

Jihočeská univerzita v Českých Budějovicích  
Přírodovědecká fakulta



# Ovládání solární elektrárny pomocí mikropočítače Raspberry Pi

Bakalářská práce

Martin Stropek

Vedoucí práce: Ing. Jan Fesl, Ph.D.

České Budějovice 2020

Jihočeská univerza v Českých Budějovicích

Přírodovědecká fakulta

## ZADÁVACÍ PROTOKOL BAKALÁŘSKÉ PRÁCE

**Student:** Martin Stropek  
(jméno, příjmení, tituly)

**Obor - zaměření studia:** Aplikovaná Informatika

**Katedra:** Ústav aplikované informatiky

**Školitel:** Ing. Jan Fesl, Ph.D.  
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, email)

**Garant z Přírodovědecké fakulty:** .....  
(jméno, příjmení, tituly, katedra - jen v případě externího školitele)

**Školitel - specialista, konzultant:**  
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, email)

**Téma bakalářské práce:**  
Ovládání solární elektrárny pomocí mikropočítače Raspberry Pi

### Popis práce:

Solární elektrárny jsou dnes velmi oblíbeným ekologickým zdrojem pro získávání elektrické energie. Pro zajištění optimální efektivity se běžně používají stavitelná řešení využívající náklonu a natočení jednotlivých solárních panelů. Systémy pro řízení solárních panelů různých výrobců zpravidla fungují na principu světelných senzorů.

Student se v této práci seznámí s možnostmi ovládání pozicování solárních panelů prostřednictvím elektromotorů a mikropočítače RaspBerry Pi na základě matematicky určených souřadnic. Za účelem dosažení praxí využitelných výsledků dojde k vytvoření jednoduchého modelu solární elektrárny a implementaci řídicího systému a dále aplikace pro monitorování jejího stavu.

### Cíle práce:

V rešeršní části se student ponoří do problematiky řízení náklonu solárních panelů pomocí elektromotorů včetně vazby na astrometrické údaje nutné pro dosažení optimálních výsledků.

V návrhové části student vytvoří model pro řízení náklonu jednoho miniaturního solárního panelu, realizuje jeho mechanickou výrobu a následné zprovoznění. Detailně budou popsána všechna schémata zapojení v soustažnosti k připojení mikropočítače.

V implementační části dojde ke zprovoznění navrženého modelu a implementaci systému, který bude umožňovat automatické řízení náklonu solárního panelu a aplikace pro zobrazení stavových hodnot přes webové rozhraní.

**Základní doporučená literatura:**

[1] M. Boxwell, Solar electricity handbook: A Simple, Practical Guide to Solar Energy - Designing and Installing Photovoltaic Solar Electric Systems, Greenstream Publishing, 2010

[2] UPTON, Eben a Gareth HALFACREE. Raspberry Pi: uživatelská příručka. Brno: Computer Press, 2013

Financování práce: .....  
Vedoucí práce: Ing. Jan Fesl, Ph.D. podpis: *JF*  
U externích vedoucích fakultní garant práce: podpis: .....  
Garant oboru bak. studia (nepožaduje se u zaměření, příprava na mag. stud. biologie): podpis: *JA*  
..... podpis: .....  
Vedoucí katedry: Rudolf Vohnout podpis: .....  
Odborný konzultant (chemická část): podpis: .....  
Případný souhlas vedoucí ústavu AV: ..... podpis: .....

V Českých Budějovicích dne: ....10.12.2019.....

Převzal/a dne: *27.1.2020*.....

podpis: *sh*.....

## **Bibliografické údaje:**

Stropek, M., 2020: Ovládání solární elektrárny pomocí mikropočítače Raspberry Pi. [Control of solar power station using Raspberry Pi. Bc. Thesis, in Czech.] – 48 p., Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic

## **Anotace:**

Cílem práce je vytvoření funkčního řídicího systému pro ovládání otočných fotovoltaických panelů pomocí jednodeskového počítače Raspberry Pi. Hlavním úkolem řídicího systému je natáčení panelu směrem ke slunci na základě času. Nedílnou součástí práce je vytvoření zmenšeného modelu otočného solárního panelu, na kterém bude odzkoušena a demonstrována funkčnost systému. Dále práce obsahuje jednoduchou webovou aplikaci pro monitorování výroby elektrické energie.

## **Klíčová slova:**

Raspberry Pi, solární energie, fotovoltaické panely, sledování slunce, Python, Raspbian

## **Annotation:**

The aim of this bachelor thesis is to create functional control system with a single-board microcomputer Raspberry Pi. The main task of the control system is to rotate the panel and track the sun based on time. An integral part of the thesis is to construct small revolving solar panel to test and demonstrate functionality of system. Thesis also include simple web application for monitoring the production of electricity.

## **Keywords:**

Raspberry Pi, solar energy, photovoltaic panels, solar tracking, Python, Raspbian

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne .....

Podpis autora. ....

## Poděkování

Rád bych poděkoval panu Ing. Janu Feslovi, Ph.D. za vedení mé bakalářské práce, cenné rady a vstřícnost při konzultacích.

# Obsah

Úvod.....	3
1 Cíle .....	4
2 Teoretická část.....	5
2.1 Solární Panely .....	5
2.1.1 Historie Fotovoltaiky .....	5
2.1.2 Princip Fotovoltaiky .....	5
2.2 Sledování slunce.....	6
2.2.1 Druhy natáčení panelů .....	6
2.2.2 Astronomické vazby .....	7
2.3 Raspberry Pi .....	11
2.3.1 Seznámení .....	11
2.3.2 Specifikace.....	11
2.3.3 Trocha historie .....	11
3 Metodika.....	12
3.1 Řídící jednotka .....	12
3.2 Model .....	12
3.3 Webová aplikace .....	12
4 Výběr hardwaru .....	13
4.1 Řídící jednotka .....	13
4.1.1 Raspberry Pi .....	13
4.1.2 Relé .....	14
4.1.3 Zdroje.....	14
4.1.4 Modul reálného času .....	14
4.1.5 Krabice.....	14
4.2 Model .....	14
4.2.1 Motory .....	14
4.2.2 Solární panel .....	14
4.2.3 Anemometr .....	14
4.2.4 Koncové spínače .....	14
4.2.5 Imitace elektroměru .....	15
5 Výběr Softwaru .....	15
5.1 Řídící systém.....	15
5.1.1 Operační systém .....	15
5.1.2 Programovací jazyk .....	15
5.1.3 Knihovny .....	15
5.2 Webová aplikace .....	16

5.2.1	Programovací jazyky .....	16
5.2.2	Frameworky/Knihovny .....	16
6	Praktická Část.....	16
6.1	Řídící systém.....	16
6.1.1	Hardwarová implementace .....	16
6.1.2	Softwarová implementace.....	21
6.2	Model .....	31
6.3	Webová aplikace .....	33
7	Testování .....	36
8	Závěr.....	38
	Použitá literatura .....	39
	Seznam obrázků.....	42
	Seznam příloh .....	42
	Přílohy.....	43
	Příloha A Schéma zapojení .....	43
	Příloha B Součástky modelu .....	44
	Příloha C Fotografie modelu a řídicí jednotky.....	46
	Příloha D Zdrojové kódy.....	48



## Úvod

Fotovoltaické panely jsou dnes velmi oblíbeným ekologickým zdrojem pro získávání elektrické energie, nejen na našich polích můžeme vidět nemalý počet elektráren s těmito panely.

