



Střední průmyslová škola a Vyšší odborná škola, Písek
Karla Čapka 402, Písek

Školní rok: 2011/2012

Kmenový obor: 2643M Elektronika

Studijní obor: 26–43-M/004 Slaboproudá elektrotechnika

Maturitní práce

Řízení sluneční elektrárny

Téma číslo: 37.

Jméno žáka: Adam Pavlát

Třída: **C4.S**

Vedoucí práce: Ing. Miroslav Paul

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

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Písku dne 27. 3. 2012

podpis:.....

PODĚKOVÁNÍ

Děkuji panu Ing. Miroslavu Paulovi za vynaložené úsilí, darovaný čas a podnětné připomínky, které mi během práce poskytoval.

ANOTACE

Cílem práce bylo optimálním způsobem zajistit natáčení fotovoltaické elektrárny zde reprezentované fotovoltaickým panelem. V první části je shrnuto základní rozdělení používaných technologií. Druhá část se zabývá především výpočtem teoretického energetického zisku při použití statického a dynamického uložení fotovoltaického panelu. Část třetí stručně popisuje použité motory, vývoj mnou navržených světlo citlivých prvků a celkové zapojení všech hardwarových prvků. Poslední část nastiňuje problematiku softwarové algoritmizace a řešení použité v projektu včetně uživatelského přístupu vhodného pro kontrolu a ovládání systému na dálku.

ANNOTATION

This project solves optimal use of sunlight photovoltaic panel. The first part describes the technology of photovoltaic panels. In the second part is the calculation of theoretical energy profit when using dynamic (movable) system and compared with static system. The third section describes the used motors, development of light-sensitive sensors and their location on the structure and the total participation of all hardware units with PLC Foxtrot CP-1014. The last part solves issue of software algorithms and solutions used in the project. Solved user access is also suitable for monitoring and controlling the system remotely.

Klíčová slova:

fotovoltaika, fotovoltaický panel, obnovitelné zdroje energie, světlo, programovatelný automat, algoritmizace, Mosaic

Key words:

photovoltaics photovoltaic panels, renewable sources of energy, light, PLC, algorithms, Mosaic

Obsah

1	ÚVOD	6
1.1	Základní charakteristika projektu	6
2	ROZDĚLENÍ FV TECHNOLOGIÍ	6
2.1	Technologie tlustých vrstev	6
2.1.1	<i>Polykrystalické panely</i>	7
2.1.2	<i>Monokrystalické panely</i>	7
2.1.3	<i>Hybridní panely</i>	7
2.2	Technologie tenkých vrstev.....	7
2.3	Kompozitní technologie	8
2.4	Nekřemíkové technologie.....	8
3	SLUNEČNÍ ENERGIE	8
3.1	Sluneční konstanta	9
3.2	Orientační výpočet vyrobené energie statickým a dynamickým FVP	9
3.2.1	<i>Výpočet pro statický FVP</i>	10
3.2.2	<i>Výpočet pro dynamický FVP</i>	11
3.2.3	<i>Porovnání vyrobené energie statických a dynamických FVP</i>	12
4	KONSTRUKCE	12
4.1	Výběr pohonu	13
4.2	Světlo citlivé snímače.....	13
4.3	Hardwarové připojení k PLC Foxtrot CP-1014.....	15
5	SOFTWAREVÁ ALGORITMIZACE	15
5.1	Popis programu.....	16
5.1.1	<i>Řídící část (prgMain)</i>	16
5.1.2	<i>Testovací část (TEST)</i>	18
5.2	Vývojový diagram.....	20
5.3	Uživatelské rozhraní	21
5.3.1	<i>WebMaker</i>	21
5.3.2	<i>PanelMaker</i>	22
6	ZÁVĚR	23

1 Úvod

Jsem studentem SPŠ a VOŠ v Písku a zhruba v polovině předminulého roku jsem se přihlásil (s dalšími 11 žáky naší školy) v rámci programu Leonardo da Vinci na třítydenní odbornou stáž v Německu. Přípravné přednášky na SPŠ a VOŠ v Písku mi poskytly spoustu podnětných znalostí a připravily mě pro, o něco konkrétnější, řešení problematiky v Německu. Stáž se konala v Regenstaufu (poblíž Regensburgu) a během ní se nám na škole, jež nás hostila (Eckert Schulen) dostalo bohatého programu zaměřeného jak na tamější kulturu, tak (a to především) na moderní technologie v oblasti ekologie alias obnovitelné zdroje energie (celým názvem: Obnovitelné zdroje – Regenerative Energien). Začleněny zde byly kategorie: tepelná čerpadla, solární termika a fotovoltaika.

Ve své práci jsem spojil poznatky o obnovitelných zdrojích energií se studovaným oborem - automatizační technikou.

1.1 Základní charakteristika projektu

V tomto projektu jde o zefektivnění výkonu fotovoltaické elektrárny v mém případě realizovanou fotovoltaickým panelem pomocí dvouosé natáčecí konstrukce. Práce se skládá jednak z návrhu, jednak z realizace samotné konstrukce, tří fotocitlivých čidel detekujících optimální směr natočení, dále pak spojení dvou motorů BELIMO LM24-S s PLC Foxtrot 1014 a programu, který má na starosti zpracovávat hodnoty jednotlivých čidel. Dále je pak vyhodnotit a stanovit závěr reprezentovaný úrovněmi poslanými do motorů a tím zajistit natočení a sklopení fotovoltaického panelu.

2 Rozdělení FV technologií

2.1 Technologie tlustých vrstev

Fotovoltaický článek je tvořen polovodičovou P-N diodou. Tyto články se vyrábějí z křemíkových plátek, ať už z monokrystalického nebo polykrystalického křemíku. V současné době se touto technologií vyrábí více než 85 % solárních článků na trhu.

2.1.1 Polykrystalické panely

Základem je křemíková podložka. Si se nechá nejdříve vykrytalizovat a až poté se řeže do článků. Účinnost 12 až 17 %. Obr. 2.1.1

2.1.2 Monokrystalické panely

Základem je křemíková podložka. Krystaly jsou větší než 10 cm a vyrábí se tažením roztaveného křemíku ve formě tyči o průměru až 300 mm. Účinnost 14 až 18 %. Obr. 2.1.2

2.1.3 Hybridní panely

Spojení solárních kolektorů (ohřev vody ve speciálních vakuových kolektorech) se solárními fotovoltaickými panely v jednu konstrukci. Obr. 2.1.3



Obr. 2. 1.1



Obr. 2. 1. 2



Obr. 2. 1. 3

2.2 Technologie tenkých vrstev

Fotovoltaický článek je tvořen nosnou plochou (například sklem, textilií a podobně), na které jsou napařené velmi tenké vrstvy amorfního nebo mikrokrytalického křemíku. Množství materiálu, použitého pro výrobu tenkovrstvého fotovoltaického článku, je nižší, než u tlustých vrstev, takže články jsou levnější.

Nevýhodou současných tenkovrstvých fotovoltaických článků je nižší účinnost a nižší životnost. Výhoda spočívá ve vyšší účinnosti při dopadu tzv. difúzního záření, tj. když sluneční paprsky nedopadají přímo, ale jsou rozptýlené např. po průchodu mrakem.



Obr. 2. 2. 1

2.3 Kompozitní technologie

Kompozitní jsou složeny z jednotlivých vrstev fotovoltaických článků. Jsou schopné efektivně využívat širokou část slunečního spektra. Je to dáno tím, že každá vrstva dokáže využít světlo v určitém rozsahu vlnových délek a to záření, které využít nemůže, propustí do hlubších vrstev, kde je využito. Teoreticky stanovená účinnost těchto nových článků se pohybuje od 50 do 72 % v závislosti na tom, zda jsou vrstvy materiálů 2, 3 nebo je jich 36 (pro hodnotu 72 %).

- Vícevrstvé solární články složené ze tří různých polovodičů, které se podařilo vyrobit v nedávné době, pracovaly se zatím nejvyšší dosaženou účinností kolem 30 %.

