

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKEJ A POTRAVINÁRSKEJ
TECHNOLÓGIE**

EVIDENČNÉ ČÍSLO: FCHPT-5415-61556

**OPTIMÁLNE RIADENIE FRESNELOVEJ ŠOŠOVKY
BAKALÁRSKA PRÁCA**

2015

Peter Bakaráč

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKEJ A POTRAVINÁRSKEJ
TECHNOLÓGIE**

OPTIMÁLNE RIADENIE FRESNELOVEJ ŠOŠOVKY

BAKALÁRSKA PRÁCA

FCHPT-5415-61556

Študijný program: automatizácia, informatizácia a manažment v chémii a potravinárstve

Číslo študijného odboru: 2621

Názov študijného odboru: 5.2.14. automatizácia, 5.2.52. priemyselné inžinierstvo

Školiace pracovisko: Ústav informatizácie, automatizácie a matematiky (FCHPT)

Vedúci záverečnej práce: doc. Ing. Michal Kvasnica, PhD.

2015

Peter Bakaráč



ZADANIE BAKALÁRSKEJ PRÁCE

Študent: **Peter Bakaráč**
ID študenta: 61556
Študijný program: automatizácia, informatizácia a manažment v chémii a potravinárstve
Kombinácia študijných odborov: 5.2.14. automatizácia, 5.2.52. priemyselné inžinierstvo
Vedúci práce: doc. Ing. Michal Kvasnica, PhD.

Názov práce: **Optimálne riadenie Fresnelovej šošovky**

Špecifikácia zadania:

Cieľom práce je zostrojiť mechanický systém schopný zohrievať vodu v zásobníku kvapaliny pomocou slnečnej energie, sústredenej na vhodný teplo-výmenný objekt, pomocou Fresnelovej šošovky.

Úlohy:

- * tvorba mechanického systému
- * zostrojenie zariadenia na automatické sledovanie pozície s maximálnou slnečnou aktivitou
- * vytvorenie platformy na automatický zber a vyhodnocovanie dát
- * tvorba riadiaceho algoritmu sledujúceho pozíciu s maximálnou slnečnou intenzitou

Rozsah práce: 40

Riešenie zadania práce od: 16. 02. 2015

Dátum odovzdania práce: 24. 05. 2015

L. S.

Peter Bakaráč
študent

prof. Ing. Miroslav Fikar, DrSc.
vedúci pracoviska

prof. Ing. Miroslav Fikar, DrSc.
garant študijného programu

Pod'akovanie

Týmto by som sa chcel poďakovať môjmu školiteľovi Michalovi Kvasnicovi za jeho proaktívny prístup k projektom. Za jeho záujem o nové nápady a vloženie dôvery, ktorá ma motivovala k práci. Za finančnú ale aj odbornú pomoc, ktorá mala nepochybny podiel na celom projekte. Tejto podpory si veľmi vážim. Ďakujem

Abstrakt

Projekt Optimálne riadenie Fresnelovej šošovky sa zaoberá konštrukciou mechanického systému, ktorý bude principiálne plniť úlohu solárneho koncentrátora a pomocou čerpadla a výmenníka tepla nachádzajúceho sa v ohnisku slnečných lúčov koncentrátora, bude schopný zohrievať teplotné médium – vodu. Úlohami projektu sú konštrukcia mechanického, pohyblivého systému, ktorý uchytí a natočí Fresnelovu šošovku. Pohyb bude riadený elektronickou platformou Arduino, ktorá je programovateľná. Navrhnutý riadiaci algoritmus vložený do procesora dosky bude mať za úlohu vyhodnotiť signály z vhodne umiestnených sústav fotosenzorov a zabezpečiť pohyb otáčavých častí systému tak, aby natočil Fresnelovu šošovku kolmo na smer šírenia slnečného žiarenia. Výmenníkom tepla umiestneným v ohnisku slnečných lúčov bude čerpadlom hnaná voda, ktorá sa vo výmenníku bude zohrievať a následne skladovať v zásobníku kvapaliny so zabudovaným teplotným senzorom. Navrhnutá platforma automatického zberu dát bude odosielať relevantné údaje cez internet, na hostingovú službu umožňujúcu skladovanie dát v ich databázach. Následne bude k týmto dátam umožnený prístup. Tieto ciele sa podarilo splniť a systém je pripravený na zber dát a ich vyhodnocovanie. Význam tohto projektu spočíva v jeho perspektíve využiteľnosti v oblasti obnoviteľných zdrojov energie.

Kľúčové slová: Fresnelova šošovka; fotosenzor; Arduino

Abstract

The Project “Optimal control of Fresnel lens“ deals with the problem of constructing a mechanical system, which would basically serve as a solar concentrator. With the use of a pump and a heat exchanger located in the concentrator’s focal point, it will be able to warm up the heat carrier – water.

The main objective of the project is the construction of a mobile mechanical system, which would hold and rotate the Fresnel lens. This movement will be controlled by an Arduino electronic platform, which is well programmable. The designed control algorithm inserted into the microcontroller will serve to evaluate the signals of conveniently placed sets of photo sensors, and secure movement of the system’s rotary parts, so that the Fresnel lens would be always faced perpendicularly towards the direction of sun rays.

With the use of a pump, water will flow through the heat exchanger located in the lens’ focal point. This will warm up the water, which is then going to be stored inside a tank with a built-in heat sensor. Our designed platform of automatic data collection will send all relevant data through the internet, to a hosting service that allows storage of data inside their databases. All the information will then become accessible.

We’ve succeeded in meeting these objectives, and the system is ready for collecting data and its evaluation. The results and methods used in this project can be significant for further research of renewable energy sources.

Key words: Fresnel lens; photo sensor; Arduino.

Obsah

Zoznam skratiek a značiek	14
Úvod.....	16
1 Energia zo Slnka	18
1.1 Solárne panely.....	18
1.2 Solárne koncentrátoory	18
1.3 Fresnelova šošovka	19
2 Cieľ práce	20
3 Konštrukcia zariadenia	22
3.1 Upevnenie a otáčanie Fresnelovej šošovky.....	22
3.1.1 Rám	22
3.1.2 Držiak rámu.....	24
3.1.3 Podstavec	24
3.2 Systém výmeny tepla	27
3.2.1 Výmenník tepla	27
3.2.2 Zásobníky kvapaliny	28
3.2.3 Cirkulácia	28
4 Riadenie	30
4.1 Arduino	30
4.2 Elektronika.....	31
4.3 Ovládanie pohybu	31
4.3.1 Tranzistor	31
4.3.2 H-mostík	32
4.3.3 Odkúšanie	41
4.4 Sledovanie Slnka.....	41
4.4.1 Fotosenzory	43

4.5	Riadenie pohybu	45
4.5.1	Hľadanie Slnka na osi X	45
4.5.2	Hľadanie Slnka na osi Y	46
4.5.3	Algoritmus	46
4.6	Spúšťanie čerpadla.....	50
4.6.1	Teplotný senzor.....	50
4.6.2	Zapnutie čerpadla.....	51
4.7	Optimálne riadenie.....	53
5	Zber a uchovávanie dát.....	54
5.1	Odosielanie dát na server	55
	Zoznam použitej literatúry	60
	Prílohy.....	61

Zoznam skratiek a značiek

DPS	doska plošných spojov
OptiFres	Optimálne riadenie Fresnelovej šošovky
GND	(z angl. Ground) – uzemnenie, v elektronike znamená referenčný potenciál v obvode
PWM	(z angl. Pulse Width Modulation) impulzovo šírková modulácia

Úvod

Každým rokom rastie dopyt ľuďstva po elektrickej energii. Vznikajú nové technológie, pribúdajú nové zariadenia a v neposlednom rade, zvyšuje sa aj samotný počet ľudí na Zemi. To vyvoláva otázku, čím bude tento dopyt po energii uspokojený? Projekt Optimálne riadenie Fresnelovej šošovky, je perspektívnou možnosťou stať sa jedným z takýchto zdrojov energie. Principiálne by malo zariadenie fungovať ako solárny koncentrátor. To znamená, koncentrovať slnečné lúče do ohniskového bodu v ktorom sa bude nachádzať výmenník tepla. Cez tento výmenník bude prúdiť teplotnosné médium s účelom toto teplo odvádzať a skladovať v zásobníku kvapaliny, v podobe teplej vody. Energia zo Slnka je zdarma, takže tento spôsob zohrievania vody je alternatívou k používaniu iných ohrevov vody, za predpokladu priaznivého počasia.

Cieľom projektu je zostrojiť také zariadenie, ktoré pozostáva z mechanickej a riadiacej časti. Mechanická časť zariadenia bude mať za úlohu uchytiť Fresnelovu šošovku a dvojsovo s ňou otáčať okolo vertikálnej a horizontálnej osi. Je to treba na to, aby bolo možné túto šošovku natočiť kolmo na smer šírenia elektromagnetického žiarenia emitovaného Slnkom. Konštrukcia musí byť pevná a stabilná, aby odolala vetru, ale zároveň, aby bola schopná otáčať pohyblivými časťami. Nebude robená podľa žiadnej predlohy, takže ju bude potrebné navrhnuť a odskúšať. Zostrojenie takejto konštrukcie si vyžaduje zručnosti v obrábaní materiálu z ktorého má byť skonštruovaná a znalosť aspoň základných mechanických komponentov ako sú ložiská, motorčeky, ozubené kolesá a pod.

Rozhodnutie použiť Fresnelovu šošovku vyplývalo z jej výhod oproti klasickej, vypuklej šošovke. Z Fresnelovej šošovky sú odstránené tie časti, ktoré sa priamo nepodieľajú na lome svetla, takže tým, že je plochá zaberá oveľa menší objem a jej hmotnosť je taktiež výrazne redukovaná. Tieto vlastnosti ju robia menej náročnú na manipuláciu.

Mechanický systém je treba riadiť. Akčným členom, ktorý zabezpečí pohyb konštrukcie sú motorčeky na jednosmerné napätie 12 voltov. Tieto motorčeky budú dva a každý z nich bude otáčať pohyblivými časťami okolo inej osi. Pohyb bude riadený elektronickou platformou Arduino. Táto elektronická doska disponuje analógovými a digitálnymi vstupmi a výstupmi, čo umožňuje jej interakciu s okolím. Je programovateľná, takže táto interakcia závisí od programátora. Signály do vstupov a výstupov z Arduina vyžadujú istý, predpísaný formát elektrického napätia, takže k projektu je nutné navrhovať potrebné elektronické obvody, ktoré zabezpečia tento signálový formát zo senzorov, prípadne jeho interpretáciu pri napájaní motorčekov.

Na to, aby mohol systém automaticky natáčať Fresnelovu šošovku za Slnkom, musí poznať jeho polohu na oblohe. K tomuto účelu musia byť navrhnuté a zostrojené sústavy fotosenzorov tak, aby boli schopné navigovať pohyb otáčavých častí potrebným smerom. Tento pohyb bude riadený riadiacim algoritmom, ktorý bude napísaný v programovacom jazyku C++ a vložený do Arduina.

Po splnení úlohy automatického sledovania Slnka budú rozmiestnené teplotné senzory do zásobníkov kvapaliny a jeden mimo nich. Dáta z teplotných senzorov budú odosielané prostredníctvom Ethernetového modulu na internet, kde budú ukladané do databáz hostingovej služby a následne k nim bude voľný prístup za účelom vyhodnotenia prevádzky systému.

1 Energia zo Slnka

Slnko je základným zdrojom energie pre planétu Zem už od jej vzniku. Slnčná energia je kľúčovým elementom takmer všetkého života na Zemi a posledné storočie sa človek snaží túto energiu využiť aj inak ako prirodzeným vstrebávaním telom. Každú sekundu vyžiari Slnko toľko energie, že by to stačilo pokryť potreby celého sveta na viac než 1000 rokov [1]. Toto množstvo vyžiarenej energie robí Slnko perspektívnym zdrojom obnoviteľnej energie. Snahy o zvýšenie efektivity premeny slnečnej energie na jej využiteľnú formu sa stále zvyšujú a počet zariadení na to určených stúpa. Napriek tomu však zostáva ešte veľký priestor k zlepšovaniu a zefektívňovaniu solárnych technológií.

1.1 Solárne panely

Jedným z najrozšírejších zariadení na získavanie energie zo Slnka sú solárne panely. Solárne panely sú tvorené solárnymi článkami, ktoré môžu byť tvorené polovodičovými alebo organickými prvkami. Ich účel je premieňať elektromagnetické žiarenie na elektrickú energiu. Existuje viacero druhov solárnych článkov. V závislosti od použitých materiálov pri výrobe a samotných technologických postupov sa odvíja aj ich účinnosť a cena. Najkvalitnejšie solárne články sa vyrábajú z monokryštalického kremíku a používajú sa tam, kde je žiaduce, aby mali panely čo najmenšie rozmery, napríklad v kozmických aplikáciách. Ich účinnosť dosahuje až 34%. Solárne články energeticky menej náročné na výrobu dosahujú účinnosť okolo 17% a sú používané aj u nás [2]. Na Slovensku sa stali bežne využívaným, doplnkovým zdrojom elektrickej energie v domácnostiach ale aj vo firmách. Avšak cena solárnych panelov je stále dosť vysoká na to, aby využívanie konvenčných zdrojov elektrickej energie bolo ekonomicky výhodnejšie.