A protože poloha slunce vzhledem ke konkrétnímu místu na naší planetě se neustále mění, je pro zajištění optimální efektivity velmi vhodné panely neustále směřovat tak, aby na ně dopadl co největší počet slunečních paprsků, což přináší značný nárůst při výrobě energie oproti fixním systémům.

Dnes se na trhu vyskytuje již řada firem nabízejících stavitelná řešení využívající náklonu a natočení jednotlivých solárních panelů. Řízení solárních panelů je u komerčních systémů realizováno prostřednictvím technologií různých výrobců. Tyto systémy zpravidla fungují na principu světelných senzorů, díky kterým je možné rozpoznat, kam dopadá nejvíce světla a podle toho nastavit směr panelů.

Alternativním řešením, kterým se zabývá tato práce, je řídicí systém, který ovládá pozicování solárních panelů podle matematicky určených souřadnic na základě času. Souřadnice jsou vypočteny dle daných astronomických algoritmů podle místa (zeměpisné délky a šířky), data a denní doby. Takto odpadá nutnost použití senzorů, které se mohou vlivem okolních podmínek snadno zašpinit či poškodit a přestat korektně pracovat. Výsledkem by mělo být robustní, bezúdržbové a jednoduché řešení.

# 1 Cíle

Hlavním předmětem práce je návrh a implementace řídicí jednotky pro pozicování solárních panelů prostřednictvím elektromotorů, jejímž základem bude mikropočítač Raspberry Pi. Důraz bude kladen zejména na spolehlivost, jednoduchost a nízkou finanční náročnost projektu. Za účelem dosažení praxí využitelných výsledků dojde k vytvoření jednoduchého miniaturního modelu solární elektrárny, na kterém bude odzkoušena a demonstrována funkčnost řídicí jednotky. Posledním úkolem bude implementace aplikace pro monitorování jejího stavu.

V rešeršní části je nutné ponořit se do problematiky řízení solárních panelů, analyzovat současná řešení a jejich klíčové funkcionality. Dále je vhodné prozkoumat vazby na astrometrické údaje nutné pro dosažení optimálních výsledků zvoleného řešení.

V návrhové části bude vytvořen koncept řídicí jednotky a model pro řízení jednoho miniaturního otočného fotovoltaického panelu, který bude následně mechanicky zkonstruován. Detailně budou popsána všechna schémata zapojení.

V implementační části dojde ke zprovoznění navrženého modelu a hardwarové i softwarové implementaci řídicího systému, který bude umožňovat automatické řízení náklonu solárního panelu a naprogramování aplikace pro zobrazení stavových hodnot přes webové rozhraní

## **Kroky k naplnění hlavního cíle:**

1. Rešerše problematiky řízení náklonu fotovoltaických panelů
2. Návrh a hardwarová i softwarová implementace řídicí jednotky
3. Návrh a mechanická realizace modelu otočného panelu
4. Implementace aplikace pro sledování výroby energie přes webové rozhraní

## 2 Teoretická část

### 2.1 Solární Panely

Solární panely se dají rozdělit na **tepelné kolektory**, které přeměňují solární energii na tepelnou a využívají se pro ohřev vody nebo vzduchu a **fotovoltaické panely**, které přeměňují energii získanou ze slunce na elektřinu. [1]

Dále budou v této práci zmiňovány pouze fotovoltaické solární panely.

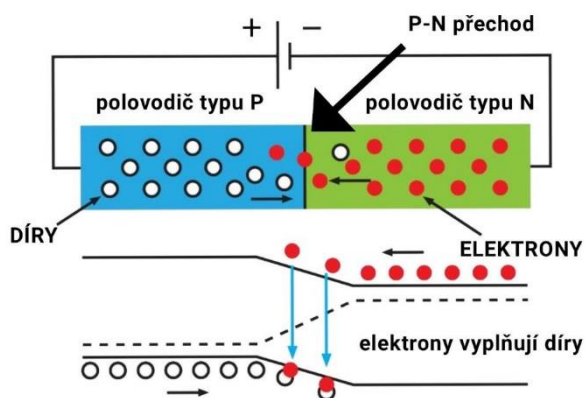
#### 2.1.1 Historie Fotovoltaiky

Fotovoltaický jev objevil v roce 1839 francouzský fyzik Edmond Becquerel. [1] Roku 1921 získal Albert Einstein Nobelovu cenu v oblasti fyziky za jeho vysvětlení. [2]

Experimentální fotovoltaické články z devatenáctého století dosahovaly účinnosti okolo 1 %. První fotovoltaický článek byl použit pro výrobu elektřiny v roce 1954 v Bellových laboratořích (jeho účinnost činila kolem 6 %). Již roku 1958 se fotovoltaické články začaly používat u kosmických družic, kde je fotovoltaika používána dodnes. Nejstarší pozemní instalace byly vybudovány v 80. letech minulého století. [3] Dnešní fotovoltaické panely dosahují účinnosti až 24 %. [1]

#### 2.1.2 Princip Fotovoltaiky

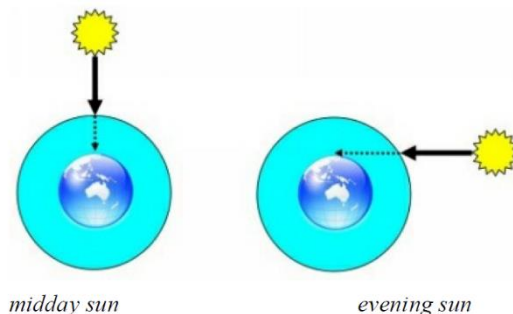
Fotovoltaický jev funguje na principu diody. Máme polovodič (zpravidla křemík) s dvěma vrstvami materiálů typu P/N, přičemž jedna vrstva má přebytek elektronů a druhá vrstva přebytek volných děr. Při vystavení slunečnímu záření do tohoto spojení narážejí fotony a způsobují, že elektrony putují k opačné elektrodě a tím vytvářejí elektrický náboj. [1], [4]