2.4 Nekřemíkové technologie

Na rozdíl od předešlých dvou se pro konverzi světla na elektrickou energii nepoužívá tradiční P-N polovodičový přechod. Používají se různé organické sloučeniny, polymery a podobně. Tyto technologie jsou většinou ve stadiu výzkumů.

3 Sluneční energie

Ve Slunci, naší hvězdě, dochází k jevu zvanému termonukleární fúze, kdy se slučují lehké prvky (Vodík – H) v těžší (Helium - He). Při této fúzi dochází k uvolňování velkého množství energie. Vzhledem k tomu, že vyčerpání zásob vodíku Slunce se čítá v řádech miliard let, označuje se využití této energie jako zdroj obnovitelný.

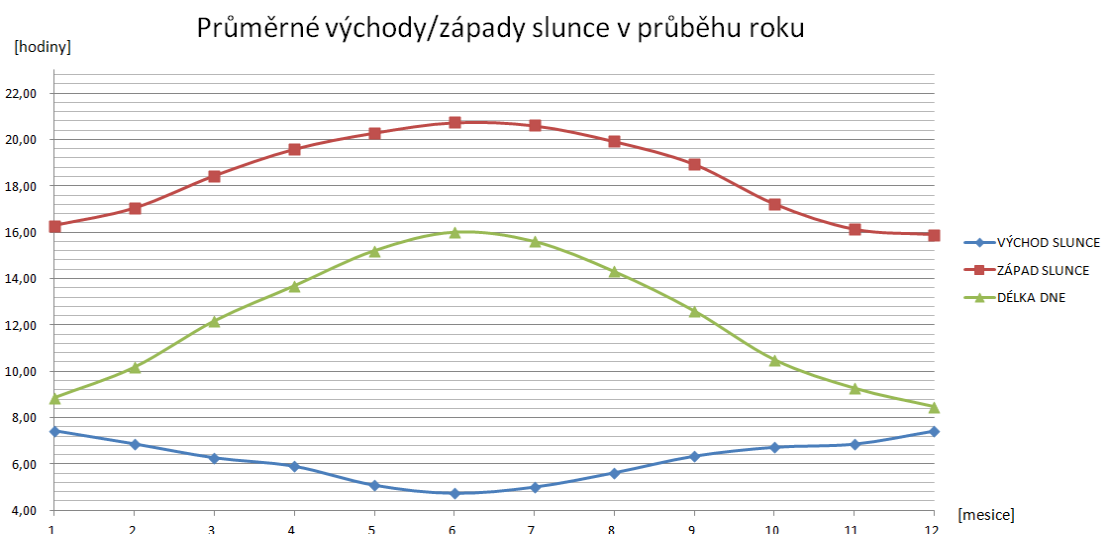
3.1 Sluneční konstanta

Sluneční konstanta je tok sluneční energie procházející plochou 1 m^2 , kolmou na směr paprsků, za 1 s ve střední vzdálenosti Země od Slunce měřený mimo zemskou atmosféru. Konstanta zahrnuje celé spektrum slunečního záření, nejen viditelné světlo. Její hodnota se pro oblast Českých Budějovic zhruba rovná $1\,040 \text{ W/m}^2$, avšak tato hodnota je myšlena k průřezu Země ($\pi \cdot R^2$). Jak planeta rotuje, je tato energie distribuována po celém Zemském povrchu ($4 \cdot \pi \cdot R^2$). $\rightarrow 1\,040 / 4 = 260 \text{ W/m}^2$.

3.2 Orientační výpočet vyrobené energie statickým a dynamickým FVP

Výchozí předpoklady:

- Na povrch Země dopadá v kolmém směru maximální intenzita záření I o přibližné hodnotě $1\,040 \text{ W/m}^2$
- S_{12} je jednotková plocha FVP 1 m^2 kolmá ke směru dopadajícího záření v pravé poledne.
- Délka dne v průběhu roku je přibližně 12,23 hod. Vyplývá to z grafu vykresleného z údajů na internetových stránkách **Českého hydrometeorologického ústavu**.



Obr. 3. 2. 1

	hodin vyšlé Slunce	
LEDEN	8,85	hod
ÚNOR	10,17	hod
BŘEZEN	12,17	hod
DUBEN	13,68	hod
KVĚTEN	15,20	hod
ČERVEN	16,00	hod
ČERVENEC	15,60	hod
SRPEN	14,30	hod
ZÁŘÍ	12,60	hod
ŘÍJEN	10,50	hod
LISTOPAD	9,27	hod
PROSINEC	8,47	hod

$$\frac{1}{12} \sum_{i=1}^{12} x_i = \frac{146,8}{12} = 12,23 \text{ hod}$$

Tabulka 3. 2. 1

Pro výpočet energie platí:

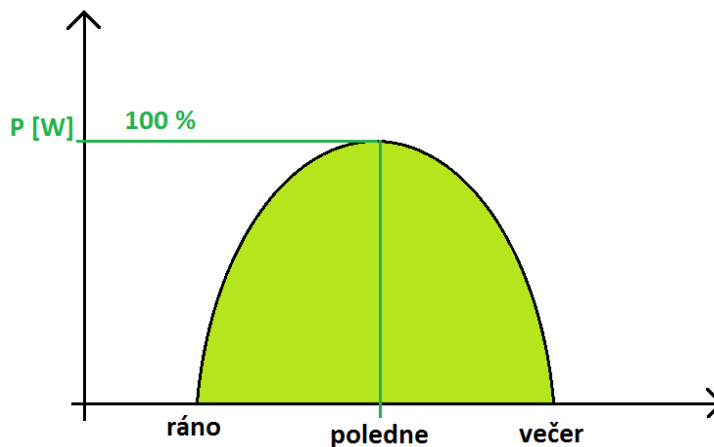
$$W = \int_{t_1}^{t_2} P dt = \int_{t_1}^{t_2} IS dt$$

Kde P je výkon dopadajícího záření (W)

S je průmět plochy S_{12} FVP při šikmém dopadu paprsků (m^2)

I je intenzita slunečního záření při kolmém dopadu (W/m^2)

3.2.1 Výpočet pro statický FVP



Obr. 3.2.1.1 Časový průběh výkonu během dne u pevně instalovaného FVP

Průběh dne: $\varphi \in \langle -\pi/2; +\pi/2 \rangle$

$$\omega = \frac{2\pi}{T} = \frac{2\pi}{24 \cdot 60 \cdot 60} \cong 7,27 \cdot 10^{-5} \text{ s}^{-1}$$

Ve výpočtech nebudeme uvažovat vliv atmosféry. Pro výpočet energie, která dopadne na jednotkovou plochu $S_{12} = 1 \text{ m}^2$.

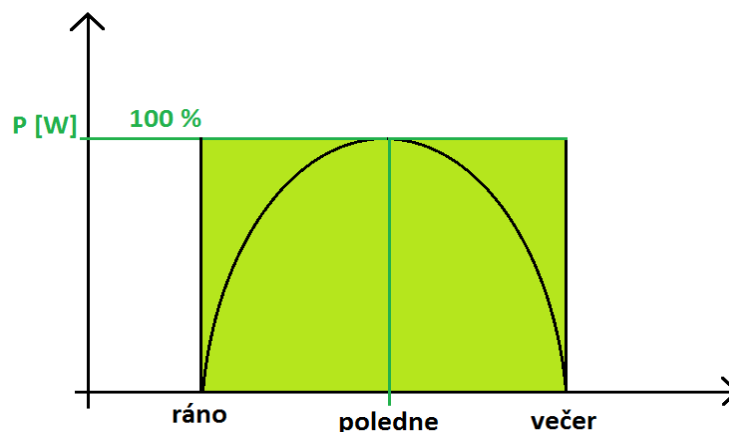
$$W = \int_{t_1}^{t_2} IS dt = \int_{t_1}^{t_2} IS_{12} \cos(\omega) dt = I \cdot S_{12} \int_{t_1}^{t_2} \cos(\omega) dt = IS_{12} \left[\frac{\sin(\omega)}{\omega} \right]_{t_1}^{t_2} = \frac{2 \cdot I \cdot S_{12}}{\omega}$$

$$W = \frac{2 \cdot I \cdot S_{12}}{\omega} = \frac{2 \cdot 1040 \cdot 1}{7,27 \cdot 10^{-5}} \cong 28,61 \cdot 10^6 \text{ Ws}$$

$$28,61 \cdot 10^6 \text{ Ws} \rightarrow \frac{28,61 \cdot 10^6 \text{ Ws}}{1000 \cdot 60 \cdot 60} = 7,947 \text{ kWh}$$

3.2.2 Výpočet pro dynamický FVP

Tento výpočet bude velmi jednoduchý, protože, jak vyplývá z obr 3. 2. 2. 1, bude FVP neustále v úhlu 90° k dopadajícím paprskům.