1.2 Solárne koncentrátory

Slnčné koncentrátory vyrábajú teplo využitím sústavy reflektorov, šošoviek alebo zrkadiel, ktoré koncentrujú slnečné žiarenie do ohniska, v ktorom sa nachádza teplonosné médium. Typický koncentračný systém pozostáva z koncentrátora, teplonosného média, ohniskovej jednotky, potrubí, generátorov elektrického prúdu a skladovacieho systému. Tieto systémy využívajú parabolické zrkadlá, ktoré koncentrujú slnečné žiarenie na potrubie umiestnené do ohniska zariadenia. V potrubí prúdi kvapalina, ktorá sa ohrieva na takmer 400 stupňov Celzia a je prečerpávaná cez sústavu tepelných výmenníkov tak, že na konci vzniká para s veľmi vysokou

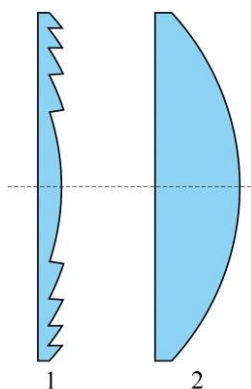
teplotou, ktorá poháňa turbínu generátora vyrábajúcu elektrinu. Najväčší takýto solárny systém na svete postavila firma Luz International začiatkom 80-tych rokov v púšti Mojave [3].

1.3 Fresnelova šošovka

Optická šošovka je predmet z priehľadného materiálu slúžiaci v optike alebo iných prípadoch na ovplyvnenie šírenia svetla – elektromagnetického žiarenia. Prechodom žiarenia cez šošovku sa zmení jeho smer šírenia v závislosti od tvaru šošovky a materiálu, z ktorého je vyrobená. Ohniskovým bodom sa nazýva priesečník trajektórií jednotlivých lúčov žiarenia prechádzajúceho šošovkou.

Fresnelovu šošovku vynášiel v roku 1822 francúzsky fyzik Augustin Jean Fresnel a je svojimi vlastnosťami podobná klasickej optickej šošovke s tým rozdielom, že sú z nej odstránené tie časti, ktoré sa priamo nepodieľajú na lome svetla. To ju robí oveľa ľahšou oproti bežnej vypuklej šošovke. Pôvodne bola vyvinutá pre námorné majáky. Vďaka nej sa svetlo z majákov dostalo do vzdialenosti niekoľko desiatok kilometrov. Dnes má Fresnelova šošovka široké využitie v zobrazovacej technike, kde sa používa ako základ 3D displejov, v signalizačných svetidlách návestí a semaforov, v svetlometoch, policajných majákoch, infračervených snímačoch pohybu či solárnych koncentrátoroch [4]. Napríklad japonská firma Toshiba ešte pred tým, než sa začali vyrábať obrazovky s LED technológiou, používala veľké fresnelove šošovky v televíznych obrazovkách na zväčšenie obrazu a tým pádom zväčšenia ich uhlopriečky.

Fresnelova šošovka s rozmermi 1m x 1m, ktorá je natočená kolmo na slnko je schopná vo svojom ohnisku lúčov vytvoriť teplotu až 1000° C [5].



Obrázok 1.1 Rozdiel medzi Fresnelovou šošovkou (1) a bežnou šošovkou (2)

2 Cieľ práce

Cieľom práce je zostrojiť zariadenie, ktoré bude principiálne fungovať ako solárny koncentrátor. To znamená zariadenie, ktoré bude schopné sledovať pohyb slnka na oblohe a premeniť slnečnú energiu na jej využiteľnú formu v podobe zohrievania teplotnosného média, koncentrovaním ohniska slnečných lúčov na výmenník tepla.

Ako koncentrátor slnečného žiarenia bude slúžiť Fresnelova šošovka s rozmermi 310 x 310 mm s ohniskovým bodom kolmo vzdialeným od šošovky 330 mm.

Na dosiahnutie tohto cieľa je treba vytvoriť mechanický systém, ktorý bude schopný uchytiť Fresnelovú šošovku do rámu a s týmto rámom dvojsovo otáčať tak, aby bolo možné jeho kolmé natočenie k slnku. Táto úloha vyžaduje aspoň minimálnu úroveň zručnosti v obrábaní materiálov, konkrétne dreva, t. j. rezanie, vŕtanie, úprava povrchu a pod. Zároveň sú potrebné aj znalosti komponentov mechanických systémov, ako sú ložiská, či spojovací materiál.

Pohyb rámu bude zabezpečený elektrickými motorčekomí na jednosmerné napätie a ich pohyb bude riadený pomocou nízko nákladovej elektronickej platformy Arduino. Arduino však nie je schopné priamo ovládať pohyb motorčekov, čiže je potrebný medzičlánok v podobe ďalších elektronických obvodov, ktoré bude treba navrhnuť tak, aby boli schopné interpretovať signály z Arduina a spúšťať motorčeky v požadovanom smere. Tieto elektronické obvody bude nutné aj zrealizovať, odskúšať ich funkčnosť a správne ich zapojiť v súčinnosti s Arduinom.

Dosku Arduino je možné programovať. Je to pomyselný mozog zariadenia, ktorý na základe signálov zo senzorov bude vydávať jednotlivé povely k akčným zásahom. Naprogramovať riadenie je jednou z primárnych úloh projektu a bude kladený dôraz na jeho efektivitu.

Vstupom do riadiaceho algoritmu budú hodnoty signálov zo senzorov. Na sledovanie Slnka je potrebné poznať jeho polohu. Túto polohu bude možné identifikovať pomocou vhodne umiestnených sústav fotosenzorov. Aby mohol riadiaci algoritmus prijímať signály zo senzorov, je potrebné navrhnuť elektronickú schému ich zapojenia a následne túto schému zrealizovať v podobe dosky plošných spojov.

Ďalším cieľom práce je umiestniť do ohniska žiarenia teplo-výmenný objekt, t. j. výmenník tepla, cez ktorý je možný prietok teplo-vodivého média, v našom prípade vody. Tento výmenník tepla musí byť dostatočne pevne pripevnený k rámu v ktorom bude uchytená fresnelova šošovka. Je to preto, aby ohnisko žiarenia zostávalo na povrchu výmenníka a nebolo potrebné zvlášť riadiť aj pohyb výmenníka za ohniskom.

Aby mohlo teplotnosné médiu tiecť cez výmenník tepla, musí byť zabezpečená cirkulácia. K tomuto projektu postačí skonštruovať zásobník kvapaliny a zabezpečiť cirkuláciu vody pomocou čerpadla. Zásobník vody bude prepojený s výmenníkom tepla plastovými hadičkami, kvôli ich poddajnosti. Aby bolo možné určiť energiu prijatú vodou vďaka cirkulácii cez výmenník tepla, je potrebné pridať ďalší – referenčný zásobník kvapaliny. Voda v tomto zásobníku nebude súčasťou cirkulácie, ale jej teplota bude slúžiť ako dôležitý údaj pri výpočte prijatej energie vodou v pracovnom zásobníku, t. j. zásobníku, ktorého voda bude ohrievaná systémom. K poskytnutiu informácií o teplote v zásobníkoch poslúžia teplotné senzory, ktoré v nich budú umiestnené.

Hodnoty teplôt v zásobníkoch kvapaliny, teplotu okolia a intenzity slnečného žiarenia budú, pomocou ďalšieho Ethernetového modulu, odosielané na internetové databázové úložisko.

Takýto solárny koncentrátor bude prenosný a jeho hmotnosť sa predpokladá od 5 do 10kg.

3 Konštrukcia zariadenia

Keďže experiment sa zaoberá aj vytvorením reálneho systému, je treba vyhotoviť konštrukciu, ktorá bude schopná splniť požiadavky, ktoré kladie zadanie projektu. Je treba vytvoriť také zariadenie, ktoré bude schopné uchytiť a natáčať Fresnelovu šošovku okolo horizontálnej aj vertikálnej osi. Ďalšou podmienkou je, aby sa výmenník tepla nachádzal vždy v ohniskovom bode šošovky a bol k nemu natočený tou stranou, ktorá je určená na prenos tepla medzi ohniskom slnečného žiarenia a cirkulujúcou vodou vo výmenníku. Konštrukcia musí byť stabilná, aby odolala vetru, prípadne, aby sa neprevrhla na niektorú stranu vplyvom vlastnej hmotnosti.

Ako konštrukčný materiál bolo zvolené drevo, pretože je dostupné, lacné, ľahko sa obrába a poskytuje dostatočnú mieru pevnosti, ktorú si konštrukcia vyžaduje. Konkrétne rozmery jednotlivých kusov konštrukcie nebudú uvádzané, len postup a metodika práce, pretože funkčnosť systému nevyžaduje presné rozmery jeho súčastí.

K systému je potrebné pripevniť aj dva zásobníky kvapaliny a zabezpečiť cirkuláciu vody tak, aby prechádzala výmenníkom tepla a vracala sa naspäť do zásobníka.

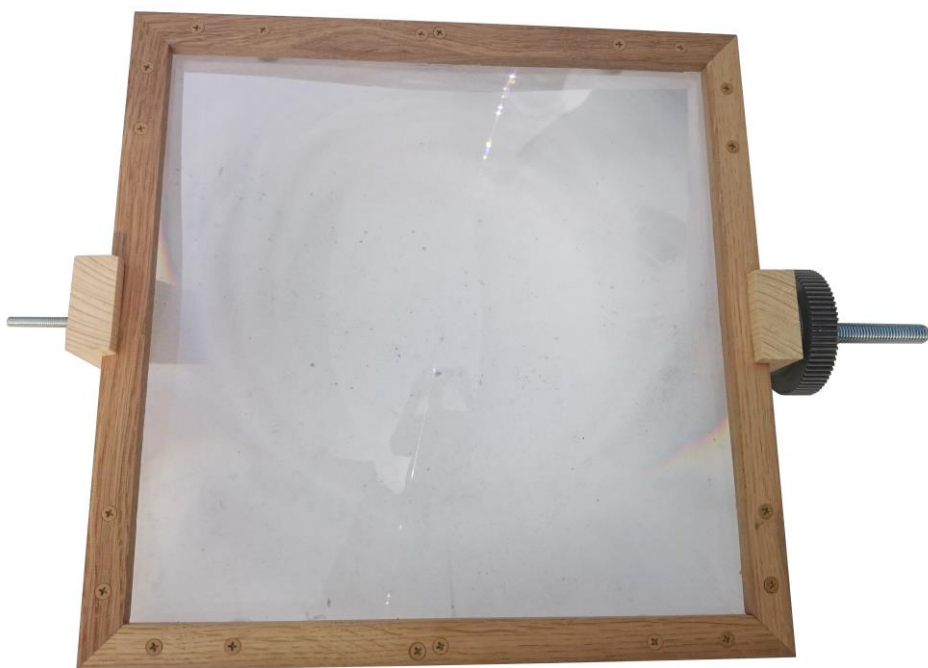
3.1 Upevnenie a otáčanie Fresnelovej šošovky

3.1.1 Rám

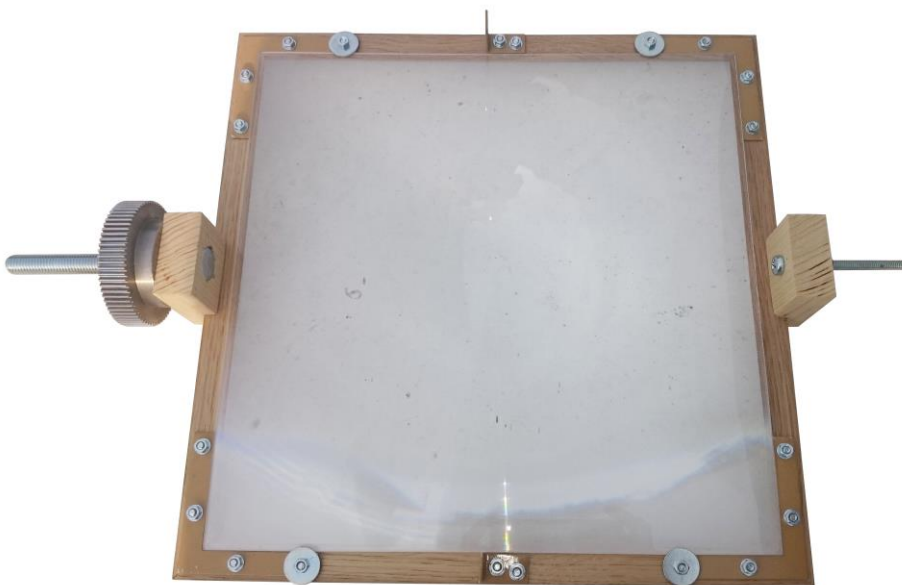
Na upevnenie Fresnelovej šošovky sme vytvorili rám z dreveného profilu. Profil sme narezali na 4 kusy, ktoré boli k sebe spojené lepidlom v pravom uhle, aby vytvorili štvorec. Tento štvorec má vnútorný rozmer menší ako šošovka a vonkajší rozmer väčší ako šošovka z toho dôvodu, aby sa dala k nemu priložiť a následne pripevniť. Tento rám bol následne spevnený kovovými, pravouhlými spojkami, ktoré sa k nemu uchytili skrutkami M4 cez predvŕtané diery v ráme. Okrem pravouhlých spojok tieto skrutky v dolnej a hornej časti rámu uchytili aj šošovku.