Obrázek 1: Fotovoltaický jev [4]

## 2.2 Sledování slunce

Nejenže slunce vychází na východě a během dne se postupně přesouvá k západu, ale i jeho výška nad obzorem (úhel elevace) se neustále mění zejména podle denní doby, ale závisí i na ročním období (respektive dnu v roce) [1]



Obrázek 3: Pozice slunce [1]



Obrázek 2: Pozice slunce 2 [1]

Z výše uvedeného vyplývá, že je nanejvýš vhodné panely natáčet tak, aby na ně dopadl co největší počet slunečních paprsků. Podle výrobců přinášejí otočné systémy nárůst výkonu až o 30 % oproti fixním systémům.[5], [6] Podle výzkumu Berkeley Lab je 70 % panelů instalovaných v roce 2018 schopno sledovat slunce. [7]

### 2.2.1 Druhy natáčení panelů

Systémy pro natáčení panelů směrem ke slunci se mohou lišit v několika aspektech.

Z hlediska konstrukce je nejzákladnější dělení podle toho, zda se panely otáčejí kolem jedné osy, která může být vertikální, horizontální nebo i různě nakloněna nebo kolem dvou os. Panely, které se otáčejí kolem dvou os mohou získat maximum solární energie, přesto nejsou tak rozšířeny kvůli složitější konstrukci a větším pořizovacím nákladům. [7],[8]

Dále rozlišujeme, jakým způsobem je natáčení řízeno. Existují zcela pasivní systémy, které nejčastěji využívají tepelné roztažnosti kapalin a gravitační síly. Aktivní systémy používají

k otáčení panelů zpravidla stejnosměrné elektromotory. U aktivních systémů se běžně využívají čidla, díky kterým je možné zjistit skutečnou polohu slunce, respektive úhel pro nejlepší dopad slunečních paprsků (metod, jak tento úhel vyhodnotit, je mnoho). Setkáme se ale i se systémy řízenými na základě času. Pomocí astronomických algoritmů lze vypočítat, kde by se mělo slunce v daný moment nacházet a podle toho panely správně natočit, Každá z možností má své kladné i stinné stránky. [8]

### 2.2.2 Astronomické vazby

Protože součástí práce je řídicí systém natáčející panel ve dvou osách závislý na čase, je třeba vědět, kde se kdy slunce nachází neboli určit azimut a elevaci. K tomuto účelu budou využity matematické vzorce dostupné z oficiálních stránek organizace **NOAA** ve formě xls souboru, [9] kde tvůrci pro výpočet využívají astronomických algoritmů, které publikoval **Jean Meeus** ve své knize *Astronomical algorithms*. [10]

#### NOAA

NOAA neboli National Oceanic and Atmospheric Administration (v překladu: Národní úřad pro oceán a atmosféru) je vědecká vládní organizace spadající pod Ministerstvo obchodu Spojených států amerických a fungující pod tímto názvem již od roku 1970. Tato organizace se zabývá předpovědí počasí, varováním před nebezpečným počasím, hlídáním oceánů i oblohy a mnoha dalšími věcmi. Vědci z NOAA mají k dispozici nejnovější vybavení k provádění moderního výzkumu, aby mohli všem poskytovat co možná nejpřesnější informace. [11]

#### Jean Meeus

Jean Meeus je belgický meteorolog a amatérský astronom narozený roku 1928. Vystudoval matematiku a napsal mnoho publikací týkajících se astronomie. Roku 1986 získal ocenění *Amateur Achievement Award of the Astronomical Society of the Pacific* a je po něm pojmenován asteroid. [12]

## Vzorce pro výpočet azimutu a elevace

Výše zmíněné vzorce ze stránek NOAA přepsány do lépe čitelné podoby:

$\Xi$  = Latitude (+ to N) -zeměpisná šířka

$\Phi$  = Longitude (+ to E) -zeměpisná délka

$\omega$  = Time Zone (+ to E) -časové pásmo

$d$  = Date -datum

$\tau$  = Time (past local midnight)- počet hodin, které uběhly od půlnoci místního času

$v$  = Julian Day -juliánský datum

$\sigma$  = Julian Century =  $\frac{v-245154536525}{36525}$  - juliánské století

$\rho$  = Geom Mean Long Sun (deg) =  
 $(280.46646 + \sigma(36000.76983 + 0.0003032\sigma)) \bmod 360$

$\xi$  = Geom Mean Anom Sun (deg) =  $357.52911 + \sigma(35999.05029 - 0.0001537\sigma)$

$\mu$  = Eccent Earth Orbit =  $0.016708634 - \sigma(0.000042037 + 0.0000001267\sigma)$

$\lambda$  = Sun Eq of Ctr =  $\sin(\xi^{rad}) (1.914602 - \sigma(0.004817 + 0.000014\sigma)) +$   
 $\sin(2\xi^{rad}) (0.019993 - 0.000101\sigma) + 0.000289 \sin(3\xi^{rad})$

$\kappa$  = Sun True Long (deg) =  $\rho + \lambda$

$\eta$  = Sun App Long (deg) =  $\kappa - 0.00569 - 0.00478 \sin((125.04 - 1934.136\sigma)^{rad})$

$\zeta$  = Mean Obliq Ecliptic (deg) =  $23 + \frac{26 + \frac{21.448 - \sigma(46.815 + \sigma(0.00059 - \sigma 0.001813))}{60}}{60}$

$\epsilon$  = Obliq Corr (deg) =  $\zeta + 0.00256 \cos((125.04 - 1934.136\sigma)^{rad})$

$\delta$  = Sun Declin (deg) =  $(\arcsin(\sin(\epsilon^{rad}) \sin(\eta^{rad})))^{\circ}$

$y$  = var  $y = ((\frac{\epsilon}{2})^{rad})^2$

$\Gamma$  = Eq of Time (minutes) =  $4(y \sin(2\rho^{rad}) - 2\mu \sin(\xi^{rad}) + 4\mu y \sin(\xi^{rad}) \cos(2\rho^{rad}) -$   
 $0.5y^2 \sin(4\rho^{rad}) - 1.25\mu^2 \sin(2\xi^{rad}))$