Obr. 3. 2. 2. 1 Časový průběh výkonu během dne u dvouosé konstrukce

$$W = P \cdot t = I \cdot S_{12} \cdot t = 1040 \cdot 1 \cdot 44028 \cong 4,579 \cdot 10^7 \text{ Ws}$$

$$4,579 \cdot 10^7 \text{ Ws} \rightarrow \frac{4,579 \cdot 10^7 \text{ Ws}}{1000 \cdot 60 \cdot 60} \cong 12,719 \text{ kWh}$$

3.2.3 Porovnání vyrobené energie statických a dynamických FVP

$$\frac{W_d}{W_s} = \frac{12,719 \text{ kWh}}{7,947 \text{ kWh}} \cong 1,6$$

Z výsledku je patrné, že zvolíme-li systém dynamický, tedy systém, který dokáže sledovat Slunce na obloze ve dvou osách, zvýší se výroba elektrické energie o celých 60 %. Je to ale pouze teoretická hodnota, která nepočítá s atmosférickými vlivy. Složitost přesného matematického popsání neodpovídá jejímu přínosu. Za zmínku stojí například doba východu a západu slunce, kdy sluneční záření musí procházet tlustší vrstvou atmosféry, a tak ztrácí energii, což by mělo za následek zaoblení horních okrajů v obr. 3. 2. 2. 1.

4 Konstrukce

Na obr. 4.1 je konstrukce umožňující dvouosé natáčení FVP, kterou používám při realizaci modelu FVE. Železné tělo umožní konstrukci pevné uložení na zemi a zamezuje tak nežádoucím pohybům. Rotační schopnost ve vodorovné a svislé ose zajišťují pevná průmyslová ložiska. Jejich vůle v hřídeli bylo nutno zajistit přidáním tzv. šneků (zde pro železný materiál zvoleny klasické hlavy šroubů a ne esteticky přijatelnější bezhlavé). Na obr. 4.2 je kompletní sestava s FVP.



Obr. 4.1



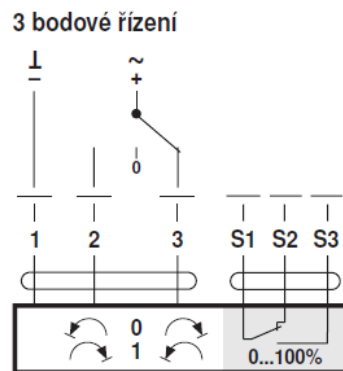
Obr. 4.2

4.1 Výběr pohonu

Pro natáčení v obou osách jsem zvolil servomotory od firmy Belimo z důvodu kompaktních rozměrů a provozní spolehlivosti. Servomotory BELIMO LM24A-S jsou sice původem předurčeny pro přestavování vzduchotechnických klapek ve vzduchotechnických a klimatizačních zařízeních budov, nicméně s přehledem zvládnou i tuto aplikaci. Krouticí moment 5 Nm nijak neomezuje pohyblivost méj konstrukce.

Napájecí napětí 24 V je stejné jako u použitého PLC Foxtrot a jelikož světlo citlivá čidla jsou vybavena stabilizátorem napětí, který upravuje velikost napětí pro potřebu čidel s elektronickými prvky, mohu použít pouze jeden napájecí zdroj.

Motor používám v tzv. třibodovém režimu, kdy, jak reprezentuje obr. 4.1.1, je jeden směr dán spojenými svorkami 1-2 a směr opačný 3-2. Tento způsob řízení zajišťuje svými výstupy programovatelný automat (PLC) Foxtrot 1014.



obr. 4.1.1

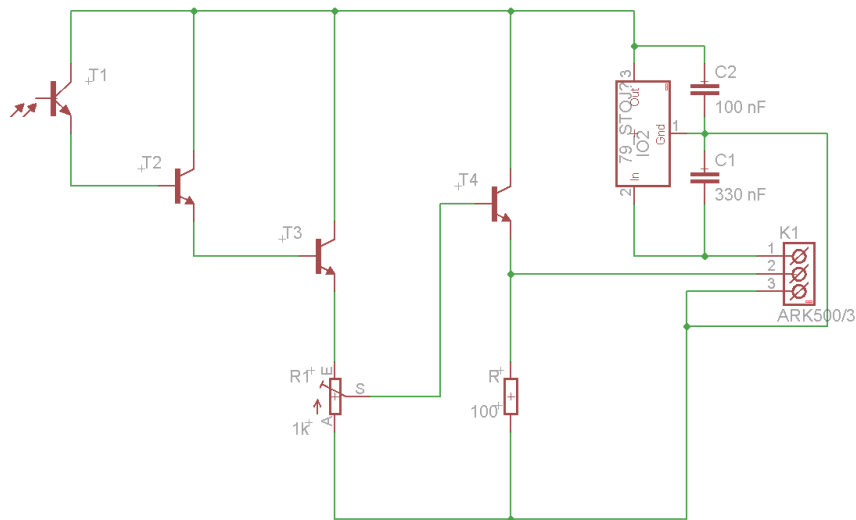
3 bodové řízení klapkových pohonů BELIMO LM24A-S

4.2 Světlo citlivé snímače

Možností realizace elektrického obvodu, jehož výstup by byl funkcí míry osvětlení, bylo mnoho. Snažil jsem se zvolit co nejjednodušší zapojení. Obvod je napájen napájecím napětím 5 V. O stálost referenčního napětí se stará stabilizátor napětí. Elektrické schéma zpracované v programu Eagle 5.6 je znázorněno na obr. 4.2.1

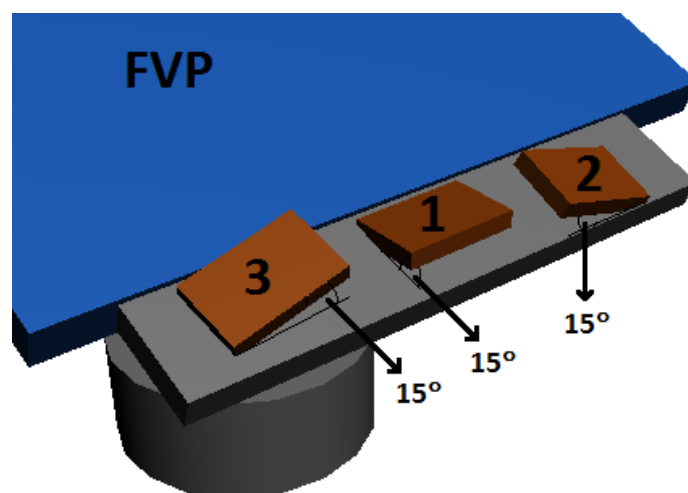
Obvod je jednoduchý, tranzistory T2 a T3 tvoří Darlingtonovův zesilovač, ten zesiluje napětí vzniklé na fototranzistoru T1. Takto upravený elektrický signál následně otevírá tranzistor T4. Stabilizátor 7805 jsem použil proto, abych mohl napájet čidla zdrojem pro

pohony a PLC. Kondenzátory C1 a C2 slouží k omezení špiček napětí. Čidla jsou tři, víc není potřeba, výrobní cena zhruba 50 Kč/kus.