Na to, aby sa rám mohol otáčať, je potrebné, pripevniť hriadeľ pozdĺž vodorovnej osi rámu. Ako hriadeľ na ľavej strane bola zvolená skrutka M6 a na strane pravej, skrutka M12. Tieto skrutky sú upevnené do drevenej kocky, ktorá má v sebe drážku, do ktorej zapadá časť rámu, ktorá ho tým zafixuje a lepidlom sa tieto dva komponenty spoja natrvalo. Skrutka M12 bola zvolená preto, lebo vnútorný priemer otvoru ozubeného kolesa 1 je 12 mm. Toto ozubené koleso

je nasadené na pravý hriadeľ a pevne pripevnené k rámu. Ozubené kolesá sú špecifikované v tabuľke 2. Takto skonštruovaný rám je možné vidieť na obrázku 2.1 a 2.2.



Obrázok 2.1 Pohľad spredu na rám so šošovkou



Obrázok 2.2 Pohľad na rám so šošovkou zo zadnej strany

3.1.2 Držiak rámu

Na uchytenie rámu je potrebné skonštruovať držiak rámu. Tento držiak obsahuje po oboch stranách otvory v ktorých sú umiestnené guľové, jednoradové ložiská. Tieto ložiská majú vnútorné rozmery 6 (Obrázok 3.3) a 12 mm (Obrázok 3.4), aby sa do nich dali zasunúť hriadeľové skrutky rámu a umožnili tak plynulé otáčanie rámu okolo horizontálnej osi. Na pravú stranu držiaku rámu bol vyvŕtaný otvor, do ktorého bol umiestnený motorček. Na ňom je pripevnené ozubené koleso 2. Otvor je situovaný tak, aby ozubené koleso 1 a ozubené koleso 2, do seba zapadali (Obrázok 3.5). Držiak rámu má v spodnej časti, v strede otvor, v ktorom je skrutka M12 a tá uchyťáva ďalšie ozubené koleso typu 1. Táto skrutka okrem uchytenia ozubeného kolesa slúži aj ako hriadeľ, ktorý je vertikálnou osou držiaku rámu (Obrázok 3.6). Prehľad použitých typov ložísk je v tabuľke 1.

3.1.3 Podstavec

Tento držiak rámu je nasadený na podstavec. Podstavec je tvorený sústavou do seba zapadajúcich drevených profilov tak, aby vytvárali stabilnú základňu. V prednej časti, v strede je otvor a sústava ložísk, aby sa skrutka držiaku rámu mohla zasunúť do nich a zabezpečiť tak jeho

pohyb okolo vertikálnej osi. Pri tomto otvore na hriadeľ je umiestnený ďalší otvor v ktorom je upevnený motorček 2 tak, aby ozubené kolesá opäť do seba zapadali.



Obrázok 3.3 Ložisko 1 zapustené do ľavej strany držiaku rámu



Obrázok 3.4 Ložisko 2, pevne uchytené k pravej časti držiaku na rám skrutkami M5



Obrázok 3.5 Jednotlivé ozubené kolesá musia do seba zapadať



Obrázok 3.6 Podstavec

Tab 1. Ložiská

Ložisko	Vnútorný priemer [mm]	Vonkajší priemer [mm]	Šírka [mm]
1	6	19	6
2	12	37	12

Tab 2. Ozubené kolesá

Ozubené koleso	Modul	Počet zubov	Vnútorný priemer otvoru [mm]
1	1	70	12
2	1	16	6



Obrázok 3.7 Kompletná konštrukcia

3.2 Systém výmeny tepla

3.2.1 Výmenník tepla

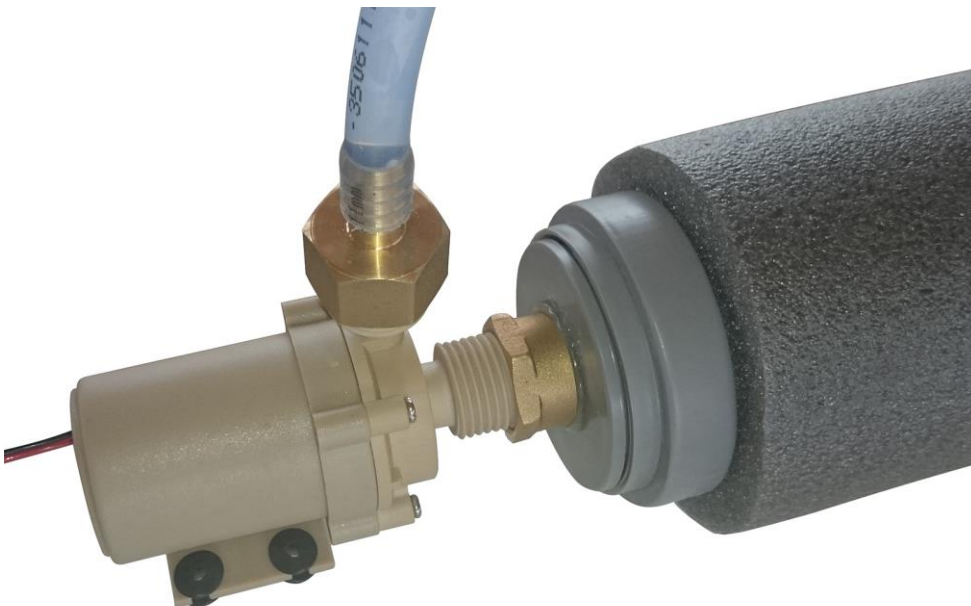
Ako bolo spomenuté, na výmenu tepla medzi slnečným žiarením a cirkulujúcou vodou bude treba výmenník tepla. Tento výmenník tepla nesmie poškodiť teplota do 200 °C a musí byť konštrukčne riešený tak, aby ním mohla cirkulovať voda, ktorá bude odvádzať teplo slnečného žiarenia z povrchu výmenníka. Tieto požiadavky spĺňa napríklad vodný chladič CPU. Jedná sa o dutý, hliníkový, štvorcový objekt, ktorý na jednej svojej strane obsahuje dva vývody na hadice. Jeden vývod slúži na prívod vody, druhý na jej odvod. Konštrukcia chladiču je navrhnutá tak, aby voda ním cirkulujúca odvádzala teplo z povrchu výmenníka. Takýto chladič sa dá kúpiť od 3.72 €.

3.2.2 Zásobníky kvapaliny

Ako zásobníky kvapaliny poslúžia plastové, vodovodné rúry. Dajú sa bežne kúpiť v rôznych inštalatérstvach, či hobby obchodoch. Zásobníky kvapaliny boli použité dva. Jeden referenčný, ktorého obsah sa nebude podieľať na cirkulácii vody a druhý pracovný. Referenčný zásobník je potrebný na to, aby bolo možné určiť množstvo prijatej energie vodou v pracovnom zásobníku. Pracovný zásobník kvapaliny tvorí vodoinštalčná rúra s priemerom 50 mm a dĺžkou 500 mm. Na jeden jej koniec sme pripevnili zátku a to tak, že koniec rúry sme nahriali nad plameňom, aby plast stratil pevnosť a stal sa pružnejším. Do otvoru sme potom natlačili zátku až po jej zarážku, čiže po jej koniec. Do tejto zátky bol v strede vyvŕtaný otvor a ním prestrčená hadica. Hadica bola prestrčená asi do dĺžky 5 mm a po jej obvode pripevnená silikónovým lepidlom k zátku, aby nedochádzalo k pretečeniu. Z druhej strany rúry je ten istý typ zátky, do ktorej bol taktiež vyvŕtaný otvor. Do tohto otvoru bola osadená mosadzná závitová spojka s veľkosťou závitov 1/2". Táto spojka bola taktiež prisilikónovaná k zátku. Na tejto strane rúry je otvor už prispôsobený zátku, takže nebolo treba nahrievať koniec rúry ako v predchádzajúcom prípade, ale stačilo do nej zátku zasunúť. Na túto závitovú spojku sa prikrútil jeden koniec čerpadla, konkrétne vstupný otvor, ktorý má závit zhodný s touto spojkou. Čerpadlo zabezpečí cirkuláciu vody v systéme. Táto rúra bola ešte obalená tepelnoizolačnou penou. Podobne bol skonštruovaný aj referenčný zásobník kvapaliny, s tým rozdielom, že do jeho zátiok sa nevŕtalo. Do oboch zásobníkov, približne v strede bola vyvŕtaná diera s priemerom 6,5 mm, aby sa ňou dal prestrčiť teplotný senzor dnu, do rúry. Tento senzor bol následne prilepený k rúre silikónom.

3.2.3 Cirkulácia

Cirkuláciu vody v systéme zabezpečuje čerpadlo (Obrázok 3.8). To ťahá vodu zo zásobníka kvapaliny cez gumenú hadičku s vnútorným priemerom 8mm a vonkajším priemerom 11mm do výmenníka tepla (Obrázok 3.9), ktorý je pevne ukotvený o rám so šošovkou. Tam sa voda zohreje a odtiaľ putuje ďalšou hadičkou späť do zásobníka kvapaliny. Hadička je poddajná, aby dobre znášala pohyb držáku rámu a následne aj pohyb rámu v tomto držáku.



Obrázok 3.8 Čerpadlo pripevnené o zásobník kvapaliny



Obrázok 3.9 Výmenník tepla, ktorým prechádza teplonosné médium - voda

4 Riadenie

4.1 Arduino

Celá konštrukcia sama o sebe nič nerobí, treba ju rozhybať a tento pohyb riadiť. Na riadenie sme zvolili platformu Arduino. Je to „open source“ elektronická platforma založená na jednoduchom používaní ako hardware, tak aj software. Je vhodná na rôzne projekty, ktoré si vyžadujú interakciu s okolím. To znamená, že pracuje so signálmi zo senzorov, napríklad fotosenzorov, akustických senzorov, tlačidiel, teplotných senzorov a pod. Zapojenie týchto senzorov k doske vyžaduje aspoň základné znalosti elektroniky. Je potrebné vedieť, čo je to rezistor, konektor, anóda, katóda a podobné pojmy. Konkrétne zapojenia jednotlivých senzorov a akčných členov k doske Arduino je možné nájsť na oficiálnych stránkach www.arduino.cc ako aj na množstve iných webových stránok zaoberajúcich sa touto tematikou, napríklad [6]. Taktiež existujú knihy venované konkrétnym projektom s ich podrobným popisom, zapojením a naprogramovaním, ako napríklad knihy [7] a [8]. Samotný hardware pozostáva z jednej dosky s mikroprocesorom. Na doske sú umiestené aj piny a konektory, cez ktoré komunikuje doska s okolím. Doska je programovateľná, písanie kódu, jeho kompilácia a stiahnutie do Arduina je možné cez originálny software, ktorý je voľne k stiahnutiu.¹

Najdôležitejšími pinmi sú analógové vstupy, digitálne vstupy/výstupy a napäťový výstup. Analógové vstupy umožňujú prijímať rôznu výšku signálu od 0 do 5 V. Toto napätie je procesorom prevedené na hodnotu medzi 0 a 1023. S touto hodnotou sa dá potom pracovať v programe ako so signálom zo senzoru.

Digitálne vstupy a výstupy, ďalej už iba digitálne I/O, definuje už ich názov. Dajú sa použiť buď ako vstup alebo výstup, ale s tou podmienkou, že ich hodnota je 0 alebo 1. To znamená, že ak je v programe poslaný výstup so signálom 1, napríklad na pin 5, tak fyzicky na doske vznikne medzi týmto pinom 5 a GND, napätie 5 V. Existujú aj špeciálne typy digitálnych I/O výstupov a tými sú výstupy PWM. Skratka PWM pochádza z anglického Pulse Width Modulation, čo znamená Impulzovo šírková modulácia. Princíp je založený na modulácii periodického signálu zmenou triedy (šírky impulzu). Signál sa tak potom javí ako analógový výstup [7].

Napäťový výstup je v podobe dvojice konektorov s + 5V a GND. Toto napätie sa používa na „budenie“ súčiastok, aby generovali signál, pre ne charakteristický.