$\gamma$  = True Solar Time (min) =  $(1440\tau + \Gamma + 4\Phi - 60\omega) \bmod 1440$

$$\beta = \text{Hour Angle (deg)} = \text{if} \left( \frac{\gamma}{4} < 0 \right) \left\{ \frac{\gamma}{4} + 180 \right\} \text{else} \left\{ \frac{\gamma}{4} - 180 \right\}$$

$$\Omega = \text{Solar Zenith Angle (deg)} = \left( \arccos(\sin(\varepsilon^{\text{rad}})\sin(\delta^{\text{rad}}) + \cos(\varepsilon^{\text{rad}})\cos(\delta^{\text{rad}})\cos(\beta^{\text{rad}})) \right)^{\circ}$$

$$\text{Solar Elevation Angle (deg)} = 90 - \Omega$$

$$\text{Solar Azimuth Angle (deg cw from N)} =$$

$$\text{if}(\beta > 0) \left\{ \left( \arccos \left( \frac{\sin(\varepsilon^{\text{rad}})\cos(\Omega^{\text{rad}}) - \sin(\delta^{\text{rad}})}{\cos(\varepsilon^{\text{rad}})\sin(\Omega^{\text{rad}})} \right)^{\circ} + 180 \right) \text{mod } 360 \right\}$$

$$\text{else} \left\{ \left( 540 - \arccos \left( \frac{\sin(\varepsilon^{\text{rad}})\cos(\Omega^{\text{rad}}) - \sin(\delta^{\text{rad}})}{\cos(\varepsilon^{\text{rad}})\sin(\Omega^{\text{rad}})} \right)^{\circ} \right) \text{mod } 360 \right\}$$

### Poznámky:

Eccent Earth Orbit( $\mu$ ), Obliq Corr( $\epsilon$ ) a var-y se mění minimálně, proto je můžeme pro naše účely chápat jako konstanty s hodnotami: 0.0167; 23.44 a 0.043.

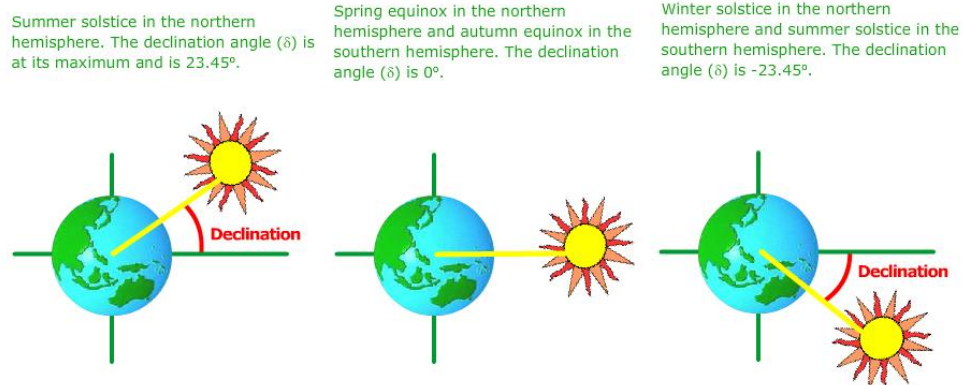
Proměnná Geom Mean Anom Sun ( $\xi$ ) slouží jen pro výpočet Eccent Earth Orbit( $\mu$ ), proto ji můžeme vypustit.

### Vysvětlení některých astronomických pojmů

**Juliánský datum** (Julian Day) je počet dní (a jejich částí), které uběhly od roku 4712 před naším letopočtem.

Jedno **Juliánské století** (Julian Century) je definováno jako 36525 dní.

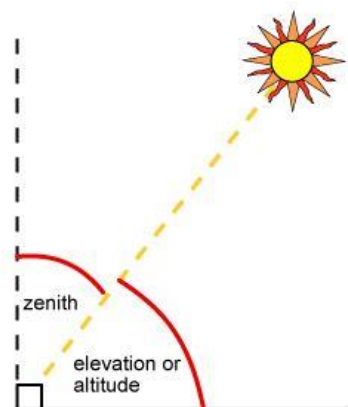
**Deklinace slunce  $\delta$**  (Sun Decline) udává úhel, pod kterým je skloněná osa rotace země. Nejvyšší hodnoty  $23.5^{\circ}$  nabývá o letním slunovratu a naopak nejnižší -  $23.5^{\circ}$  o zimním slunovratu. [13]



Obrázek 4: Deklinace Slunce [13]

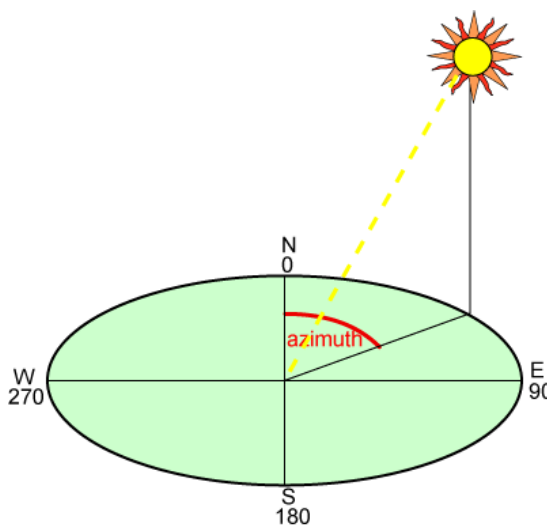
**Elevace slunce** (Solar Elevation Angle) neboli výška slunce nad obzorem představuje úhel mezi pozorovacím místem (vodorovnou osou) a sluncem. Tento úhel lze určit podle času, zeměpisných souřadnic a sluneční deklinace (musíme znát i datum).

**Zenitový úhel slunce** (Solar Zenith Angle) je úhel mezi sluncem a svislou osou, proto elevaci získáme tak, že tento úhel odečteme od  $90^\circ$ . [14]



Obrázek 5: Elevace slunce [14]

**Azimut** (Solar Azimuth Angle) udává ze které strany přichází sluneční svit, přičemž sever je brán jako  $0^\circ$ .



Obrázek 6: Azimut [15]



## 2.3 Raspberry Pi

### 2.3.1 Seznámení

Raspberry Pi je miniaturní plnohodnotný počítač, který se po připojení patřičných periférií dá provozovat jako běžný kancelářský počítač, prohlížet na něm web nebo dokonce hrát nenáročné hry.

Největší oblibě se však těší díky jeho rozhraní **GPIO** s dnes již obvykle 40 piny, díky kterému dokáže interagovat s okolním světem. Tyto piny lze nastavovat jako vstupy i výstupy a připojovat k nim různé senzory, spínače, led diody, relé nebo celé přídatné moduly.