Obr. 4.2.1

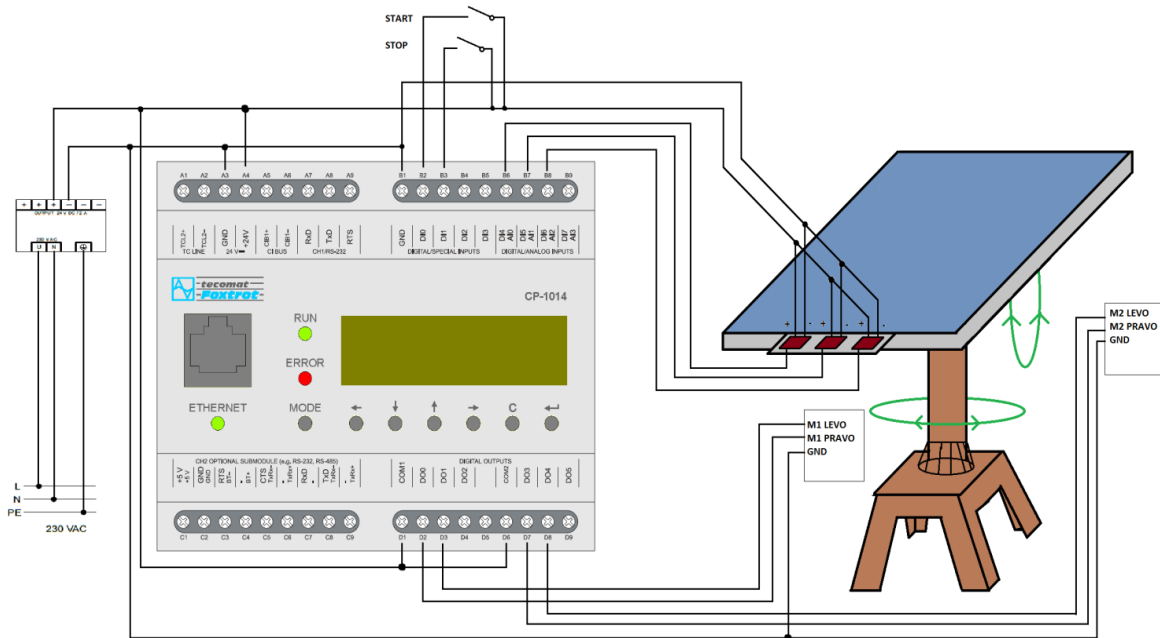
Správné natočení čidel je zásadní pro celý systém. Směrnost fototranzistorů by měla teoreticky zajistit úzké a vhodné pásmo příjmu, ale praxe ukázala, že to není dostatečné řešení. Proto byly instalovány bužírky okolo fototranzistorů, které směrnost upravily.



Obr. 4.2.2 – reálné rozmístění čidel na konstrukci panelu

4.3 Hardwarové připojení k PLC Foxtrot CP-1014

Všechny hardwarové části (PLC Foxtrot CP-1014, dva motory BELIMO LM24(S) a tři světlocitlivá čidla) jsou napájeny z jednoho zdroje. Měnič 230 AC – 24 DC zajišťuje napájení. Zdroj nemá vypínač. Vypínač je realizován jističem, který je předřazen zdroji.



Obr. 4.3.1

5 Softwarová algoritmizace

S programováním v prostředí Mosaic jsem se seznámil ve čtvrtém ročníku na naší škole. Jde o prostředí pro programování PLC (u mě konkrétně PLC Foxtrot CP-1014). Mosaic nabízí čtyři možnosti zápisu logického kódu (dva grafické – LD, FBD, dva textové – ST, IL), všechny jsou deklarovány normou IEC 61 131-3. Já jsem zvolil programování v jazyce ST, protože je mi nejbližší zápisem kódu.

Algoritmus jsem rozdělil do dvou programů (prgMain, TEST). Učinil jsem tak pro přehlednost a rozdělení dílčích činností. Program prgMain se stará o časování spouštění a spánkový režim a program TEST o filtrování hodnot z čidel, jejich vyhodnocování a spouštění konkrétního motoru v konkrétním směru.

Uživatelské rozhraní je řešeno ovládáním displeje zabudovaného přímo v PLC (popis v 5.3.2). Navíc je k dispozici webové rozhraní pro přístup na dálku. Pro přehled tvoří webové rozhraní pouze jednu komplexní stránku. Celkový popis webových stránek viz. 5.3.1

5.1 Popis programu

Po spuštění programu se čeká na stisknutí tlačítka START (prozatím je, stejně jako STOP, pouze programově řešené), po kladné odezvě přejde program ke kontrole (za jistých podmínek). Spánkový režim jsem nejdříve ovládal vybíráním příkazem CASE s měsíční přesností. To znamená, že jsem si vypočetl průměrné doby východu a západu slunce s přesností na měsíce a následně jsem podle aktuálního systémového času vyhodnocoval, zdali je tento čas v intervalu, či nikoliv. Bylo to nepotřebně dlouhé řešení, které nakonec vystřídala kratší a komplexnější knihovna AstroLib. Knihovna AstroLib je standardně dodávána jako součást programovacího prostředí Mosaic. Knihovna obsahuje funkce umožňující počítat základní údaje o poloze slunce (zde je použito pouze výpočtu pro východ a západ slunce = funkce SunTime). Konstanty Zenith_Official až Zenith_Astronomical určují, pro jaký soumrak má funkce SunTime() vypočítat časy východu resp. západu slunce. Zvolil jsem Zenith_Civil (Občanský soumrak). Ráno občanský soumrak začíná okamžikem, kdy se střed slunečního kotouče nachází 6° pod horizontem, a končí východem Slunce; večer začíná západem Slunce a končí okamžikem, kdy se střed slunečního kotouče nachází 6° pod horizontem.

5.1.1 Řídící část (prgMain)

Nyní potřebujeme k dispozici informaci o tom, zdali je den (*PODMINKADNE.Q1* = true) či nikoli (false). Pokud den není, povolí program přístup k algoritmu výpočtu doby spánku pomocí funkčního bloku z knihovny AstroLib funkcí SunTime. Spánkový režim je navíc ošetřen proti výpadkům proudu a následnému znovuoživení a přepočtení doby spánku pro aktuální čas v daných souřadnicích, tedy poloze panelu. *PROBUZENIV* je údaj použitý pro displej PLC a webové rozhraní.

```
VAR
DNES: DATE;
VYCHOD: TIME;
ZAPAD: TIME;
LONGITUDE: LREAL :=14.0851; //ŠÍŘKA 14.0851° VÝCHODNĚ
LATITUDE: LREAL := 49.1834; //VÝŠKA 49.1834° SEVERNĚ (SOUŘADNICE PÍSKU)
PODMINKADNE: RS;
CAS_REAL: TIME;
```



```

CELY_DEN: TIME := 86400000; //24*60*60*1000 (ms)
END_VAR

//-----SPÁNKOVÝ REŽIM S VYUŽITÍM ASTROLIB-----
DNES := GETDATE();
VYCHOD := SUNTIME(DIRECTION := SUNRISE, ZENITH := ZENITH_OFFICIAL, LONGITUDE :=
LONGITUDE, LATITUDE := LATITUDE, UTCOFFSET := 1., ACTDATE := DNES);

ZAPAD := SUNTIME(DIRECTION := SUNSET, ZENITH := ZENITH_OFFICIAL, LONGITUDE :=
LONGITUDE, LATITUDE := LATITUDE, UTCOFFSET := 1., ACTDATE := DNES);

CAS_REAL := GETTIME();

PODMINKADNE(S:= (CAS_REAL > VYCHOD) AND (CAS_REAL < ZAPAD),
R1:= (CAS_REAL < VYCHOD) OR (CAS_REAL > ZAPAD));

IF PODMINKADNE.Q1 = FALSE
THEN SLEEP:=VYCHOD + (CELY_DEN - ZAPAD);
    SPANEK(IN:=NOT(PODMINKADNE.Q1), PT:=SLEEP);
    IF CAS_REAL > ZAPAD
    THEN DIFERENCE_CAS:= CAS_REAL - ZAPAD;
    ELSE DIFERENCE_CAS:= (CELY_DEN - ZAPAD) + CAS_REAL;
    END_IF;
    PROBUZENIV:=SLEEP - (DIFERENCE_CAS);
ELSE //-----HLAVNÍ ŘÍZENÍ CYKLŮ-----