¹ <http://www.arduino.cc/en/Main/Software>

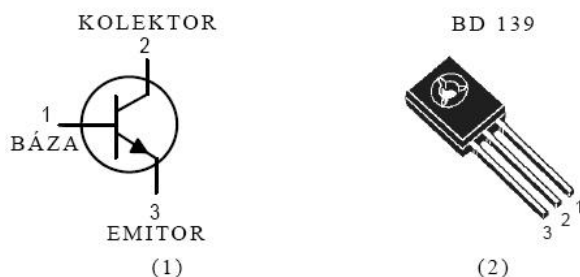
4.2 Elektronika

Ako bolo spomenuté, zapojenie senzorov k doske Arduino si vyžaduje určitú úroveň znalostí z elektroniky a elektronických obvodov. Na to, aby bol človek schopný zistiť ako majú byť jednotlivé súčiastky zapojené, musí poznať ich schematické značky a vedieť zadanú elektronickú schému zrealizovať v podobe fyzického zapojenia daných súčiastok medzi sebou. Elektronická schéma slúži len ako informácia o prepojení súčiastok ale nevypovedá nič o ich polohe pri zapojení. Je to len akýsi teoretický plán hovoriac ktorá súčiastka je s ktorou vodiivo spojená. Fyzicky sa dajú súčiastky spojiť rôznymi spôsobmi. Takzvaný „prototype board“ ponúka možnosť zostavenia vlastných elektronických obvodov tak, že jednotlivé súčiastky a vodiče sú zasúvané do dier, ktoré sú podľa určitého systému vodiivo prepojené. Ďalšou možnosťou je súčiastky leťovať na predvŕtané dosky. Najčastejšie sa však zapojenie súčiastok realizuje na doske plošných spojov - DPS. Aby bolo možné vyrobiť DPS je potrebné urobiť návrh vodivých ciest, otvorov na nožičky súčiastok, otvorov na skrutky, konektory a pod. Návrh DPS pre tento projekt sme vyhotovovali v programe EAGLE, ktorý je určený na vytváranie elektronických schém a následne ich návrhu DPS. Tento program bol zvolený preto, lebo firmy, ktoré sa zaoberajú výrobou DPS na Slovensku, akceptujú jeho súborové formáty ako predlohy do výroby. Výroba DPS pozostáva z viacerých krokov ako frézovanie, vŕtanie, drážkovanie dosky a je možné zvoliť aj rôzne, doplnkové vylepšenia dosky, ako je sieťotlač, maska a pod. Chceli sme však dosiahnuť čo najnižšie náklady, takže výroba našich DPS pozostávala iba z vyfrézovania vodivých ciest a vyvŕtania otvorov na súčiastky a uchyťavacie skrutky.

4.3 Ovládanie pohybu

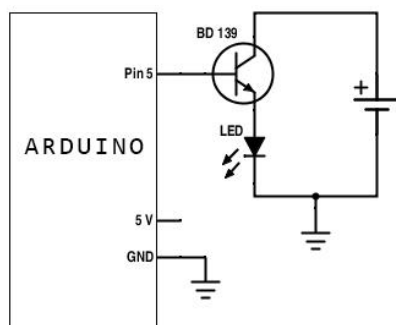
4.3.1 Tranzistor

Tranzistor alebo zriedkavo polovodičová trióda je polovodičová súčiastka, používaná ako zosilňovač, spínač, stabilizátor a modulátor elektrického napätia alebo prúdu [9]. Zjednodušene povedané, tranzistor sa dá použiť aj ako spínač. Od klasického spínača sa však líši tým, že ho nespína prst užívateľa zariadenia ale elektrický impulz. Tranzistor má tri vývody - bázu, emitor a kolektor. Elektrický impulz, ktorý je potrebný na otvorenie tranzistora, čiže zopnutia pomyselného spínača, je vedený do bázy. Medzi kolektorom a emitorom sa tak vytvorí vodivé spojenie – tranzistor sa otvorí a medzi týmito vývodmi tak môže tiecť elektrický prúd, ktorý je oveľa väčší ako je prúd medzi bázou a emitorom.



Obrázok 4.1 Schematická značka tranzistora (1), BD 139 s očíslovanými vývodmi (2)

V tomto projekte boli použité výkonové tranzistory BD 139 ako spínacie prvky. Tieto tranzistory sa otvárajú privedením napätia 5V medzi bázu a emitor, takže je vhodný pre naše použitie, pretože digitálny výstup z Arduina je pri hodnote HIGH, 5V. V skratke to znamená, že keď bázu tranzistora pripojíme na digitálny výstup Arduina, napríklad na pin 5 a na tento pin odošleme hodnotu 1(HIGH), otvorí sa prechod kolektor – emitor a týmto prechodom následne môže tiecť elektrický prúd a LED dióda zasvieti. Namiesto diódy sa dá pripojiť akékoľvek iné zariadenie, ktoré potrebuje na svoju činnosť elektrickú energiu, napr. motorček a pod.



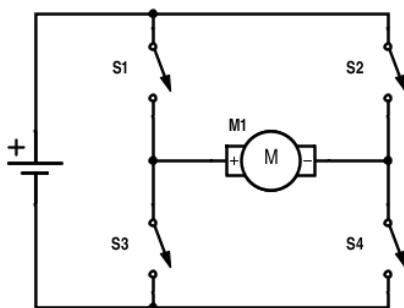
Obrázok 4.2 Základné zapojenie tranzistora s Arduino

Ďalšie definície, či odbornejší popis tranzistora nie je pre pochopenie jeho použitia v tomto projekte potrebný.

4.3.2 H-mostík

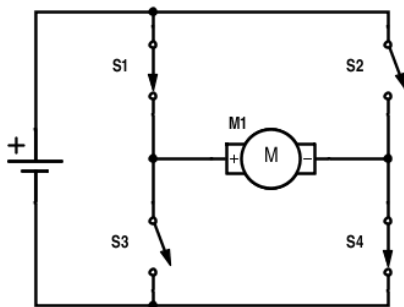
Na otáčanie rámom a držiakom rámu sú použité motorčeky na jednosmerné napätie 12 voltov. Motorčeky majú svorky so značkami + (plus) a – (mínus) a sú reverzibilné, čiže ak pripojíme kladné napätie zdroja na konektor motorčeka so značkou plus a zem zdroja na konektor so

značkou mínus, hriadeľ motorčeka sa bude točiť v smere hodinových ručičiek. Ak kontakty vymeníme, bude sa točiť proti smeru hodinových ručičiek bez poškodenia motorčeka. Riadiť smer otáčania motorčeka priamo Arduinoom sa nedá, pretože napätie z jeho digitálneho výstupu nie je dostačujúce na pohon motorčeka. Preto je potrebné navrhnuť taký elektronický obvod, ktorý bude schopný signály z digitálnych výstupov Arduina interpretovať a napájať motorček z iného zdroja napätia tak, aby sa jeho hriadeľ otáčal požadovaným smerom. K tomuto účelu si postavíme tzv. H-mostík. H-mostík je kombinácia zapojenia štyroch tranzistorov, ktorá len na základe správneho spínania tranzistorov umožňuje napájanie motorčeka tak, aby sa otáčal požadovaným smerom. Ako bolo spomenuté, tranzistory použijeme ako spínače, takže nasledujúca schéma demonštruje princíp H-mostíka.



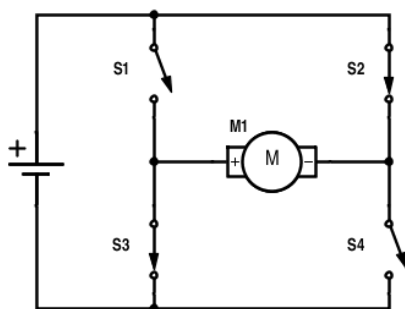
Obrázok 4.3 Principiálna schéma H-mostíka

Ak zopneme spínače 1 a 4, k vývodu s označením plus na motorčeku sa dostane kladné napätie a ten sa bude točiť v smere hodinových ručičiek, ako je vidno na obrázku 4.4



Obrázok 4.4 Motorček sa točí v smere hodinových ručičiek

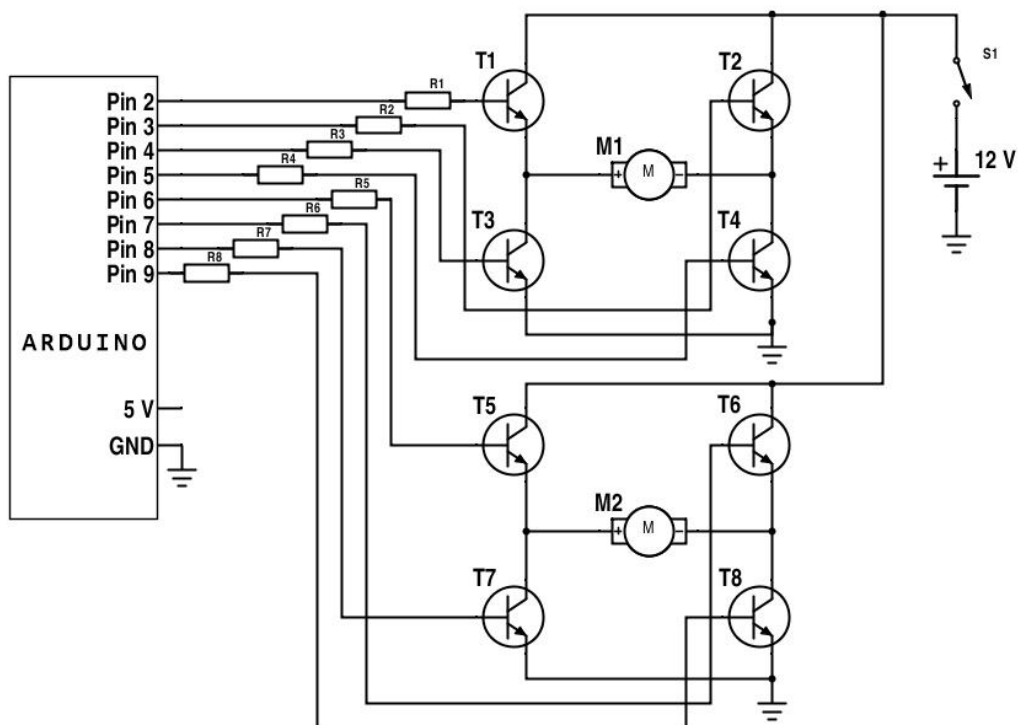
Ak zopneme spínače 2 a 3, kladné napätie zo zdroja sa dostane na vývod motorčeka s označením mínus a ten sa bude otáčať opačným smerom.



Obrázok 4.5 Motorček sa točí v protismere hodinových ručičiek

Ak sa zopnú spínače v nesprávnej kombinácii, napríklad 1 a 3 alebo 2 a 4, príde ku skratu napájacieho zdroja a ten sa následne môže poškodiť, preto je potrebné dbať na správne zapojenie a indexovanie tranzistorov v zapojení. Viac o H-mostíku sa dá dočítať v knihe [7].

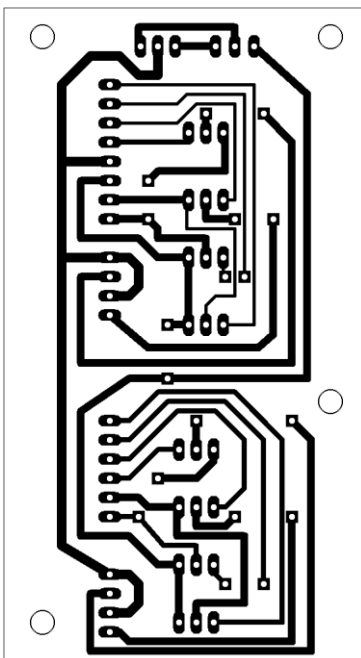
Naša schéma zapojenia v súčinnosti s Arduino, ktorú sme navrhli vyzerá nasledovne.



Obrázok 4.6 Navrhnutá schéma H-mostíkov

Namiesto spínačov sú tranzistory a tie sú spínané digitálnymi výstupmi z Arduina, ktoré vedú do bázy tranzistorov. Skupina tranzistorov 1 až 4 tvoria H-mostík pre motorček 1, ktorý otáča rámom hore a dolu. Tranzistory 5 až 8 tvoria H-mostík pre motorček 2, ktorý hýbe držiakom rámu doľava a doprava. Obidva H – mostíky premost'ujú do k nim prislúchajúcich motorčekov napájanie z jedného zdroja s napätím 12 V. Hodnoty rezistorov medzi bázami tranzistorov a Arduino su všetky rovnaké, 270 Ω .

Podľa schémy sme navrhli dosku plošných spojov. Táto doska okrem tranzistorov obsahuje aj miesto pre tranzistor spínajúci napätie k čerpadlu, čo je predmetom 4.6. podkapitoly.



Obrázok 4.7 Návrh DPS

Do nej sme priletovali príslušné súčiastky a ešte vizuálne skontrolovali správnosť zapojenia. Následne sme premerali aj či sú všetky kontakty vodivo spojené tak, ako je vyobrazené na schéme, pomocou elektrického multimetra. Keďže ešte nemáme zapojené fotosenzory, tak systém nevie kam sa má natáčať a či vôbec sa má hýbať. Preto potrebujeme zapojiť k Arduinu mini klávesnicu, cez ktorú mu môžeme dávať pokyny ktorým smerom a ktorým motorčekom má otáčať.