Lidé tyto mikropočítače používají k vytváření vlastních meteostanic, provozování webových serverů, ovládání robotů a mnoha dalším účelům. [16], [17] Využití těchto zařízení je téměř neomezené a záleží jen na fantazii a schopnostech uživatele. Mezi nejextrémnější projekty patří pořízení fotek země z balónu na hranici vesmíru nebo spojení 64 Pi pomocí Lego kostek a vytvoření experimentálního superpočítače. [16]

### 2.3.2 Specifikace

Raspberry Pi je s oblibou označováno jako počítač o velikosti kreditní karty, což platí u nejvýkonnějších modelů řady B/B+ (v době psaní je nejnovější model 4), existují však ještě menší a úspornější řady A/A+ a nejmenší a nejlevnější Zero.

Základem všech Raspberry Pi je mikroprocesor architektury ARM [18].

Doporučovaným operačním systémem je Raspbian (odvozený z Debianu), který je pro Raspberry Pi přímo vytvořen, lze však nainstalovat i některé jiné upravené Linuxové distribuce nebo dokonce speciální verzi Windows 10. [19]

### 2.3.3 Trocha historie

První Raspberry Pi byl vyvinut britskou organizací Raspberry Pi Foundation v roce 2012 a byl původně určený k tomu, aby se na něm děti mohli učit programovat. [18]

Název vznikl spojením slov **Raspberry** (malina), které bylo zvoleno, protože názvy ovoce mají v počítačovém průmyslu tradici a slovo **Pi** je zkomolená zkratka slova *Python*, protože to byl první (a původně jediný) podporovaný programovací jazyk. [18]

## 3 Metodika

Projekt lze rozdělit do třech hlavních částí. Přičemž stěžejním předmětem práce je návrh a implementace řídicí jednotky, druhou částí konstrukce miniaturního modelu elektrárny s jedním solárním panelem a posledním krokem implementace monitorovací aplikace.

### 3.1 Řídicí jednotka

Základem řídicí jednotky je dle zadání mikropočítač Raspberry Pi, zvolena byla verze Zero. Na mikropočítači běží skript napsaný v jazyce Python verze 3, který podle vzorce na základě denní doby, data a zeměpisných souřadnic určuje úhly pro natočení a náklon panelu. Úhly jsou následně přepočítávány na čas, po který je nutné sepnout konkrétní relé, která ovládají příslušné elektromotory. Díky zvolení této metody je možné využít motory libovolného otočného zařízení, bez ohledu na to, zda jsou vybaveny zpětnou vazbou a tím je dosaženo vysoké univerzálnosti řídicí jednotky. Vše je umístěné v elektroinstalační krabici s odolností proti prachu a vodě, tak aby byl mikropočítač s ostatními součástkami dobře ochráněn proti povětrnostním vlivům.

### 3.2 Model

Model je konstruován tak, aby co nejvíce odpovídal reálným slunečním elektrárnám. Tak by mělo být zajištěno, že řídicí systém, který je schopný správně obsluhovat tento miniaturní model, by byl po následném připojení ke skutečnému velkému otočnému panelu schopný fungovat beze změny nebo s co nejmenším počtem drobných úprav. Jedná se o systém otočný ve dvou osách, přičemž hlavní je vertikální osa připevněná k základně. Jako základ konstrukce byly použity dvě nerezové tyče a většina ostatních dílů byla vyrobena pomocí 3D tisku. K pohonu slouží dva stejnosměrné motory napájené 12V.

### 3.3 Webová aplikace

Monitorovací aplikace přijímá data z Raspberry Pi, ukládá je do databáze a do grafu zobrazuje výrobu elektrické energie v závislosti na čase.

## 4 Výběr hardwaru

### 4.1 Řídící jednotka

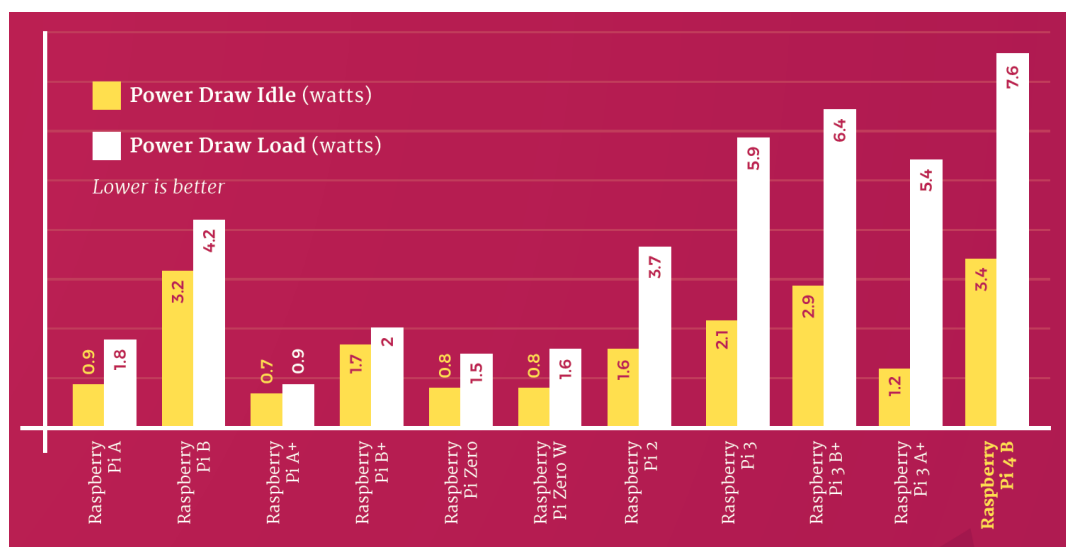
#### 4.1.1 Raspberry Pi

Jako základ řídicí jednotky bylo zvoleno **Raspberry Pi Zero WH**. Zero je nejmenší, nejlevnější a nejspornější verzi Raspberry Pi. Písmeno **W** značí, že tato varianta obsahuje vestavěnou Wifi a Bluetooth anténu. Písmeno **H** znamená, že tato varianta je osazena GPIO rozhraním se 40 piny. Tento model se dá v době psaní práce pořídit za cenu lehce přesahující 400 Kč. [20]

Zero je postaveno na starším čipsetu BCM2835 s jednojádrovým procesorem ARMv6 a disponuje operační pamětí 512 MB. Pro operační systém a ukládání slouží (jako u všech Raspberry Pi) paměťová karta, která není součástí zařízení. Pro tento účel byla použita microSDHC karta o kapacitě 32GB.

Tento model byl zvolen nejen pro jeho nízkou cenu, ale především díky jeho **nízké spotřebě energie**. Z malé spotřeby zákonitě plyne, že se bude méně zahřívat, což bude při zavření do voděodolné krabice klíčové.

Procesor mikropočítače byl osazen pasivním hliníkovým chladičem o velikosti 14 x 14 x 10mm.



