den_m != dayOfMonth){       //po prechodu na novy den
            asm volatile ("  jmp 0");     //restart Arduina
          }
351       if (minute != cas_m){           //kontrola kazdou minutu
            cas_m = minute;
353         cas_h = 100 * hour + minute;
            if (pole[radek + 1][0] != 0){   //kontrola pole dat
355           if (100 * pole[radek + 1][0] + pole[radek + 1][1] <= cas_h){
                radek++;
357             azimut_n = pole[radek][3];  //nastaveni pozadovanych hodnot
                zenit_n = pole[radek][2];
359           }
            }
361       }
          if (i >= 300){                  //kazda trista iterace
363         lcd.clear();
            lcd.setCursor(0,0);
365         lcd.print(azimut, 1);         //vypis skutecneho azimutu
            lcd.print("/");
367         lcd.print(zenit, 1);          //vypis skutecneho zenitu
            lcd.setCursor(12,0);
```

```
369           lcd.print("~");
              lcd.print(necitlivost_u, 0);     //tolerance uhlu
371           lcd.setCursor(0,1);
              lcd.print("~");
373           lcd.print(azimut_n);             //pozadovany azimut
              lcd.print("/");
375           lcd.print(zenit_n);              //pozadovany zenit
              i = 0;
377         }
            if (azimut_n > azimut + necitlivost_u){
379           motorX.step(-1);
              delay(1);
381         }else if (azimut_n < azimut - necitlivost_u){
              motorX.step(1);
383           delay(1);
            }
385         if (zenit_n > zenit + necitlivost_u){
              motorY.step(1);
387           delay(1);
            }else if (zenit_n < zenit - necitlivost_u){
389           motorY.step(-1);
              delay(1);
391         }
        break;
393     case 3: //manualni
            if (xPozice < 300){          //kontrola vychylky joysticku
395           motorX.step(-1);
              delay(1);
397         }else if (xPozice > 700){
              motorX.step(1);
399           delay(1);
            }
401         if (yPozice < 300){
              motorY.step(1);
403           delay(1);
            }else if (yPozice > 700){
405           motorY.step(-1);
              delay(1);
407         }
        break;
409   }
      i++;                               //pripocitani iterace
411 }
```

D ŘÍZENÍ SLEDOVAČE POMOCÍ ZPRACOVÁNÍ OBRAZU

Další možností, jak řídit pohyb solárního zařízení, je pomocí snímání a následného zpracování obrazu. Lze snímat buďto oblohu a vyhledávat pozici slunce, nebo výměník tepla (u solárních koncentrátorů) a vyhodnocovat polohu vzniklého ohniska. Jestliže se slunce (nebo ohnisko) nachází mimo střed (resp. stanovenou oblast) obrazu, je třeba polohu vyhodnotit a zařízením otočit příslušným směrem.

Zpracování obrazu v potřebném rozsahu není možné provést pomocí Arduina, proto nebylo do návrhu zahrnuto. Níže uvedený kód představuje pouze strohý nástin řešení daného problému v prostředí MATLAB pomocí dvou různých přístupů a předpokládá jako vstupní soubor snímek oblohy. Vyhledání nejjasnějšího místa je realizováno pomocí multiregionální segmentace a prahování (snímek je rozdělen na sedm oblastí, oblasti odpovídající nejjasnějším místům jsou poté převedeny na absolutně bílou, zbylé oblasti na absolutně černou barvu) a pomocí převodu snímku na barevný model HSV (nejjasnější místo je určeno přímo na základě údaje o jasu). Skript poté vypočítá a do obrázku vyznačí těžiště nejjasnější oblasti a zároveň vyznačí i střed snímku. Rozdíl těchto dvou bodů určuje potřebné natočení solárního zařízení.

```
clc;
clear all;
close all;

ii = imread('foto.jpg');
%ii = imread('foto.png');

% extrakce R G B slozek
rr = ii(:,:,1);
g = ii(:,:,2);
b = ii(:,:,3);

% multiregionalni segmentace
% definice prahu - metoda Otsu
% 6 prahu, 7 oblasti
thresh = multithresh(b,6);
seg = imquantize(b,thresh);
imagesc(seg)

[r, s] = size(seg);
for i1 = 1:r
    for j1 = 1:s
        if seg(i1,j1) == 7 % posledni oblast
            seg2(i1,j1) = 1; % prirazeni absolutni bile
```

```
25          else
                seg2(i1,j1) = 0; % absolutni cerna
27          end
        end
29  end

31  figure()

33  vyska = size(ii,1);
    sirka = size(ii,2);
35
    stred_y = floor(vyska/2);
37  stred_x = floor(sirka/2);

39  % konverze na HSV
    hsv = rgb2hsv(ii);
41  % jasova slozka
    v = hsv(:,:,3);
43  % nejvyssi jas
    max_v = max(max(v));
45  % souradnice pixelu s nejvyssim jasem
    [r, s] = find(v == max_v);
47
    % vypocet teziste
49  tez_y = mean(r);
    tez_x = mean(s);
51
    seg2 = insertMarker(seg2,[stred_x stred_y]);
53  seg2 = insertMarker(seg2,[tez_x tez_y],'x');
    imshow(seg2)
55
    figure()
57
    ii = insertMarker(ii,[stred_x stred_y]);
59  ii = insertMarker(ii,[tez_x tez_y],'x');
    imshow(ii)
61
    rozdil_x = stred_x - tez_x
63  rozdil_y = stred_y - tez_y
```

E OBSAH PŘILOŽENÉHO CD

- `/` ..... kořenový adresář
  - `arduino` ..... zdrojové kódy a knihovny pro Arduino
    - `lib` ..... knihovny pro import
      - `Adafruit_HMC5883_U.zip` HMC5883U
      - `Adafruit_Sensor.zip` ..... Sensor
      - `Arduino-KalmanFilter-master.zip` ..... KalmanFilter
      - `HMC5883L.zip` HMC5883L
      - `LiquidCrystal.zip` ..... LiquidCrystal
      - `MPU6050.zip` MPU6050
      - `SD.zip` ..... SD
      - `Servo.zip` ..... Servo
      - `Stepper.zip` ..... Stepper
    - `model1.ino` ..... zdrojový kód pro model s krokovými motory
    - `model2.ino` ..... zdrojový kód pro model se servomotory
    - `tracker.ino` ..... základní zdrojový kód trackeru
  - `data` ..... vypočtená data o poloze slunce, časy východu a západu
    - `2017.zip` ..... pro rok 2017
    - `2018.zip` ..... pro rok 2018
    - `2019.zip` ..... pro rok 2019
    - `2020.zip` ..... pro rok 2020
    - `2021.zip` ..... pro rok 2021
    - `2022.zip` ..... pro rok 2022
  - `matlab` ..... zpracování snímku slunce na obloze
    - `foto.jpg` ..... vstupní fotografie
    - `foto.png` ..... vstupní fotografie
    - `slunce.m` ..... algoritmus pro MATLAB
  - `deska.eps` ..... návrh desky
  - `deska_horni.eps` ..... matrice horní strany desky
  - `deska_spodni.eps` ..... matrice spodní strany desky
  - `obsah.txt` ..... tento obsah
  - `stinitko.stl` ..... 3D model čtyřkvadrantového stínítka