Obrázok 4.9 Fotka klávesnice (keypad)

Stlačenie čísla 2 znamená otáčanie rámom hore, 8 zasa dolu. Stlačenie čísla 4 znamená otáčanie držiakom rámu doľava a 6 znamená doprava. Stlačením tlačidla 5 sa oba motorčeky zastavia. Táto mini klávesnica sa dá už vyhotovená kúpiť ako je na obrázku 4.9, takže nie je potrebné navrhovať dosku plošných spojov a ani nič letovať, stačí ju správne pripojiť priamo na Arduino.

Posledným krokom je programovanie Arduina. Algoritmus je v podobe vývojového diagramu naznačený v schéme 4.1. Začiatok znamená rutinné importovanie potrebných knižníc a ďalšie riadky obligátneho kódu, ktorý nemá principiálny vplyv na algoritmus. Na začiatku sa čaká na stlačenie jednej z kláves. Podľa stlačeného čísla je potom vyhodnotený výstup z Arduina, ktorý pošle impulz do báz príslušných tranzistorov. HIGH znamená, že výstup je aktivovaný, LOW zasa, že je vypnutý. Klávesnicu sme pripojili na digitálne výstupy: 30, 32, 34, 36, 38, 40, 42, 44, pretože takto sú na doske jeden vedľa druhého čo umožňuje pohodlné pripojenie klávesnice. Správne pripojenie klávesnice k Arduinu a jej naprogramovanie je názorne vysvetlené v článku [10], odkiaľ sme aj my použili časti kódu, ktoré pracujú s klávesnicou.

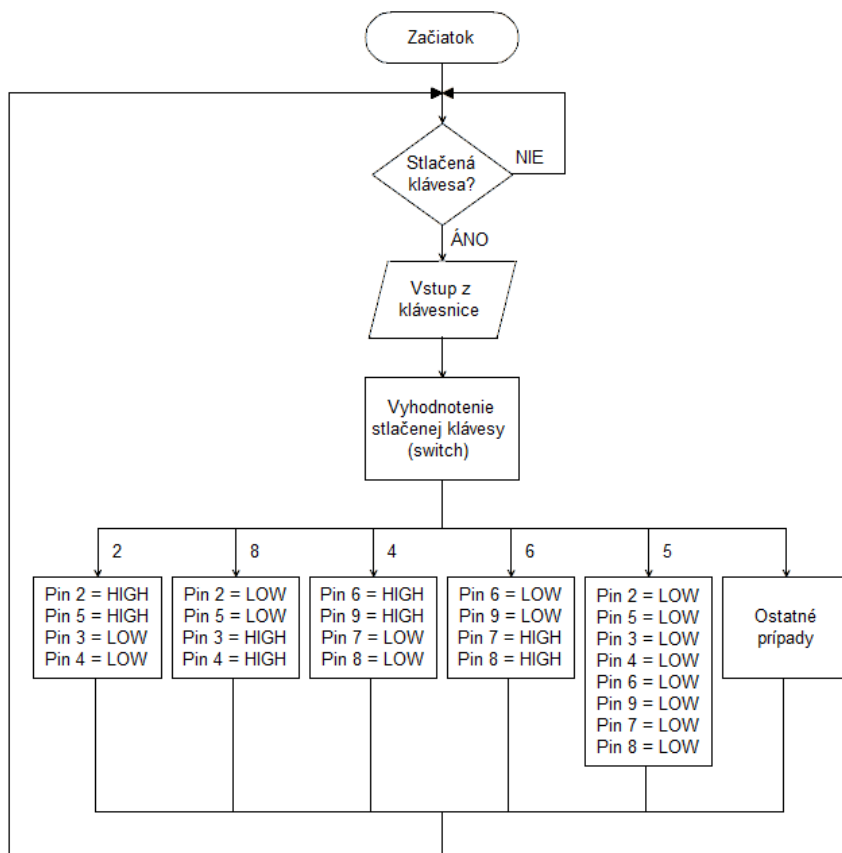


Diagram 4.1 Vývojový diagram algoritmu na ovládanie pohybu systému OptiFres

Výpis 4.1 Otáčanie systémom manuálne

```
#include <Keypad.h> // načítanie knižnice potrebnej ku klávesnici
const byte numRows= 4; // počet riadkov klávesnice
const byte numCols= 4; // počet stĺpcov klávesnice

// zadefinovanie znakov klávesnice presne v poradí, v akom sú na nej
char keymap[numRows][numCols]=
{
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};

// do premenných rowPins a colPins sú uložené čísla pinov ku ktorým je
//pripojená klávesnica
byte rowPins[numRows] = {30,32,34,36};
byte colPins[numCols]= {38,40,42,44};

//vytvorenie inštancie triedy Keypad
Keypad myKeypad= Keypad(makeKeymap(keymap), rowPins, colPins, numRows,
numCols);

//adresy digitálnych výstupov pre jednotlivé tranzistory
const int T1 = 2;
const int T2 = 3;
const int T3 = 4;
const int T4 = 5;
const int T5 = 6;
const int T6 = 7;
const int T7 = 8;
const int T8 = 9;

void setup(){
    pinMode(T1, OUTPUT);
    pinMode(T2, OUTPUT);
    pinMode(T3, OUTPUT);
    pinMode(T4, OUTPUT);
    pinMode(T5, OUTPUT);
    pinMode(T6, OUTPUT);
    pinMode(T7, OUTPUT);
    pinMode(T8, OUTPUT);
}

void loop(){
    // do premennej keypressed sa uloží stlačený znak na klávesnici
    char keypressed = myKeypad.getKey();
    if (keypressed != NO_KEY){
        switch(keypressed){
            // na základe stlačeného čísla je volaná príslušná funkcia
            case '2':{hore(); break;}
            case '8':{dole(); break;}
            case '4':{vlavo(); break;}
            case '6':{vpravo(); break;}
        }
    }
}
```

```

        case '5':{stopni(); break;}
    }
}
    delay(20);
}

void dole(){ // každý akčný zásah má svoju vlastnú funkciu
    digitalWrite(T1, HIGH);
    digitalWrite(T4, HIGH);
    digitalWrite(T2, LOW);
    digitalWrite(T3, LOW);
}

void hore(){
    digitalWrite(T2, HIGH);
    digitalWrite(T3, HIGH);
    digitalWrite(T1, LOW);
    digitalWrite(T4, LOW);
}

void vpravo(){
    digitalWrite(T5, HIGH);
    digitalWrite(T8, HIGH);
    digitalWrite(T6, LOW);
    digitalWrite(T7, LOW);
}

void vlavo(){
    digitalWrite(T6, HIGH);
    digitalWrite(T7, HIGH);
    digitalWrite(T5, LOW);
    digitalWrite(T8, LOW);
}

void stopni(){
    digitalWrite(T1, LOW);
    digitalWrite(T2, LOW);
    digitalWrite(T3, LOW);
    digitalWrite(T4, LOW);
    digitalWrite(T5, LOW);
    digitalWrite(T6, LOW);
    digitalWrite(T7, LOW);
    digitalWrite(T8, LOW);
}

```

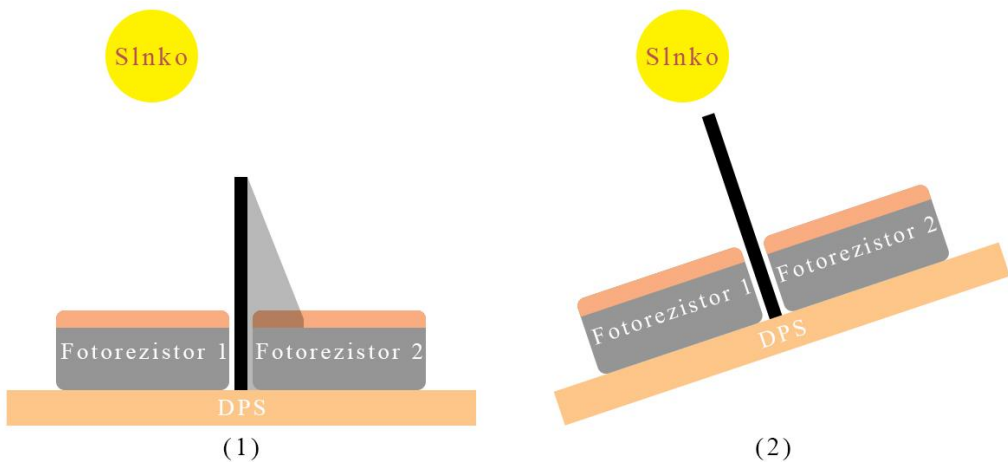

4.3.3 Odkúšanie

Po prepojení dosiek medzi sebou je treba toto prepojenie ešte minimálne raz skontrolovať pred tým, než zapneme Arduino a pripojíme zdroj napätia spínačom S1. Ak je všetko správne, môžeme zapnúť Arduino buď pripojením k počítaču alebo 9 voltovej batérie k príslušnému konektoru. Potom môžeme zopnúť spínač S1.

Systém motorčekov fungoval podľa očakávaní, t.j. po stlačení nejakého z ovládajúcich čísel sa rám, či držiak rámu otáčali príslušným smerom a po stlačení čísla 5 sa zastavili. Ostatné stlačené klávesy boli ignorované. Pri skúšaní sa vždy ukázu nedostatky, ktoré boli opomenuté alebo zabudnuté pri navrhovaní. V tomto prípade je to chladenie tranzistorov. Po dlhšej dobe nepretržitého otáčania (asi 3 minúty) sa začali zohrievať. Chyba to nie je, pretože natočenie Fresnelovej šošovky za Slnkom prebehne rádovo v sekundách, ale je to varovanie do budúcnosti, že pri iných aplikáciách tranzistorov treba myslieť aj na ich chladenie.

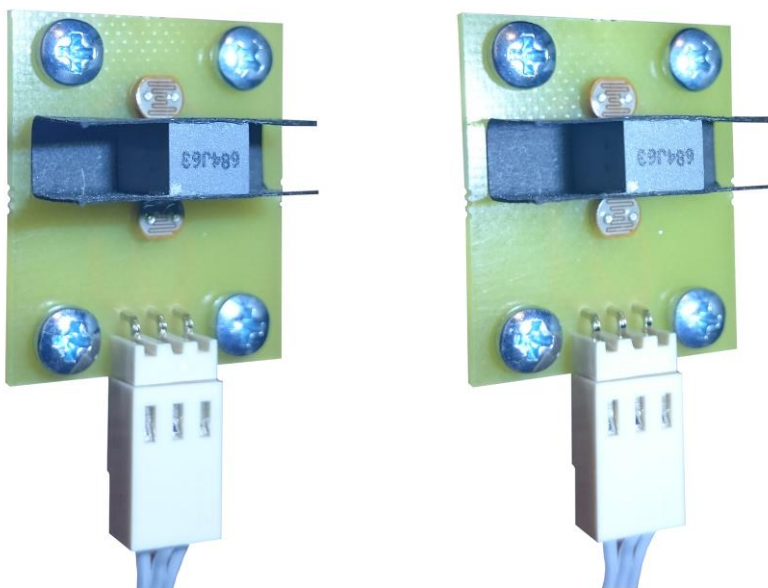
4.4 Sledovanie Slnka

Jednou z úloh projektu je automatické sledovanie pozície Slnka. Z toho vyplýva potreba poznať, kde na oblohe sa Slnko nachádza. Na vyriešenie tohto problému použijeme sústavu fotosenzorov. Fotosenzor je také zariadenie, ktoré je citlivé na intenzitu svetla. Skladá sa z fotorezistora a rezistora. Vieme, že v tieni je intenzita svetla menšia ako na priamom Slnku a presne tento základný poznatok využijeme na to, aby sme boli schopní určiť kde sa nachádza Slnko na oblohe. Ak si oblohu predstavíme ako súradnicový systém s osou X a osou Y, potom konkrétnu polohu Slnka bude možné určiť súradnicami X a Y. To znamená, že hľadanie Slnka bude prebiehať v dvoch etapách a to určovanie súradnice X a potom určovanie súradnice Y. Máme už vyrobený otáčavý systém, ktorý sa otáča v dvoch osiach – vertikálnej a horizontálnej. Otáčanie okolo vertikálnej osi využijeme na určenie súradnice X a otáčanie okolo horizontálnej osi zasa na určenie súradnice Y. Vytvorili sme sústavu fotosenzorov tak, že na základe signálu z nich sme schopní určiť, či je treba systém natočiť na jednu, či na druhú stranu. Dva fotorezistory sú umiestnené tesne vedľa seba, no medzi nimi je prekážka. Táto prekážka spôsobí, že keď je dvojica senzorov otočená inak ako kolmo na zdroj svetla - Slnko, vznikne tieň na jednom zo senzorov.