Obrázek 7: Porovnání spotřeby RPi [21]

### 4.1.2 Relé

Ke spínání motorů slouží čtyři relé s třemi přepínacími kontakty. Tyto relé jsou spínány pomocí 4-kanálového reléového modulu s optočleny, který lze ovládat pomocí 3.3V přímo z Raspberry Pi.

### 4.1.3 Zdroje

K napájení Raspberry Pi byl zakoupen oficiální 5.1V/2.5A zdroj. Motory jsou napájeny zdrojem 12V/2A od firmy JDC.

### 4.1.4 Modul reálného času

Aby mohla řídicí jednotka správně fungovat i bez připojení k internetu je třeba k Raspberry Pi připojit hodiny reálného času. Konkrétně byl vybrán oblíbený model **DS1307**, který s mikropočítačem komunikuje přes sběrnici I2C.

### 4.1.5 Krabice

Řídicí jednotka je umístěna v plastové elektroinstalační krabici o rozměrech 240x190x90mm s certifikací odolnosti IP56.

## 4.2 Model

### 4.2.1 Motory

Pro otáčení i naklápění panelu byly zvoleny stejnosměrné motory napájené **12V** s označením **S30K**. Tento model byl vybrán pro jeho nízkou rychlost **14RMP**(otáček za minutu), které je dosaženo vestavěnou převodovkou. Konstrukce s úchyty umožňuje snadné přišroubování k modelu. Prodejci také udávají velký kroutivý moment. [22]

### 4.2.2 Solární panel

Byl použit solární panel o velikosti 6x10cm s maximálním udávaným výkonem 1W a jmenovitým napětím 6V.

### 4.2.3 Anemometr

Jako anemometr slouží vrtulka s pulzním výstupem původně určená k domácí meteostanici.

### 4.2.4 Koncové spínače

Na modelu jsou nainstalovány mikrospínače MSW-1 s jedním přepínacím kontaktem.

## 4.2.5 Imitace elektroměru

Elektroměr imituje PLC Siemens LOGO 6ED10522CC010BA6.

# 5 Výběr Softwaru

## 5.1 Řídící systém

### 5.1.1 Operační systém

Zvolen byl obligátní **Raspbian**, který je oficiálním operačním systémem Raspberry Pi Foundation. [23] Proto lze očekávat, že bude nejlépe odladěn a také bude nabízet nejlepší podporu. Konkrétně byla nainstalována verze **10** (Buster) s jádrem Linux 4.19.75+armv6l.

### 5.1.2 Programovací jazyk

**Python** byl vybrán z toho důvodu, že je to doporučovaný a nejčastěji používaný programovací jazyk pro vývoj na Raspberry Pi. Jedná se o vysokoúrovňový skriptovací dynamicky interpretovaný programovací jazyk. Interpret (pro verzi 2 i 3) bývá defaultně předinstalovaný v operačním systému Raspbian. Obsahuje všechny potřebné knihovny pro práci se vstupy a výstupy a práci s časem (viz další kapitola). Také asi největší počet různých návodů pro projekty s Raspberry Pi lze dohledat právě pro tento jazyk. Konkrétně byl použit Python verze 3.7.3.

### 5.1.3 Knihovny

Pro kontrolu vstupů a výstupů slouží knihovna **RPi.GPIO**, která umožňuje ovládat piny rozhraní GPIO.

Knihovna **math** je určena pro výpočet různých matematických funkcí, zajímavé pro nás budou zejména ty goniometrické.

Základním modulem pro práci s časem je knihovna **time**, ta obsahuje mimo jiné stejnojmennou funkci `time()`, která vrátí aktuální čas ve formátu desetinného čísla představující počet sekund, které uběhly od 1.1.1970. Další užitečná funkce `sleep()` umožní program zastavit na dobu, kterou předáme jako parametr v jednotkách sekund.

Knihovna **datetime** slouží též pro práci s časem a podporuje uživatelsky přívětivější objektový přístup.

## 5.2 Webová aplikace

### 5.2.1 Programovací jazyky

Jelikož všechny ostatní části byly psané v **Pythonu**, i pro backend webové aplikace byl zvolen tento programovací jazyk. Zobrazovaný formulář je napsán ve značkovacím jazyce **HTML** a použitý graf je vytvořen pomocí **JavaScriptu**.

### 5.2.2 Frameworky/Knihovny

Webový server je napsán pomocí pythonovského microframeworku **Flask**.

Kromě standartních knihoven pro práci s časem byla v aplikaci použita knihovna **sqlite3** pro ukládání dat do vestavěné relační databáze SQLite.

O vykreslování grafu se stará JavaScriptová knihovna **Chart.js**.