```

Cyklické spouštění kontrol od stisknutí tlačítka *START* až po tmu *STOP* = 1 (nebo přepnutí do režimu spánku) je ovládáno soustavou časovačů (*CYKL*, *PROCYKL*, *DOBACYKLU*). RS klopný obvod (*UZJECAS*) ovládá svým nastavením či vynulováním spuštění kontrol a pauzu. Proměnná *AKTIV_CONTROL* byla zavedena kvůli webovému rozhraní (viz.5.3.1), bylo vhodnější z hlediska uživatelské přívětivosti volit dobu aktivní kontroly, než dobu, která by měla do konce cyklu zbývat (*CYPLUS*).

```

//-----HLAVNÍ ŘÍZENÍ CYKLŮ-----
UZJECAS(S:=(PODMINKADNE.Q1) AND (CYKL.Q) OR NOT(DOBAODCYKLU.Q),
      R1:=NOT(START_RS.Q1) OR (PROCYKL.Q) OR (DOBAODCYKLU.Q) OR SPANEK.Q);

CYPLUS:=POJEDE-AKTIV_CONTROL;      // POJEDE = CELKOVÁ DOBA CYKLU;
                                     // AKTIV_CONTROL = DOBA KONTROLY

CYKL(IN:=(START), PT:=POJEDE);
PROCYKL(IN:=(UZJECAS.Q1), PT:=POJEDE);
DOBAODCYKLU(IN:=PROCYKL.Q, PT:=CYPLUS);

```

5.1.2 Testovací část (TEST)

Testovací část se skládá z několika částí. První je filtrace hodnot. Při prvním zapojení čidel narazilo prvotní řešení na problém, a to nežádoucí jevy na fotosnímačích vlivem rozptylu záření při použití systému v exteriérovém měřítku. Hodnoty čtené z čidel nebyly v tomto prostředí stálé (kolísaly kolem správné hodnoty), a tak bylo zapotřebí zavést tuto filtraci.

Jednou za tři sekundy (čítání systémového registru %S13.2, tj. sekunda, ve funkčním bloku CTU (*REFRESHCTU*) se spustí cyklický zápis z čidel (všech tří) do jednorozměrného programového pole s rozsahem padesáti hodnot typu *real PRUMER1[index]*. Čítač CTU se musí resetovat, aby mohl čítat znovu od nuly do předvolby *REFRESHCTU.PV*. Pro nulování čítače CTU se dá použít stavu dosažení čítače předvolby. V tomto okamžiku se objeví na výstupu Q log. 1, která, když se přivede na vstup RESET čítače, jej vynuluje. PLC může v použitém řešení indexovat maximálně 50 hodnot, vyplývá to z doby cyklu programu. Výsledky rozdílu čidel vyplývají z rozmístění čidel na konstrukci panelu viz. obr 5.2.2. *ROZDILX* je rozdílem vyfiltrovaných (zprůměrovaných) hodnot čidel 2 a 3. *ROZDILY* je rozdílem hodnot čidel 1 a 2.

```

REFRESHCTU(CU:=SEKUNDY, R:=RESET_PRO_CTU.Q, PV:=3);
RESET_PRO_CTU(IN:=REFRESHCTU.Q, PT:=T#500mS);    //DOBA PRO NAPLNĚNÍ POLÍ

IF REFRESHCTU.Q = TRUE THEN M1P:=FALSE; M2P:=FALSE; //ZKUŠENOSTI -> NUTNO
                                     //BĚHEM TESTU VYPLNOUT

```

```
M1L:=FALSE;M2L:=FALSE; //MOTORY (ALGORITMUS
//ROZHODUJÍCÍ O SMĚRU MÁ TOTIŽ K DISPOZICI NEPROFILTROVANÁ DATA)
```

```
index:=0;
```

```
FOR index:=0 TO 50 DO //PLC NAPLNÍ 50 HODNOT DO POLE
  PRUMER1[index]:=SIGNAL1;
  PRUMER2[index]:=SIGNAL2;
  PRUMER3[index]:=SIGNAL3;
END_FOR;
```

```
index1:=0; CELEK1:=0.; CELEK2:=0.; CELEK3:=0.;
```

```
FOR index1:=0 TO 50 DO //SUMA PRO NÁSLEDUJÍCÍ PRŮMĚR
  CELEK1:=CELEK1+PRUMER1[index1];
  CELEK2:=CELEK2+PRUMER2[index1];
  CELEK3:=CELEK3+PRUMER3[index1];
END_FOR;
```

```
ROZDILX := (CELEK3/50. - CELEK2/50.); //PRŮMĚR A ROZDÍL ČIDEL TŘI A DVA
ROZDILY := (CELEK1/50. - CELEK2/50.); //PRŮMĚR A RODÍL ČIDEL JEDNA A DVA
```

```
END_IF;
```

ROZDILX a ROZDILY jsou zvoleny za směrodatné a následně se díky nim vyhodnotí, který motor má být na jakou stranu spuštěn. Záměrně je zvolena chronologie posuvu, tj. první posuv bude do stran (kolem svislé osy, natočení X) a druhý směr nahoru či dolů (kolem horizontální osy, natočení Y), a to z důvodu přesnosti. Vezmeme-li v úvahu natočení čidel dva a tři, je zřejmé, že pokud by se nejprve představoval směr Y, došlo by k nepřesnosti. Na čidlech dva a tři by nebyly stejné hodnoty a tudíž by Slunce nebylo tam, kde podle výpočtu vychází, ale o tento rozdíl čidel dva a tři mimo. Touto chronologií se program vyhnul problematice výpočtu křivky této závislosti, a to jen za cenu nesouběžné činnosti motorů.

```
//-----JE VHODNÉ PŘESTAVIT PANEL?-----
```

```
IF ((ABS(ROZDILX) > 0.5)) OR ((ABS(ROZDILY) > 0.5)) //SIGNALIZACE POTŘEBY PŘESUNU
  THEN PROSIM_PRESTAV := TRUE;
  ELSE PROSIM_PRESTAV := FALSE;
END_IF;
```

```

//-----KAŽDÝ MOTOR MÁ K DISPOZICI STEJNÝ ČAS-----
  JAKYM(IN:=UZJECAS.Q1, PT:=POJEDE);
  IF JAKYM.ET < (POJEDE/2) AND UZJECAS.Q1 = TRUE THEN

//-----SVISLÁ OSA-----
  IF ABS(ROZDILX) > 0.01 THEN           //0.01 JE TOLERANCE, POD JEJÍ HRANICÍ
                                          //SE POVAŽUJE
                                          //SMĚR PANELU ZA IDEÁLNÍ
    IF ROZDILX < 0. THEN M1L := TRUE; M1P := FALSE;
      ELSE M1P := TRUE; M1L := FALSE;

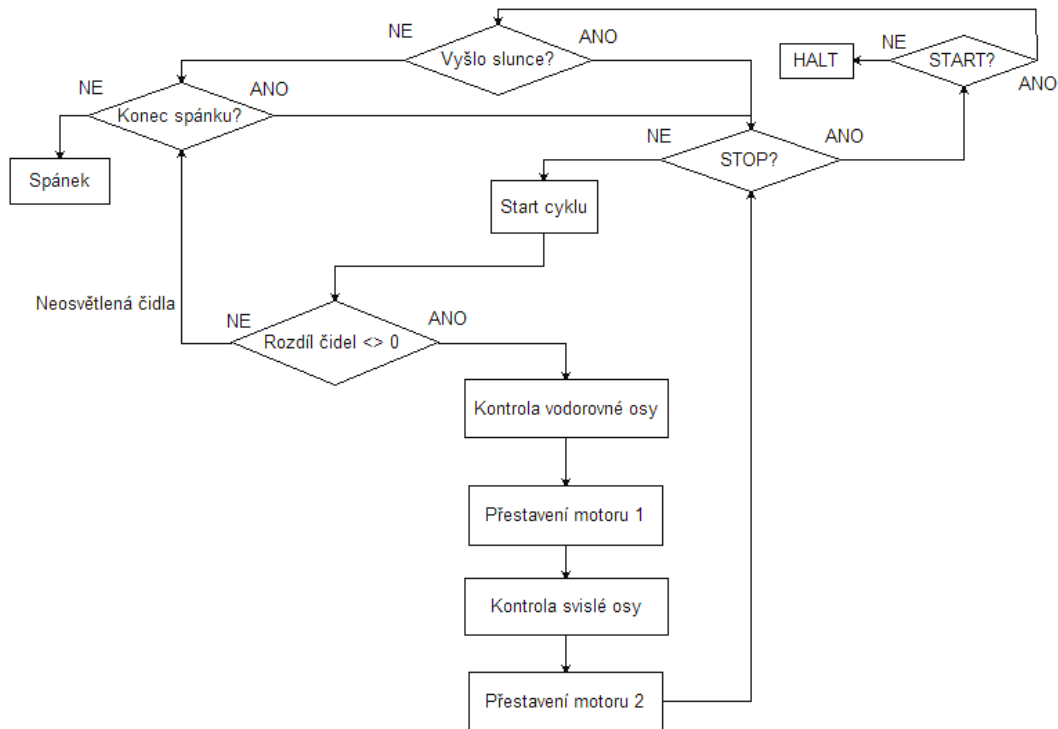
    END_IF;
  END_IF;

ELSE
//-----VODOROVNÁ OSA-----
M1L := FALSE; M1P := FALSE;
  IF ABS(ROZDILY) > 0.01 THEN
    IF ROZDILY > 0. THEN M2L := TRUE;
      M2P := FALSE;
    ELSE M2P := TRUE;
      M2L := FALSE;

    END_IF; END_IF;