Obrázok 4.10 Bočný rez dosky s fotorezistormi mimo Slnka (1), alebo kolmo na Slnko (2)

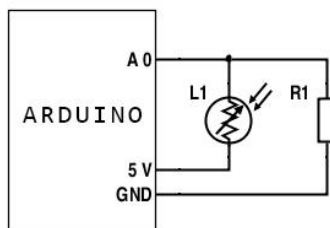
Na základe porovnania ich signálov je možné určiť či sa svetelný zdroj nachádza vpravo(hore) alebo vľavo(dolu) od tejto sústavy senzorov. Fotorezistor na ktorom je tieň má väčší elektrický odpor a teda signál z neho je menší. Jedna dvojica sníma intenzitu slnečného svitu tak, že signály z nich sú relevantné pri vyhodnocovaní vertikálnej polohy Slnka a druhá dvojica poskytuje signály potrebné k vyhodnoteniu horizontálnej polohy Slnka. Využitím týchto informácií je potom možné riadiť motorčeky tak, aby bol rám natočený kolmo na Slnko.



Obrázok 4.11 Reálne zhotovenie sústav fotosenzorov

4.4.1 Fotosenzory

Keď už je principiálne jasné, ako chceme dosiahnuť sledovania Slnka, je potrebné tieto sústavy fotosenzorov zrealizovať. Ako bolo spomenuté, fotosenzor sa skladá z fotorezistora a klasického rezistora. Ako fotorezistor sme zvolili typ VT93N2 a k nemu sme pripojili 10K Ω rezistor. Túto schému sme prevzali z návodu, ktorý sa nachádza na oficiálnej stránke Arduino [11]. Elektornické schémy zostavujeme na webovej stránke [12], ktorá ponúka registrovaným užívateľom vytváranie, ukladanie a exportovanie vlastných elektronických schém.

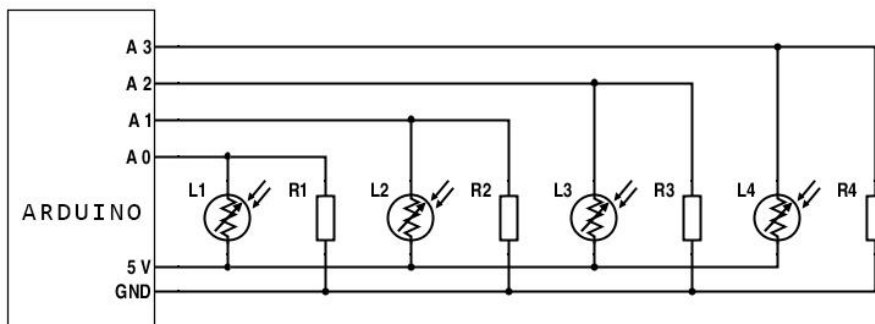


Obrázok 4.12 Schéma základného zapojenia fotosenzora k Arduino



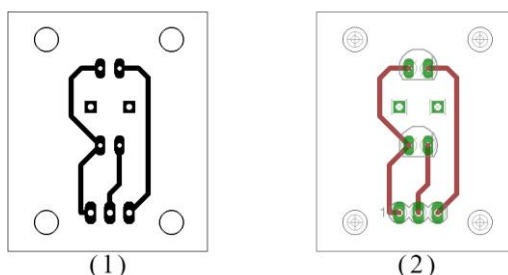
Obrázok 4.13 Fotorezistor VT93N2

Na vyhotovenie našich foto sústav potrebujeme takto zapojiť 4 fotosenzory.



Obrázok 4.14 Schéma zapojenia štyroch fotosenzorov

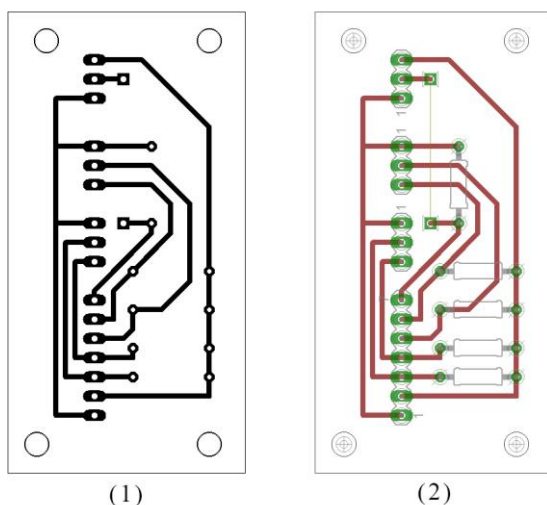
Ako vidno na schéme, senzory sú zapojené do analógových vstupov, pretože tie sú schopné prijímať úroveň signálu a nie len hodnoty 1 alebo 0. Typ fotorezistora aj rezistora je vo všetkých štyroch prípadoch rovnaký. Zapojenie tejto schémy sme však rozdelili na tri časti. Dve identické časti pozostávajú z fotorezistorov, z čoho jedna bude umiestnená na ráme so šošovkou a druhá na držiaku rámu otočená o 90°. Tieto časti možno vidieť zrealizované na obrázku 4.11. Druhá časť pozostáva z rezistorov. Výsledkom sú teda dva rôzne návrhy (z jedného návrhu dva kusy) DPS, ktoré sú dokopy realizáciou jednej schémy zapojenia.



Obrázok 4.15 Návrh DPS s fotorezistormi v mierke 1:1 (1), plán osadenia súčiastok (2)

Do takejto dosky sme potom priletovali súčiastky, čím sme dostali elektronický obvod s dvomi fotorezistormi a konektorom so zámkom (Obrázok 4.11).

Druhá časť, pozostávajúca len z rezistorov a konektorov je osadená v riadiacej jednotke, vedľa dosky Arduino. Riadiaca jednotka je plastová krabička nachádzajúca sa v strednej časti podstavca a je tak nazvaná preto, lebo obsahuje dosky Arduino a elektroniky, čiže je to také „riadiace stredisko“. V tejto riadiacej jednotke je umiestnená aj doska s H-mostíkmi.



Obrázok 4.16 Návrh DPS s rezistormi (1), plán osadenia súčiastok (2), mierka (1:1)

Ako vidno na návrhu DPS, ten obsahuje aj rezistor, ktorý nie je v zapojení s fotorezistormi. Je to ten horný, natočený kolmo na ostatné rezistory. Je to rezistor pre teplotný senzor, ktorý sa nachádza na výmenníku tepla.

4.5 Riadenie pohybu

Máme pohyblivú konštrukciu a máme zhotovené sústavy fotosenzorov, podľa ktorých je možné určiť ktorým smerom je Slnko. Ako bolo spomenuté v predchádzajúcej podkapitole, Slnko budeme hľadať v dvoch fázach a to na osi X a potom na osi Y.

4.5.1 Hľadanie Slnka na osi X

Držiak rámu je časť konštrukcie, ktorá je osadená na podstavci a v tomto podstavci je schopná otáčať sa okolo vertikálnej osi, t.j. natáčať sa doprava alebo doľava. Jednu sústavu fotosenzorov umiestnime, teda priskrutkujeme na tento držiak rámu. Hodnotu signálu z ľavého fotosenzora pomenujeme F1 a z pravého F2.

Uvedme si príklad. Ak svieti Slnko na túto sústavu fotosenzorov z pravej strany, vytvorí sa tieň na ľavom fotosenzore. To znamená, že hodnota signálu z F1 bude menšia ako z F2. Na základe tohto rozdielu signálov sa potom odošle pokyn na motorček otáčajúci držiakom rámu, aby krútil týmto držiakom doprava a tým zmenšoval tieň na ľavom fotosenzore, až kým úplne nezmizne a držiak rámu bude tým pádom natočený k Slnku.

4.5.2 Hľadanie Slnka na osi Y

Analogicky ako v predchádzajúcom prípade sa bude postupovať aj teraz. Na rám so šošovkou sa umiestni druhá sústava fotosenzorov. Tento krát tak, že jeden bude dolu a jeden hore. Ak bude tieň na spodnom fotosenzore, rám sa začne otáčať hore až kým tieň nezmizne. Ak bude tieň na hornom fotosenzore, rám sa otočí smerom dolu.

Ak teda odstránime tieň z prvej sústavy senzorov a následne aj z druhej, znamená to, že rám s Fresnelovou šošovkou je natočený kolmo na smer šírenia elektromagnetického žiarenia emitovaného Slnkom. V skratke, rám je otočený za Slnkom.

4.5.3 Algoritmus

Teoreticky je teda princíp natáčania jednotlivých častí systému jasný, takže môžeme pristúpiť k jeho naprogramovaniu. Riadená veličina je v tomto prípade rozdiel signálov medzi pravým (dolným) alebo ľavým (horným) fotosenzorom. Žiadaná veličina je nula, keďže nám ide o to, aby sa tieto signály rovnali. Riadiacou veličinou je smer otáčania motorčeka. V tomto prípade sme nepoužili žiadny regulátor, ani sme nesledovali systém za účelom identifikácie jeho prenosu. Riadenie sme zabezpečili jednoduchým algoritmom, ktorý ako sa ukázalo pri testovaní, je funkčný a plní účel. Vývojový diagram 4.2 znázorňuje algoritmus natáčania držiaku rámu vpravo a vľavo.

Po začiatku sa vyhodnotí, či bolo stlačené tlačidlo na klávese, ak áno, program vstúpi do módu manuálneho riadenia, ktoré bolo popísané na začiatku kapitoly. Ak tlačidlo stlačené nebolo, začne sa regulácia. Najprv sa vyhodnotí rozdiel signálov z fotosenzorov, nazvime si ho r . Ak je jeho absolútna hodnota menšia ako 2, znamená to, že regulácia nie je potrebná. Ak je však väčšia, určí sa dĺžka kroku motorčeka v milisekundách. Znamená to, ako dlho bude motorček spustený. Ak je hodnota r menšia ako 10, krok motorčeka je menší a tým pádom je regulácia jemnejšia. Možno sa zdá, že krok 50 alebo 20 ms je primálny, no pohyb motorčeka sa v konečnom dôsledku javí ako spojitý. Určili sme si, že ľavý fotosenzor je F1 a pravý F2. Ak je hodnota r menšia ako nula, znamená to, že tieň je na ľavom fotosenzore a držiak rámu sa musí otočiť doprava. Takto sa to opakuje dovtedy, pokiaľ nie je absolútna hodnota r jedna alebo nula. Delta okolie 1 sme zvolili preto, lebo signál z fotosenzorov je mierne zašumený a v ustálenom stave kmitá okolo hodnoty s odchýlkou ± 1 .