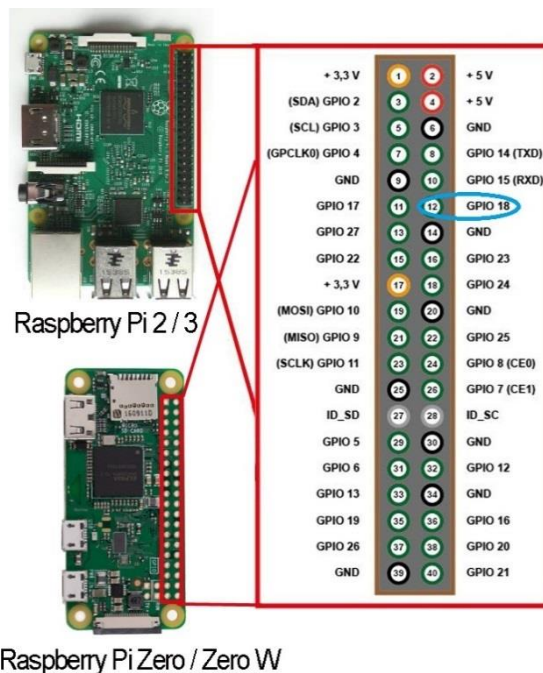
## 6 Praktická Část

### 6.1 Řídící systém

#### 6.1.1 Hardwarová implementace

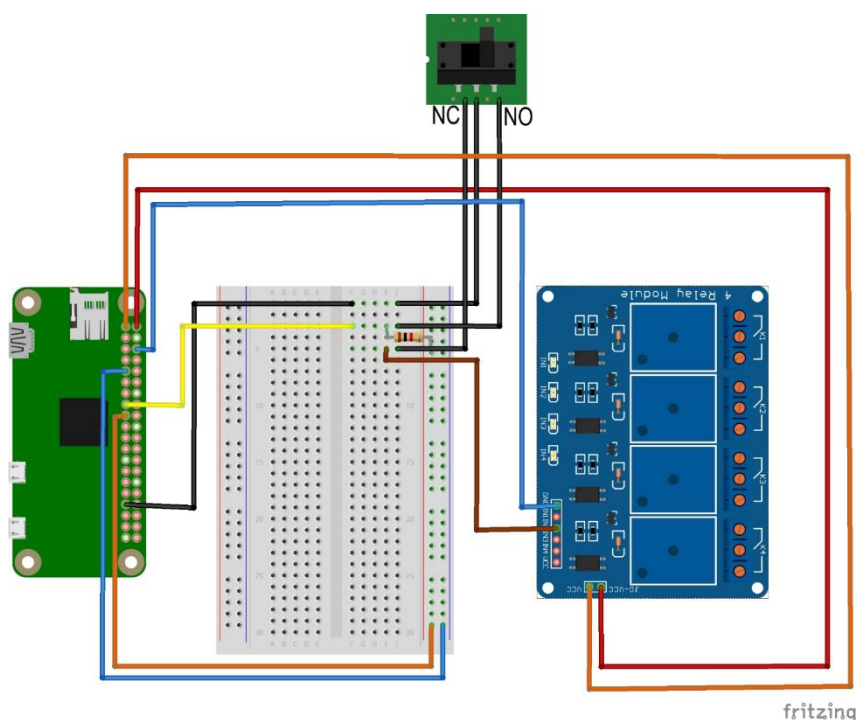
Mozkem řídicí jednotky je, jak již bylo několikrát zmíněno, mikropočítač Raspberry Pi.

Nejprve je třeba se podívat, jak má rozmístěné GPIO piny. Existují dva druhy číslování BCM a BOARD. V našem případě bylo zvoleno číslování BCM (tedy čísla psaná z vnějšku).



Obrázek 8: Raspberry Pi GPIO piny [24]

Protože výstupy Raspberry Pi pracují s napětím 3.3V, nelze s nimi přímo ovládat silová relé, která spínají motory. Je třeba použít reléový modul určený pro něj (respektive pro Arduino). Vstupy tohoto modulu jsou vybaveny optočleny, které galvanicky oddělují výstupy Raspberry Pi od ostatních obvodů modulu, čímž jsou výstupy dokonale ochráněny. Raspberry Pi je propojeno s modulem podle následujícího obrázku. Pro přehlednost je znázorněno propojení pro jednu osu a jeden směr.



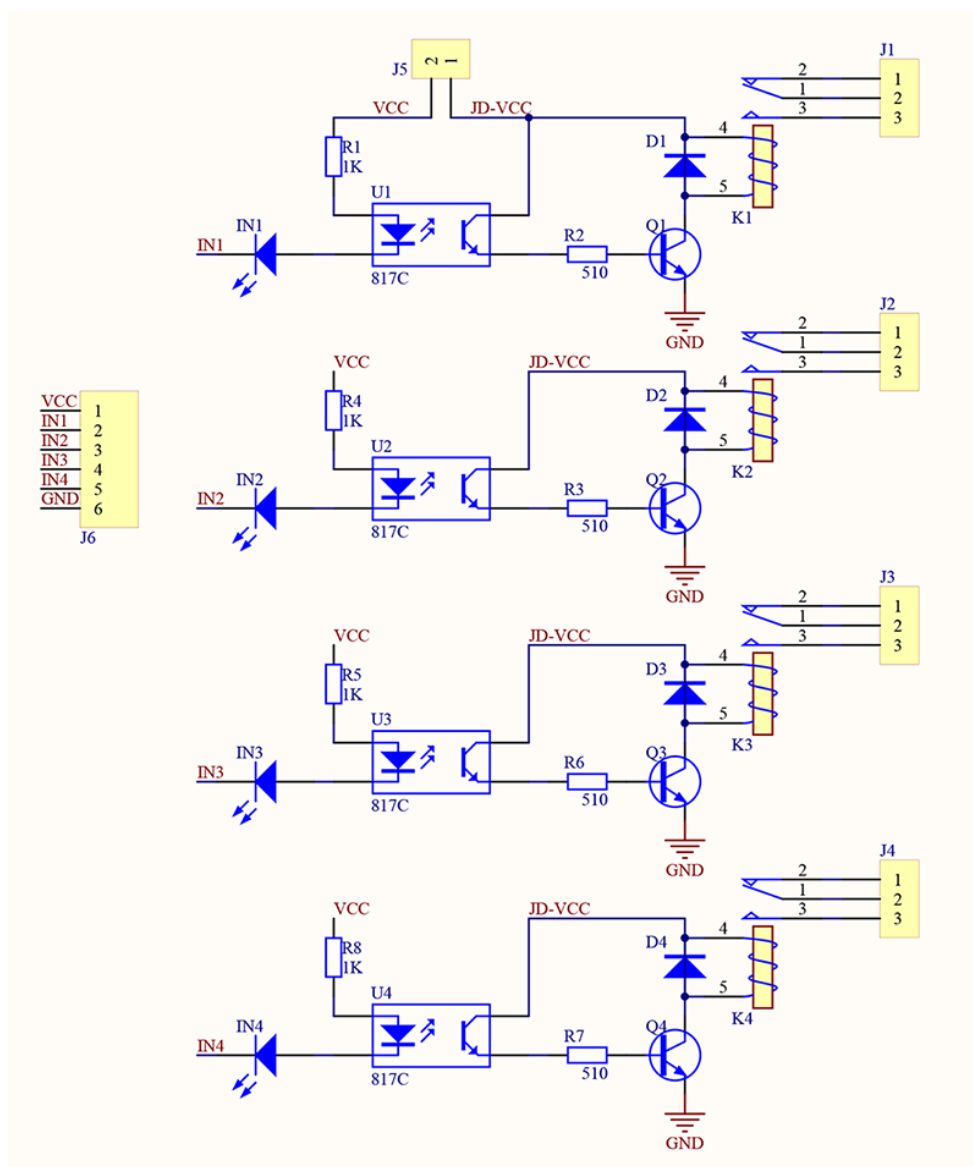
Obrázek 9: Zapojení relé-modulu

Jak je z obrázku patrné, pinem 13 je aktivován vstup IN2, který spíná relé K2. Do cesty je mu vřazen rozpínací (NC) kontakt koncového spínače pro příslušný směr. Když se osa otočí do koncové polohy, dojde k přepnutí koncového spínače a tím i k přerušení aktivace vstupu. Tak je zajištěno, že se osa přestane otáčet vždy po dojetí na koncový spínač bez ohledu na to, jaká je hodnota na výstupu z Raspberry. A zároveň se přes spínací (NO) kontakt aktivuje vstup 22 na Raspberry Pi.