```

5.2 Vývojový diagram



Obr. 5.2.1 – blokové schéma algoritmu

5.3 Uživatelské rozhraní

Personál ovládající tento systém nepotřebuje znát všechny proměnné, podmínky a funkční bloky, ale pouze údaje, které z použité logiky vyplývají.

5.3.1 WebMaker

Programovatelný automat Foxtrot CP-1014 má možnost stát se zároveň webovým serverem a to za podmínky vložené paměťové karty. Součástí programového prostředí Mosaic je i Webmaker, který umožňuje právě takové weby vytvářet, přímo je propojit s programem a umožnit tak vzdálený přístup (zjištění stavu systému či provedení změn tohoto stavu).

Pro pohodlnost používá mnou navržený web pouze jednu stranu, na které je vše potřebné pro vzdálený provoz navrženého systému. Držel jsem se mojí prvotní strohé koncepce s černým pozadím a snažil se maximalizovat užitek webového rozhraní. V příloze je přiložen náhled tohoto rozhraní.

Popis částí:

1. Tlačítko START a STOP, významově stejně provázané jako stejnojmenná hardwarová tlačítka, která nejsou instalována.
2. Přítomný čas ukazuje reálný systémový čas. Je vidět neustále, v jakémkoli stavu systému. Na rozdíl od toho je viditelnost „Probudím se: čas do probuzení) v (čas východu slunce)“ podmíněno nocí.
3. „Doba celkového cyklu“ ovládá proměnnou POJEDE, která udává čas (primárně 60 sekund), ze kterého se pak odečte čas (vždy menší než POJEDE, z rychlosti motorů vychází minimální hodnota 45 s – kompletní přestavení motorů z jedné strany na druhou, změřeno) potřebný pro kontrolu. V této sekci je dále zobrazen údaj o čase do konce nebo začátku kontroly (vhodná aktivace).
4. Stiskem „Stiskem povol ruční zobrazení“ se nastaví priorita takovým způsobem, že nedochází k cyklickým testům, ale systém je podmíněn pouze ručním ovládním, tedy stiskem vhodné šipky. Smysl je orientovaný z pohledu panelu. Směr pohybu panelu je zobrazen vždy jednou šipkou, tento ukazatel je viditelný i v době cyklického testování.

5. V pátém dílu jsou zobrazeny hodnoty po filtraci, tedy ty, z kterých se vychází při vyhodnocení závěru pro motory. Pro každou osu jsou zde dva sloupce ovládané proměnnou, pokud je výsledek záporný, zvětšuje se levý sloupec a naopak.
6. Z pátého dílu (tedy z rozdílu čidel) se vyhodnotí (jeli rozdíl $>$ tolerance) hláška „Systém je vhodné přestavit.“. To je užitečné když neběží testovací činnost. Stiskem STOP a následně Start dojde k novému přestavení.
7. Vertikální výška sloupců reprezentuje konkrétní a aktuální hodnotu úrovně získávaného z čidel.

5.3.2 *PanelMaker*

V PLC Foxtrot CP-1014 je zabudovaný displej velikosti 20 x 4 znaků. Využil jsem ho pro zobrazení základních údajů o stavu systému. Stav displeje jsou rozříděny do tří skupin, a to STARTOVACÍ, TESTOVACÍ a NETESTOVACÍ. Ve všech skupinách se může obsluha PLC manuálně (šipkami) pohybovat v několika podskupinách.

STARTOVACÍ bude logicky zobrazena při nečinnosti, tedy před stiskem tlačítka START a po stisknutí tlačítka STOP. Mohlo by se zdát, že zde splývají skupiny STARTOVACÍ a NETESTOVACÍ, nicméně bylo úrovnově mnohem jednodušší rozdělení. Ve STARTOVACÍ skupině je možné zobrazit tři podskupiny, a to aktuální hodnoty na čidlech, rozdíl profiltrovaných hodnot a možnost stisknutí tlačítka START programově.

TESTOVACÍ skupina zobrazuje primárně to, který motor se kam pohybuje a kolik času zbývá do konce kontroly. Je zde opět možné přepnout šipkou na PLC do podkategorií o aktuálním stavu čidel a o rozdílu v obou směrech.

NETESTOVACÍ skupina je zobrazena ve dvou případech, buď pokud je spuštěný automatický provoz (cyklické kontroly), a netestuje se (zobrazí, za jakou dobu se spustí kontrola znovu, a má možnost spustit tuto kontrolu ručně) nebo v případě spuštění režimu spánku (ukáže čas do východu slunce).

6 Závěr

Pro vypracování tohoto projektu bylo nutné využít několik procesů od návrhu elektrických obvodů (např. také v jejich simulaci př. v programu Multisim), leptání plošných spojů až po následné pájení součástek atd. Především jsem se zaměřil na logiku celého systému a nad programem strávil mnoho hodin, čímž jsem se snažil najít nejkolektivnější a nejsympatičtější řešení, které by bylo elegantní a prosté. I přes to jsem se úspěšně nevyhnul všem problémům, např. stav systému při působení silného větru na konstrukci nebo důsledky tíhy sněhu. Zatím mechanismus ochrany vůči těmto vlivům leží v řešení přes překonání krouticích momentů motoru. Například pokud bude více sněhu v dolní části panelu, bude tlačít jeho tíha směrem k zemi. Tato tíha překoná krouticí moment motoru horizontální osy a panel se překloupí do svislé polohy (vypadne západka motoru a zpět se nevrátí). Problém s větrem není bez větrného čidla tak jednoduché vyřešit. Překonání momentu je neúčinné v neideálním úhlu.

Bylo vhodné navrhnout a zrealizovat pro systém webové rozhraní, které přes internet umožňuje interaktivně sledovat a ovládat systém. Pro komfort při bezprostředním ovládní uživatelem v místě umístění konstrukce pomáhá text přímo na displeji PLC.