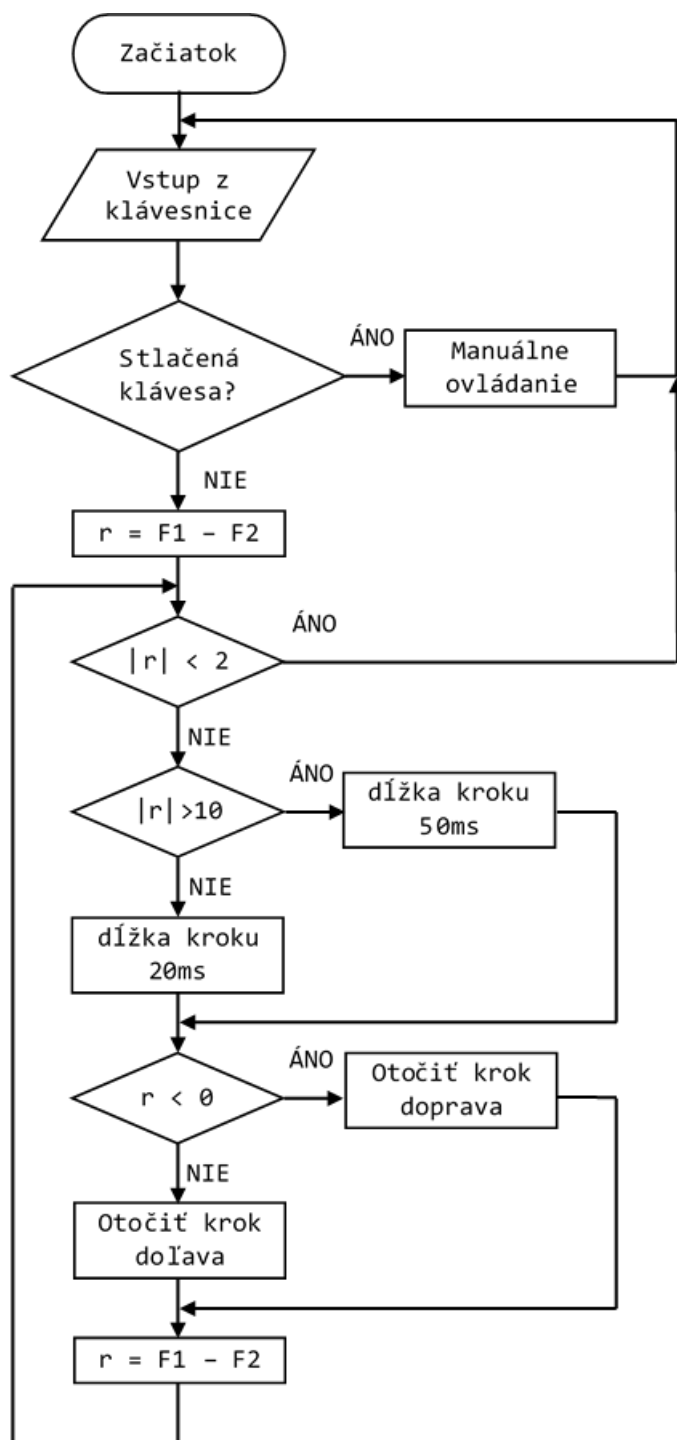


Diagram 4.2 Automatické natáčanie za Slnkom okolo vertikálnej osi

Výpis 4.2 Automatické natáčanie za Slnkom

```
#include <Keypad.h>
const byte numRows= 4;
const byte numCols= 4;
char keymap[numRows][numCols]=
{
{'1', '2', '3', 'A'},
{'4', '5', '6', 'B'},
{'7', '8', '9', 'C'},
{'*', '0', '#', 'D'}
};
byte rowPins[numRows] = {30,32,34,36};
byte colPins[numCols]= {38,40,42,44};
Keypad myKeypad= Keypad(makeKeymap(keymap), rowPins, colPins, numRows,
numCols);

//adresy digitálnych výstupov pre jednotlivé tranzistory
const int H1 = 2;
const int H2 = 3;
const int H3 = 4;
const int H4 = 5;
const int V1 = 6;
const int V2 = 7;
const int V3 = 8;
const int V4 = 9;

//adresy analógových vstupov pre príslušné fotosenzory
int FH1 = 0;
int FH2 = 1;
int FV1 = 2;
int FV2 = 3;

void setup(){
pinMode(H1, OUTPUT);
pinMode(H2, OUTPUT);
pinMode(H3, OUTPUT);
pinMode(H4, OUTPUT);
pinMode(V1, OUTPUT);
pinMode(V2, OUTPUT);
pinMode(V3, OUTPUT);
pinMode(V4, OUTPUT);
}

void loop(){
char keypressed = myKeypad.getKey();
if (keypressed != NO_KEY)
{
int manual = 1;
while(manual == 1){
keypressed = myKeypad.getKey();
switch(keypressed){
case '2':{otocenie(H2, H3, H1, H4); break;}
case '8':{otocenie(H1, H4, H2, H3); break;}
case '4':{otocenie(V2, V3, V1, V4); break;}
case '6':{otocenie(V1, V4, V2, V3); break;}
}
```



```

        case '5':{stopni(); break;}
        case '*':{manual=0; break;} //výstup z manuálneho módu
    }
    delay(20);
}
}
//automatické natáčanie rámu sa začne len ak svieti Slnko
if(analogRead(FH1) > 800 || analogRead(FH1) > 800 || analogRead(FH1) >
800 || analogRead(FH1) > 800){
    //regulácia otočenia okolo vertikálnej osi
    regulacia(FV1, FV2, V1, V1, V3, V4);
    //regulácia otočenia okolo horizontálnej osi
    regulacia(FH1, FH2, H1, H2, H3, H4);
}
delay(200);
}

```

```

void regulacia(int Foto1, int Foto2, int T1, int T2, int T3, int T4){
    int F1 = analogRead(Foto1);
    int F2 = analogRead(Foto2);
    int r = F1 - F2;
    int krok;
    while(abs(r) > 1){
        if(abs(r) > 10) krok = 50;
        else krok = 20;
        if(r > 0){
            otocenie(T2, T3, T1, T4); // otočenie vľavo/hore
            delay(krok);
            stopni();
        }
        else {
            otocenie(T1, T2, T3, T4); // otočenie vpravo/dolu
            delay(krok);
            stopni();
        }
        r = analogRead(Foto1) - analogRead(Foto2);
    }
}

```

```

void otocenie(int T1, int T2, int T3, int T4){
    digitalWrite(T1, HIGH);
    digitalWrite(T4, HIGH);
    digitalWrite(T2, LOW);
    digitalWrite(T3, LOW);
}

```

```

void stopni(){
    digitalWrite(V1, LOW);
    digitalWrite(V2, LOW);
    digitalWrite(V3, LOW);
    digitalWrite(V4, LOW);
    digitalWrite(H1, LOW);
    digitalWrite(H2, LOW);
    digitalWrite(H3, LOW);
    digitalWrite(H4, LOW);
}

```

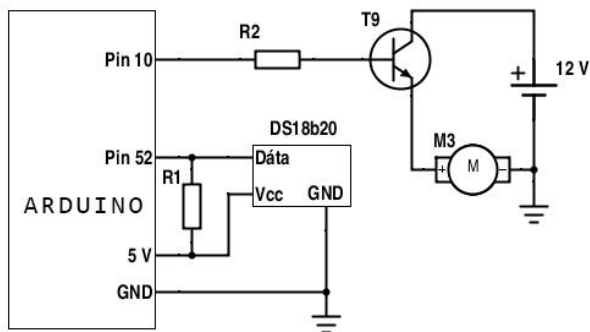
Vo výpise 4.2 je možné si všimnúť istého vylepšenia oproti výpisu 4.1, kde sme mali pre každý pohyb motorčeka zvlášť funkciu. Teraz je na to len jedna funkcia, ktorá ako vstupné argumenty prijíma adresy digitálnych výstupov, na ktorých sú pripojené tranzistory. Správnym poradím týchto adries ako argumentov funkcie pri jej volaní dosiahneme otáčanie konkrétneho motorčeka požadovaným smerom.

4.6 Spúšťanie čerpadla

V cieľoch práce je, okrem iného, aj spomenuté, že systém by mal byť schopný zohrievať vodu v zásobníku kvapaliny. Toto zohrievanie vody sa deje tak, že tá preteká výmenníkom tepla, ktorý je zohrievaný Slnkom. Do výmenníka tepla je hnaná čerpadlom, ale občas sa však stáva, že Slnko na chvíľku zájde za mraky, či úplne sa zamračí a slnečný deň sa zmení na tmavý a sychravý. V tom prípade by bolo absolútne zbytočné, aby čerpadlo hnalo vodu do systému, cez studený výmenník tepla. Preto sme umiestnili na zadnú stranu výmenníka tepla teplotný senzor. Ešte pred jeho umiestnením sme pri priebežných testoch systému merali teplotu na tejto strane výmenníka. Použili sme na to teplomer merajúci teplotu na základe odozvy infračerveného svetla. Konštrukciu sme umiestnili na miesto dobre osvetlené Slnkom, zapli sme automatické natáčanie za Slnkom a priebežne merali teplotu na zadnej strane výmenníka tepla. Výmenník tepla je hliníkový, štvorcový objekt so stranou 41 mm a hrúbkou 12 mm, čiže teplo z prednej časti je rýchlo odvedené do celého jeho objemu vrátane časti zadnej. Najvyššia teplota akú sme namerali na tejto časti výmenníka bola 140° C. Z predchádzajúcich faktov sme usúdili, že umiestnenie senzora na zadnú stranu je postačujúce na to, aby dokázal relevantne zmerať teplotu vhodnú na spustenie čerpadla. V praxi to teda znamená, že ak sa výmenník tepla zohreje na teplotu vyššiu ako 50° C, zapne sa čerpadlo. Cirkulácia vody začne výmenník tepla chladiť a čerpadlo sa následne vypne ak teplota na výmenníku klesne pod 35° C. Tieto teploty sme si zvolili len na základe vlastného úsudku a dajú sa veľmi ľahko, kedykoľvek zmeniť a ďalej s týmto teplotným intervalom experimentovať.

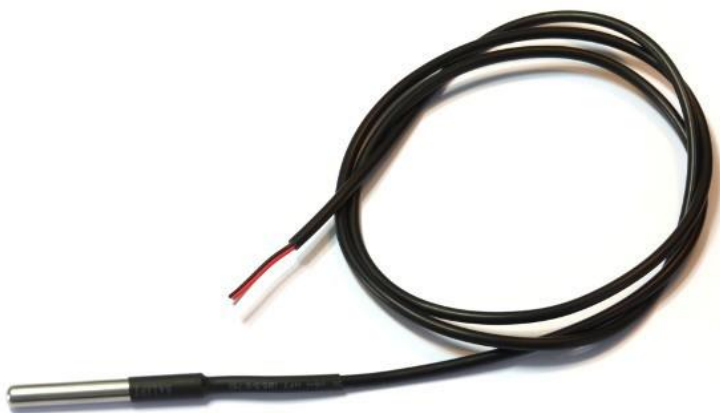
4.6.1 Teplotný senzor

Ako teplotný senzor sme použili DS18B20. Je ľahko dostupný, lacný a vhodný k nášmu použitiu. O jeho zapojení a programovaní sme čerpali informácie z internetu, konkrétne z webovej stránky [13].



Obrázok 4.16 Schéma zapojenia teplotného senzoru a čerpadla k arduinu

Rezistor R2 má hodnotu $270\ \Omega$ a rezistor R1, $4,7\ k\Omega$. Motorček M3 je motorček čerpadla. Konektor a rezistor pre teplotný senzor je na doske, na obrázku 4.16 v hornej časti dosky. Stačí ak pripojíme senzor na vstupný konektor dosky a výstup na digitálny vstup Arduina, napríklad na pin 52.



Obrázok 4.17 Teplotný senzor DS18b20

4.6.2 Zapnutie čerpadla

Ako spínač čerpadla sme zvolili opäť výkonový tranzistor BD 139, ktorý je umiestnený na doske s tranzistormi (Obrázok 4.7), vo vrchnej časti. Bázu tohto tranzistora pripojíme na digitálny výstup Arduina číslo 10 a čerpadlo pripojíme na príslušný konektor na doske.

Ak máme zapojenie hotové, môžeme zdrojový kód vo výpise 4.2 obohatiť o nasledovné riadky.

Výpis 4.3 Spúšťanie čerpadla v závislosti od teploty

```
#include <OneWire.h>
#include <DallasTemperature.h>

#define ONE_WIRE_BUS 52 //teplotný senzor je pripojený na I/O 52

OneWire oneWire(ONE_WIRE_BUS);

DallasTemperature sensors(&oneWire);

void setup()
{
    pinMode(10, OUTPUT); // na digitálny výstup 10 je pripojené čerpadlo
    sensors.begin();
}

void loop(){

    char keypressed = myKeypad.getKey();
    if (keypressed != NO_KEY)
    {
        manual = 1;
        while(manual == 1){
            keypressed = myKeypad.getKey();
            switch(keypressed){
                case '2':{otocenie(H2, H3, H1, H4); break;}
                case '8':{otocenie(H1, H4, H2, H3); break;}
                case '4':{otocenie(V2, V3, V1, V4); break;}
                case '6':{otocenie(V1, V4, V2, V3); break;}
                //manuálne zapnutie čerpadla
                case 'C':{digitalWrite(10, HIGH);break;}

                //manuálne vypnutie čerpadla
                case 'D':{digitalWrite(10, LOW);break;}
                case '5':{stopni(); break;}
                case '*':{manual=0; break;}
            }
            delay(20);
        }
    }

    sensors.requestTemperatures();
    //uloží hodnotu teploty do premennej
    float teplota = sensors.getTempCByIndex(0);
    if(teplota > 50)
        digitalWrite(10 HIGH);//čerpadlo sa zapne
    if(teplota < 35)
        digitalWrite(10 LOW);//čerpadlo sa vypne

    delay(200);
}
```

Šedé časti vo výpise sú časti z pôvodného kódu, slúžiace na orientáciu umiestnenia kódu nového. Ako vidno v zdrojovom kóde, vytvorili sme aj možnosť manuálneho spúšťania a vypínania čerpadla.

4.7 Optimálne riadenie

To, či je takéto riadenie Fresnelovej šošovky optimálne sa nedá dokázať matematicky, no boli podniknuté viaceré kroky, ktoré robia toto riadenie optimálnym v rámci istých možností.

Prvým takým krokom je samotný princíp natáčania šošovky. Sústavy fotosenzorov sú vytvorené tak, že je možné podľa ich signálov natočiť rám so šošovkou na Slnko priamo a bez iných zbytočných pohybov. Toto zapojenie nie je prvoplánové a je výsledkom experimentovania a zdokonaľovania procesu hľadania Slnka na oblohe.