- Kalkulace použité konstrukce:

PLC – 11 300 Kč, BELIMO motory – $2 \cdot 3276$, FVP – 7 000 Kč. Celkové náklady se rovnají částce přibližně 25 000 Kč. Podle statistik dopadá na území ČR přibližně 1 000 kWh/m² ozáření sluneční energie za rok. Plocha panelu je přibližně ½ m². Tím pádem lze z FVP dostat maximálně 500 kWh/rok ($500 \cdot \text{účinnost tj. } 0,16 = 80 \text{ kWh/rok}$). Připojíme naši konstrukci v roce 2012 do rozvodné sítě (vychází se z aktuálních údajů Energetického regulačního úřadu). Výkup elektřiny = 6,16 Kč/kWh plus zelený bonus 5,08 Kč/kWh. Celkem tedy 11,24 Kč/kWh. Vytvořená konstrukce vyrobí maximálně 80 kWh/rok, tj. $80 \cdot 11,24 =$ přibližně 900 Kč/rok. Systém se tedy při zanedbání spotřeby energie zaplatí za $(25\,000 / 900)$ let, tedy za 27,8 let.

- Kdybychom například počítali s deseti panely, výsledná doba by se znatelně zkrátila, a to konkrétně na desetinu:

$$\frac{25000}{11,24 \cdot 10 \cdot 80} \cong \frac{25000}{9000} = 2,78 \text{ let}$$

Zde nejsou odečteny náklady např. na silnější motory, materiál konstrukce apod.

Návrh se tedy nehodí pro reálné použití, ale slouží pouze jako demonstrace dvouosého natáčecího systému. Z výsledků vyplývá, že pro zkrácení doby, za kterou by se součásti a použité prvky zaplatili, by bylo nutné použít výkonnější (větší) panel.

Použitá literatura:

[1] – Haselhuhn, Ralf: Budovy jako zdroj proudu

[2] – internet: <http://www.czrea.org/>, <http://www.eru.cz/>

PŘÍLOHA:

Webové rozhraní:

The screenshot shows a web interface for a testing system. It features several control panels and status indicators:

- 1**: START and STOP buttons.
- 2**: Input fields for 'Přítomný čas: 0.' and 'Probudím se za: 0.' V, and '0.'.
- 3**: 'Doba celkového cyklu: 0.' (Primárně 60.) and 'Z toho čas kontroly: 0.' (< celkový cyklus). Below it, 'Do konce testu zbývá: 0.' and 'Další test za: 0.'.
- 4**: 'Stiskem povol ruční přestavení' and 'Směr pohybu panelu.' with four directional arrows (up, down, left, right).
- 5**: '(Testuje se) Horizontální rozdíly čidel [V]: 0.' and '(Testuje se) Vertikální rozdíly čidel [V]: 0.' with two horizontal bar charts showing sensor differences.
- 6**: 'Systém je vhodné přestavit.' (System is suitable for adjustment).
- 7**: A bar chart showing 'Číslo 1', 'Číslo 2', and 'Číslo 3' with voltage levels of 2 V, 1 V, and 0 V respectively.

Výpis programu:

Globální proměnné:

```
VAR_GLOBAL

MESIC AT %S11 : USINT; //SYSTÉMOVÝ REGISTR MESICE
HODINY AT %S8 : USINT; //SYSTÉMOVÝ REGISTR HODINY
MINUTY AT %S7 : USINT; //SYSTÉMOVÝ REGISTR MINUTY
SEKUNDY AT %S13.2 : BOOL

START_RS : RS; //TOTAL VYP NEBO ZAP
PODMINKADNE : RS; //DEN NEBO NOC
UZJECAS : RS; //SPLNENI PODMINEK PRO KONTROLU

POJEDE : TIME :=T#60S; //CELÝ ČAS CYKLU
CYPLUS : TIME; //POJEDE - CYPLUS = KONTROLA AKTIVNÍ (AKTIV_CONTROL)
AKTIV_CONTROL: TIME :=T#45S
DOBAODCYKLU : TOF; //ČASOVAČ UDRŽENÍ KONTROLY NA DOBU POJEDE
PROCYKL : TON; //POMOCNÝ ČASOVAČ PRO DOBACYKLU

DALSIZA : TIME; //DOBA CYKLU - CAS PRO KONTOLU (PRO DISPLAY)
KONECZA : TIME; //JAK DLOUHO SE JEŠTĚ BUDE KONTROLOVAT (PRO DISPLAY)
CASPRODISPLAY : TOF; //POMOCNÝ ČASOVAČ (PRO DISPLAY)
SLEEP : TIME; //DOBA SPÁNKU PRO TON ČASOVAČ SPANEK (PRO KAŽDÝ MĚSÍC JINÝ)
PROBUZENIV : TIME; //ZA JAK DLOUHO SE PROBUDÍ (PRO DISPLAY)
DIFERENCE_CAS: TIME;

HALTDISP : BOOL; //POMOCNÉ
HALTDOCAS : BOOL; //PROMĚNNÉ
JESTOP : BOOL; //PRO
DOCASTOP : BOOL; //DISPLAY

ROZDILX : REAL; //ROZDIL CIDEL V OSE X (VODOROVNÉ)
ROZDILY : REAL; //ROZDIL CIDEL V OSE Y (SVISLÉ)

MANUAL : BOOL;
POHYB : USINT;

END_VAR
```

Program PrgMain:

```
PROGRAM prgMain
VAR_INPUT
END_VAR
VAR_OUTPUT
END_VAR
VAR
CYKL : TOF;
SPANEK : TON;
//-----PROMĚNNÉ PRO WWW-----
ZAPOR_X : REAL;
ZAPOR_Y : REAL;
PSEUDO_1 : REAL;
PSEUDO_2 : REAL;
```

```

PSEUDO_3 : REAL;
ZAPORX : BOOL;
ZAPORY : BOOL;

//-----SPÁNKOVÝ REŽIM-----
DNES: DATE;           //DNEŠNÍ DATUM
LONGITUDE :LREAL :=14.0851; //DÉLKA
LATITUDE :LREAL :=49.1834; //ŠÍŘKA
VYCHOD :TIME;        //ČAS
ZAPAD :TIME;
CAS_REAL :TIME;
CELY_DEN : TIME :=86400000;
VIDIM_OKENKO :BOOL;
END_VAR
VAR_TEMP
END_VAR

START_RS(S:=START, R1:=STOP);

//-----SPÁNKOVÝ REŽIM S VYUŽITÍM ASTROLIB-----
DNES := GETDATE();
VYCHOD := SUNTIME(DIRECTION := SUNRISE, ZENITH := ZENITH_CIVIL, LONGITUDE := LONGITUDE,
LATITUDE := LATITUDE, UTCOFFSET := 1., ACTDATE := DNES);

ZAPAD := SUNTIME(DIRECTION := SUNSET, ZENITH := ZENITH_CIVIL, LONGITUDE := LONGITUDE,
LATITUDE := LATITUDE, UTCOFFSET := 1., ACTDATE := DNES);

CAS_REAL:= GETTIME();

//-----PODMÍNKADNE.Q1 JE JEDNOU Z PODMÍNEK PROVOZU-----
PODMINKADNE(S:= (CAS_REAL > VYCHOD) AND (CAS_REAL < ZAPAD),
R1:= (CAS_REAL < VYCHOD) OR (CAS_REAL > ZAPAD));

IF PODMINKADNE.Q1 = FALSE THEN

SLEEP:=VYCHOD + (CELY_DEN - ZAPAD);
SPANEK(IN:=NOT(PODMINKADNE.Q1), PT:=SLEEP);

IF CAS_REAL > ZAPAD //PO NEBO PŘED PŮLNOCÍ
THEN DIFERENCE_CAS:= CAS_REAL - ZAPAD; //PRO PŘÍPAD, KDY DOJDE K VÝPADKU
PROUDU PŘED
ELSE DIFERENCE_CAS:= (CELY_DEN - ZAPAD) + CAS_REAL; //PŮLNOCÍ
//PRO PŘÍPAD, KDY DOJDE
//K VÝPADKU PROUDU
//O PŮLNOCI

END_IF;

PROBUZENIV:=SLEEP - (DIFERENCE_CAS);

ELSE

//-----HLAVNÍ ŘÍZENÍ CYKLŮ-----
UZJECAS(S:=(PODMINKADNE.Q1) AND (CYKL.Q) OR NOT(DOBAODCYKLU.Q),
R1:=NOT(START_RS.Q1) OR (PROCYKL.Q) OR (DOBAODCYKLU.Q) OR SPANEK.Q);

CYPLUS:=POJEDE-AKTIV_CONTROL;
CYKL(IN:=(START), PT:=POJEDE);

```

```

PROCYKL(IN:=(UZJECAS.Q1), PT:=POJEDE);
DOBAODCYKLU(IN:=PROCYKL.Q, PT:=CYPLUS);

//-----AKTIVACE DISPLEJE PLC-----
JESTOP := FALSE;
IF (STOP = TRUE)
  THEN JESTOP := TRUE;
  ELSE JESTOP := FALSE;
END_IF;

DOCASTOP := FALSE;
IF (START_RS.Q1 = TRUE) AND (UZJECAS.Q1 = FALSE) AND (JESTOP = FALSE)
  THEN DOCASTOP := TRUE;
  ELSE DOCASTOP := FALSE;
END_IF;

//-----PŘEPOČTY PRO DISPLAY-----
KONECZA:=REAL_TO_TIME(TIME_TO_REAL(POJEDE)-TIME_TO_REAL(PROCYKL.ET));
DALSIZA:=REAL_TO_TIME(TIME_TO_REAL(CYPLUS) - TIME_TO_REAL(CASPRODISPLEY.ET));
CASPRODISPLEY(IN:=PROCYKL.Q, PT:=POJEDE);

IF START_RS.Q1 = FALSE THEN DALSIZA:=REAL_TO_TIME(0.0);
END_IF;

END_IF;

//-----OVLÁDÁNÍ WEBMAKERU-----
IF UZJECAS.Q1=FALSE AND PODMINKADNE.Q1=TRUE
  THEN VIDIM_OKENKO:=TRUE;
  ELSE VIDIM_OKENKO:=FALSE;
END_IF;

ZAPOR_X:=ABS(ROZDILX)*60.;
ZAPOR_Y:=ABS(ROZDILY)*60.;

IF ROZDILX < 0. THEN ZAPORX:=TRUE;
ELSE ZAPORX:=FALSE;
END_IF;

IF ROZDILY < 0. THEN ZAPORY:=TRUE;
ELSE ZAPORY:=FALSE;
END_IF;

PSEUDO_1:=SIGNAL1*60.; PSEUDO_2:=SIGNAL2*60.; PSEUDO_3:=SIGNAL3*60.;

END_PROGRAM

```

PROGRAM TEST:

```

TYPE
PRUMER1_1      :ARRAY[0..50] OF REAL;
PRUMER2_2      :ARRAY[0..50] OF REAL;
PRUMER3_3      :ARRAY[0..50] OF REAL;
END_TYPE

```

```

PROGRAM TEST
VAR_INPUT
END_VAR
VAR_OUTPUT
END_VAR
VAR

//-----FILTROVÁNÍ HODNOT-----
PRUMER1: PRUMER1_1;
PRUMER2: PRUMER2_2;
PRUMER3: PRUMER3_3;
CELEK1: REAL;
CELEK2: REAL;
CELEK3: REAL;
index : INT;
index1 : INT;
//-----VYBÍRÁNÍ OSY-----
JAKYM : TON;
PROSIM_PRESTAV : BOOL;
REFRESHCTU : CTU;
RESET_PRO_CTU : TON;
//-----RUČNÍ OVLÁDÁNÍ-----
MAN1 : RS;
MAN2 : RS;
MAN3 : RS;
MAN4 : RS;
PSEUDO1 : BOOL;
PSEUDO2 : BOOL;
PSEUDO3 : BOOL;
PSEUDO4 : BOOL;
HORIZONTAL : BOOL;
VERTIKAL : BOOL;
END_VAR
VAR_TEMP
END_VAR
//-----ROZDÍLY ČIDEL-----
REFRESHCTU(CU:=SEKUNDY, R:=RESET_PRO_CTU.Q, PV:=3);
RESET_PRO_CTU(IN:=REFRESHCTU.Q, PT:=T#500mS);
IF REFRESHCTU.Q = TRUE THEN M1P:=FALSE;M2P:=FALSE;M1L:=FALSE;M2L:=FALSE;
index:=0;

FOR index:=0 TO 50 DO
PRUMER1[index]:=SIGNAL1;
PRUMER2[index]:=SIGNAL2;
PRUMER3[index]:=SIGNAL3;
END_FOR;

index1:=0; CELEK1:=0.; CELEK2:=0.; CELEK3:=0.;

FOR index1:=0 TO 50 DO
CELEK1:=CELEK1+PRUMER1[index1];
CELEK2:=CELEK2+PRUMER2[index1];
CELEK3:=CELEK3+PRUMER3[index1];
END_FOR;

ROZDILX := (CELEK3/50. - CELEK2/50.);
ROZDILY := (CELEK1/50. - CELEK2/50.);

```

```

END_IF;

IF ((ABS(ROZDILX) > 0.5)) OR ((ABS(ROZDILY) > 0.5)) THEN PROSIM_PRESTAV := TRUE;
      ELSE PROSIM_PRESTAV := FALSE;
END_IF;

IF MANUAL = FALSE THEN

//-----KAŽDÝ MOTOR MÁ K DISPOZICI STEJNÝ ČAS-----
  JAKYM(IN:=UZJECAS.Q1, PT:=POJEDE);
  IF JAKYM.ET < (POJEDE/2) AND UZJECAS.Q1 = TRUE THEN

//-----SVISLÁ OSA-----
  IF ABS(ROZDILX) > 0.01 THEN
    IF ROZDILX < 0. THEN M1L := TRUE;
      M1P := FALSE;
    ELSE M1P := TRUE;
      M1L := FALSE;
    END_IF;
  END_IF;

ELSE
//-----VODOROVNÁ OSA-----
M1L := FALSE; M1P := FALSE;

  IF ABS(ROZDILY) > 0.01 THEN
    IF ROZDILY > 0. THEN M2L := TRUE;
      M2P := FALSE;      //M2L = TRUE = NAHORU
    ELSE M2P := TRUE;
      M2L := FALSE;
    END_IF;
  END_IF;
END_IF;

//-----MOTORY MOHOU BĚŽET POUZE PŘI KONTROLE-----
IF (DOCASTOP = TRUE) OR (START_RS.Q1 = FALSE) THEN
  M1L := FALSE;
  M1P := FALSE;
  M2L := FALSE;
  M2P := FALSE;
END_IF;

//-----PRO WEB MAKER-----
ELSE
MAN1(S:=PSEUDO1, R1:=NOT(PSEUDO1), Q1 => M1L);      //RUČNÍ OVLÁDÁNÍ
MAN2(S:=PSEUDO2, R1:=NOT(PSEUDO2), Q1 => M1P);
MAN3(S:=PSEUDO3, R1:=NOT(PSEUDO3), Q1 => M2L);
MAN4(S:=PSEUDO4, R1:=NOT(PSEUDO4), Q1 => M2P);
END_IF;

IF (M1L=FALSE AND M1P=FALSE AND M2L=FALSE AND M2P=FALSE) THEN POHYB:=4; //VÝBĚR ZE
SMĚRŮ ŠIPEK
END_IF;
IF M1L = TRUE THEN POHYB:=0;
END_IF;
IF M1P = TRUE THEN POHYB:=1;

```

```
END_IF;
IF M2L = TRUE THEN POHYB:=2;
END_IF;
IF M2P = TRUE THEN POHYB:=3;
END_IF;

IF UZJECAS.Q1 = TRUE THEN
IF KONECZA < POJEDE/2
THEN HORIZONTAL:=FALSE; VERTIKAL:=TRUE;
ELSE HORIZONTAL:=TRUE; VERTIKAL:=FALSE;
END_IF;
ELSE HORIZONTAL:=FALSE; VERTIKAL:=FALSE;
END_IF;

IF ((ABS(ROZDILX) > 0.5)) OR ((ABS(ROZDILY) > 0.5))
THEN PROSIM_PRESTAV := TRUE;
ELSE PROSIM_PRESTAV := FALSE;
END_IF;

END_PROGRAM
```