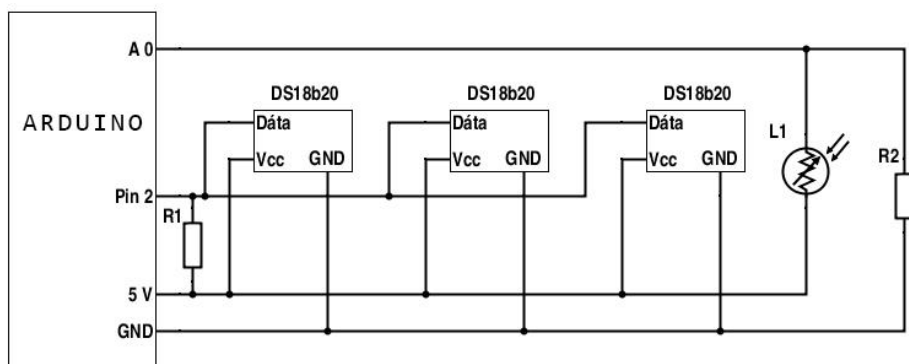
Ďalším krokom je to, že automatické natáčanie šošovky sa spustí len vtedy, ak je zaznamenaná aspoň minimálna intenzita slnečného žiarenia, na ktoromkoľvek fotosenzore. Bez dostatočného slnečného svitu, by nebolo čo koncentrovať, tým pádom by sa výmenník tepla nezohrieval a natáčanie šošovky by sa stalo zbytočným.

Cirkulácia vody si tiež vyžaduje energiu v podobe napájania čerpadla. Toto napájanie sme zautomatizovali tak, že cirkulácia začne až keď je výmenník tepla vyhriaty na istú teplotu a nedochádza tým k plytvaniu energie v podobe permanentne spusteného čerpadla.

5 Zber a uchovávanie dát

V tejto fáze projektu už sa rám s Fresnelovou šošovkou automaticky natáča za Slnkom, čerpadlo sa zapína len keď to má efekt a voda vo výmenníku by sa mala zohrievať. To či sa zohrieva zatiaľ ale nevieme a zber týchto dát je predmetom tejto kapitoly. Ako bolo v prvej kapitole spomenuté, v oboch zásobníkoch kvapaliny sú umiestnené teplotné senzory. Pridáme ešte jeden senzor, ktorý bude merať teplotu okolia a fotosenzor, ktorý bude merať intenzitu slnečného žiarenia. Dáta z týchto senzorov poputujú do dosky Arduino. Konkrétne do dosky Arduino UNO, ktorá je menšia ako Arduino MEGA 2560 použitej v predchádzajúcej kapitole. Obsahuje aj menej digitálnych aj analógových vstupov, avšak jej možnosti postačujú na toto množstvo senzorov. K nej pripojíme Ethernet Shield, ktorý slúži na pripojenie Arduina k internetu [14].

Teplotné senzory použijeme takého istého typu ako v kapitole 4.6.1 a fotosenzor ten istý ako v kapitole 4.4.1.



Obrázok 5.1 Schéma zapojenia teplotných senzorov a jedného fotosenzoru

K tejto schéme sme už nerobili návrh dosky plošných spojov, ale rovno sme ju zrealizovali na doske s predvŕtanými dierami, pretože úroveň náročnosti jej realizácie je pomerne nízka. Po pripojení senzorov k Arduinu, pripojíme k vyrobenej doske aj teplotné senzory a fotosenzor. Ten umiestnime potom pevne upevníme na časť konštrukcie, ktorá je na Slnku.

5.1 Odosielanie dát na server

K Arduinu je pripojený Ethernet Shield a k nemu sieťový kábel, ktorý umožňuje pripojenie na internet. Na zber našich dát online, sme sa rozhodli využiť službu ThingSpeak [15]. Tá prijíma dáta z rôznych elektronických platforiem, či operačných systémov a ukladá ich do databáz. Po registrovaní na ich webovej stránke je užívateľovi vygenerovaný unikátny kód, ktorý slúži na identifikáciu užívateľa a jeho dát pri vkladaní do databázy. To znamená, že tento kód budeme musieť použiť v zdrojovom kóde pri každom odoslaní dát na server. Podrobné informácie o programovaní Arduina k posielaniu dát a aj samotný skript sme si našťudovali z internetovej stránky [15]. Tento skript sme si upravili k vlastným potrebám, t.j. prispôbobi formu odosielaných dát tak, aby to boli teploty zo senzorov a fotosenzoru.

Výpis 5.1 Odosielanie údajov o teplotách a intenzite Slniečného žiarenia službe ThingSpeak

```
#include <SPI.h>
#include <Ethernet.h>
#include <OneWire.h>
#include <DallasTemperature.h>

#define ONE_WIRE_BUS 2

OneWire oneWire(ONE_WIRE_BUS);

DallasTemperature sensors(&oneWire);
// ThingSpeak nastavenia
char thingSpeakAddress[] = "api.thingspeak.com";
//jedinečný API kód slúžiaci na identifikáciu užívateľa
String writeAPIKey = "XXXXXXXXXXXXXXXXXX";

EthernetClient client;

const int POL_MINUTY= 30 * 1000;

int status;
int failedConnectionAttempCounter;

String dblToString(double input, int decimalPlaces)
{
    if( decimalPlaces != 0)
    {
        String string = String((int)(input*pow(10,decimalPlaces)));

        if(abs(input) < 1)
        {
            if(input > 0)
            {
                string = "0" + string;
            }
        }
    }
}
```

```

        else if(input < 0)
        {
            string = string.substring(0,1) + "0" + string.substring(1);
        }
    }

    return string.substring(0,string.length()-decimalPlaces) + "." +
string.substring(string.length() - decimalPlaces);
}
else
{
    return String((int)input);
}
}

void setup()
{
    Serial.begin(9600);
    sensors.begin();
    connectToInternet();
}

void connectToInternet()
{
    if (client.connected())
    {
        client.stop();
    }

    Serial.println("Pripájam sa na internet...");

    byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

    if (Ethernet.begin(mac) == 0) {
        Serial.println("Zlyhanie konfiguracie pomocou DHCP");
        for(;;){
            ;
        }
    }

    Serial.println(Ethernet.localIP());
}

void loop()
{
    Serial.println("\n");

    sensors.requestTemperatures();
    Serial.print(sensors.getTempCByIndex(0));
    Serial.print(" ");
    Serial.print(sensors.getTempCByIndex(1));
    Serial.print(" ");
    Serial.print(sensors.getTempCByIndex(2));
    Serial.print(" ");
    Serial.println(analog.Read(A0);

```



```

    ReportTemperatureToThingspeak(sensors.getTempCByIndex(0),
sensors.getTempCByIndex(1), sensors.getTempCByIndex(2), analogRead(A0));
}

void ReportTemperatureToThingspeak(float pracovny, float referencny, float
    okolie, int intenzita)
{
    //Tvorba reťazcu, ktorý sa odosiela na server ThingSpeak
    String fields = "1=" + dblToString(pracovny, 2);
    fields += "&2=" + dblToString(referencny, 2);
    fields += "&3=" + dblToString(okolie, 2);
    fields += "&4=" + intenzita;
    Serial.println(fields);

    if (client.connect(thingSpeakAddress, 80))
    {
        // Vytvorenie hlavičky HTTP
        client.print("POST /update HTTP/1.1\n");
        client.print("Host: api.thingspeak.com\n");
        client.print("Connection: close\n");
        client.print("X-THINGSPEAKAPIKEY: "+writeAPIKey+"\n");
        client.print("Content-Type: application/x-www-form-urlencoded\n");
        client.print("Content-Length: ");
        client.print(fields.length());
        client.print("\n\n");
        client.print(fields);

        Serial.print(fields);
        Serial.print("\n");
        Serial.println("Úspešné odoslanie na server www.thingspeak.com");
        delay(POL_MINUTY);
    }
    else
    {
        Serial.println("Pripojenie na server www.thingspeak.com zlyhalo");
        Serial.println();
        failedConnectionAttempCounter++;

        if (failedConnectionAttempCounter > 3 )
        {
            Serial.println("Resetujem pripojenie...");
            connectToInternet();
            failedConnectionAttempCounter = 0;
        }
    }
}
}

```

Hodnoty z každého fotosenzoru sa zobrazujú separátne v inom grafe. Zobrazenie a osi týchto grafov je možné si pohodlne upravovať podľa vlastných potrieb. Po upravení zobrazenia grafov, popísaní ich osí je možné tieto grafy verejne sprístupniť ostatným užívateľom. Na zobrazenie verejných dát nie je potrebná registrácia, len príslušný internetový odkaz. Zatiaľ sme však vytvorili len skúšobné dáta, aby sme si overili funkčnosť skriptu v Arduine a k nemu pripojených senzorov².

² <https://thingspeak.com/channels/36418>

6 Záver

Celý projekt je výnimočný tým, že sa zaoberá kompletným návrhom celého systému. Od konštrukcie samotného zariadenia až po návrh a realizáciu jeho riadenia. Obsiahlosť celého projektu si vyžadovala rozdeliť prácu do viacerých fáz. Popis práce na týchto fázach projektu boli popísané v jednotlivých kapitolách v poradí, ktoré si vyžadovala ich nadväznosť. Nebolo totiž možné navrhovať algoritmus riadenia pokiaľ nebolo čo riadiť. Každá jedna kapitola je výsledkom plánovania, experimentovania, testovania a nie raz aj prerábania celého konceptu.

Podarilo sa však úspešne skonštruovať mechanický systém, ktorý dokáže dvojosofo otáčať s Fresnelovou šošovkou. Konštrukcia je piaty koncept, ktorý staval na chybách jeho predchodcov. Predchádzajúce mechanické systémy ukázali svoje nedostatky a tak museli byť vylepšené, prípadne niektoré veci, ako napríklad návrh pohonu motorčekom a ozubenými kolesami, úplne prerobený, dokonca niekoľko krát, až pokiaľ sa nenašiel spôsob, ktorý fungoval. Napriek tomu, že systém je funkčný, dá sa stále vylepšovať, napríklad použitím na mieru vyrobených súčastí konštrukcie. V tomto prípade boli robené svojpomocne, čo síce znížilo náklady ale odrazilo sa to v estetike konštrukcie.

Sledovanie pozície Slnka nebola ľahká úloha a taktiež si vyžadovala niekoľko predchádzajúcich neúspešných pokusov. Pôvodný koncept pracoval s tromi fotosenzormi v každej sústave a dokonca ďalším zariadením, ktoré malo nájsť najväčšiu intenzitu slnečného žiarenia na horizontálnej osi a údaj o tejto polohe odoslať riadiacemu algoritmu. Namiesto toho postačili dve sústavy fotosenzorov, ktoré spĺňajú účel a dokážu poskytnúť informácie potrebné priamemu natočeniu šošovky na Slnko.

Systém je však dokončený, schopný pohybu a aplikovaným riadiacim algoritmom, ktorý ho riadi požadovaným spôsobom, stanoveným v cieľoch práce. Je tu však aj priestor na ďalšiu prácu s týmto projektom. Vyhodnocovanie nazbieraných dát, experimentovanie s veľkosťami zásobníkov kvapaliny, experimentovanie s prietokom čerpadla, ktoré by sa nenapájalo systémom vypnúť -zapnúť ale spojito, k dosiahnutiu požadovaného prietoku. Projekt by mohol byť použitý ako základ pre zariadenia zohrievajúce vodu pomocou slnečnej energie. Možnosti využitia takéhoto zariadenia sú všade tam, kde svieti Slnko a kde je potrebná teplá voda. Ďalším krokom vo vývoji by mohlo byť premieňanie tepelnej energie obsiahnutej vo vode na elektrickú energiu použitím tepelno-elektrických článkov, akým je napríklad Peltierov článok.

Zoznam použitej literatúry

- [1] <http://sk.wikipedia.org/wiki/Slnko>
- [2] http://sk.wikipedia.org/wiki/Fotovoltický_článok
- [3] PETRÁŠ, P. 2006. Slnčné koncentrátory - všeobecné informácie Dostupné na internete:
<<http://www.zdruzeniepcola.org/view.php?cisloclanku=2006111403>>
- [4] http://sk.wikipedia.org/wiki/Fresnelova_šošovka
- [5] <http://www.greenoptimistic.com/solar-concentrator-diy/#.VV0VXqiBXPn>
- [6] <http://arduino-info.wikispaces.com>
- [7] MONK, Simon. 2010. 30 Arduino Projects for the Evil Genius
- [8] BANZI Massimo. 2011. Getting Started with Arduino / 2. vyd: U.S.A.: Make Books, 2011.
118s. ISBN: 978-1-449-309879
- [9] [http://sk.wikipedia.org/wiki/Tranzistor_\(polovodičová_súčiastka\)](http://sk.wikipedia.org/wiki/Tranzistor_(polovodičová_súčiastka))
- [10] <http://www.learningaboutelectronics.com/Articles/Arduino-keypad-circuit.php>
- [11] <http://playground.arduino.cc/Learning/PhotoResistor>
- [12] <http://www.digikey.com/schemeit/>
- [13] <http://www.hobbytronics.co.uk/ds18b20-arduino>
- [14] <http://www.arduino.cc/en/Main/ArduinoEthernetShield>
- [15] <http://community.thingspeak.com/tutorials/arduino/using-an-arduino-ethernet-shield-to-update-a-thingspeak-channel/>
- [16] <http://community.thingspeak.com/forum/arduino/ds18b20-temp-sensor/>

Prílohy

Príloha A: Konštrukcia systému OptiFres