Pro korektní funkčnost je ke vstupu připojen jako pull-down rezistor odpor 1k. Bez tohoto rezistoru se vstup choval značně nespolehlivě (docházelo k falešným aktivacím), což se takto podařilo eliminovat.

Dále je třeba vyřešit napájení reléového modulu. Jelikož obvody modulu jsou napájeny 5V a piny Raspberry jsou limitovány 3.3V není možné připojit 5V napájení na pin VCC bloku J6.

Je nutné vyjmout propojku mezi piny 1 a 2 bloku J5. Na pin 1(JD-VCC) přivedeme 5V a na pin 2(VCC) 3.3V.

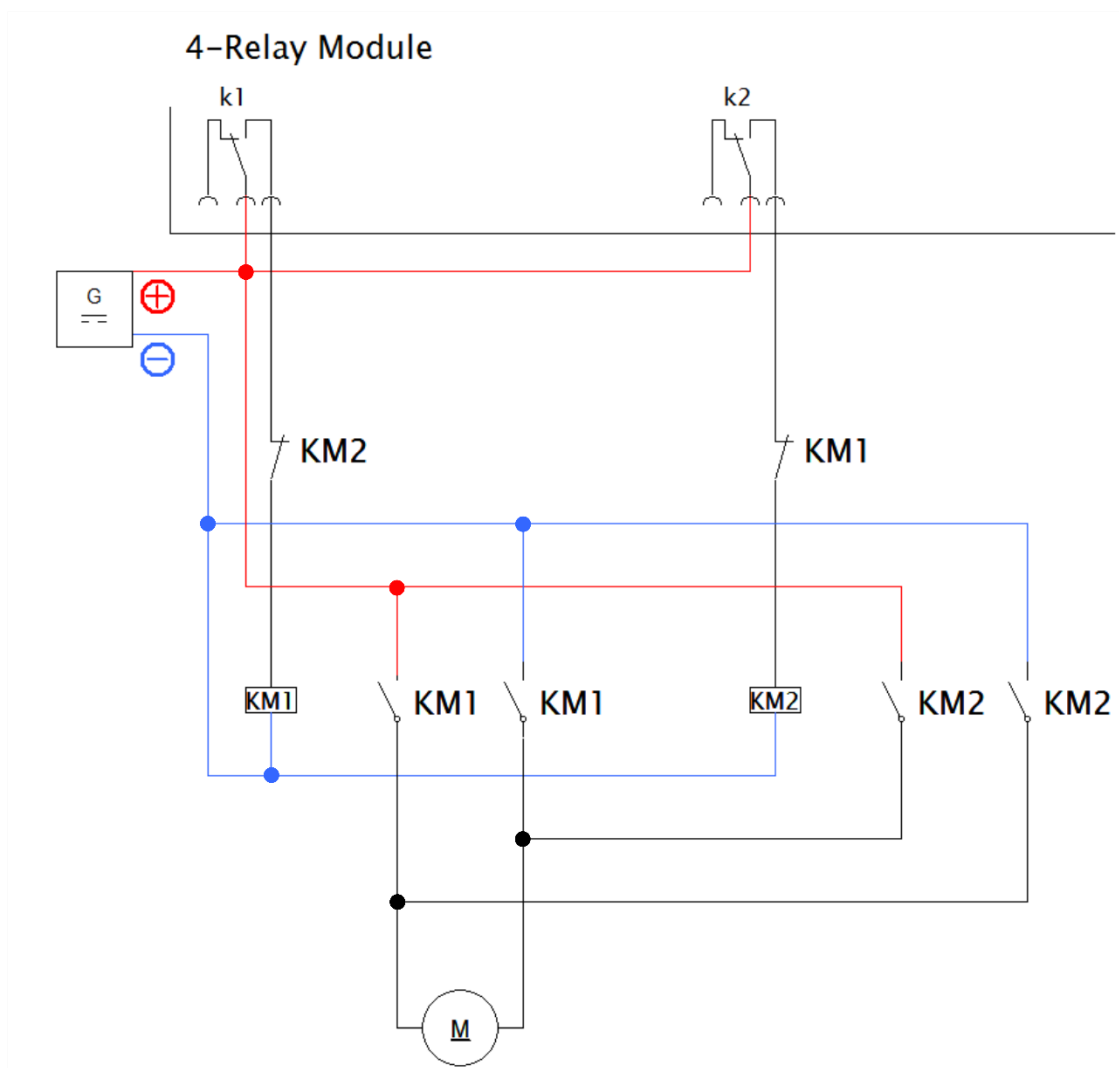


Obrázek 10: Vnitřní zapojení reléového modulu [25]

Jak je ze schématu zřejmé, tímto zajistíme, že 5V napájení obvodů modulu je odděleno od výstupů Raspberry pomocí optočlenů, které mají své napájení 3.3V. Pro sepnutí optočlenu je tedy na vstup nutné přivést logickou nulu. Z toho vyplývá, že na výstupech musíme používat obrácenou logiku.



Výstupními relé modulu k1-k4 jsou spínány silová relé KM1-KM4, která slouží pro spouštění motorů. Následující schéma řeší ovládání motoru jedné osy v obou směrech. Konkrétně se jedná o horizontální osu (pohyb nahoru/dolů).

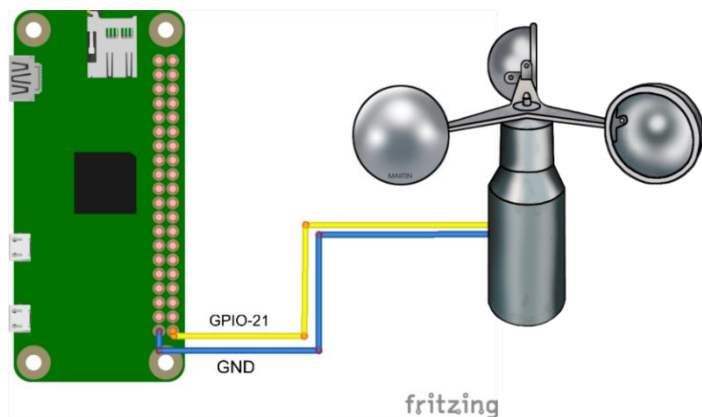


Obrázek 11: Schéma relé logiky

Relé k1 spíná relé KM1, jehož dva spínací kontakty převádí napájecí napětí pro motor M (pohyb dolů). Relé k2 spíná relé KM2, jehož dva spínací kontakty rovněž převádí napájecí napětí pro motor M, které je ovšem k motoru připojeno v opačné polaritě než z KM1, tím je docíleno, že se motor otáčí opačným směrem (pohyb nahoru).

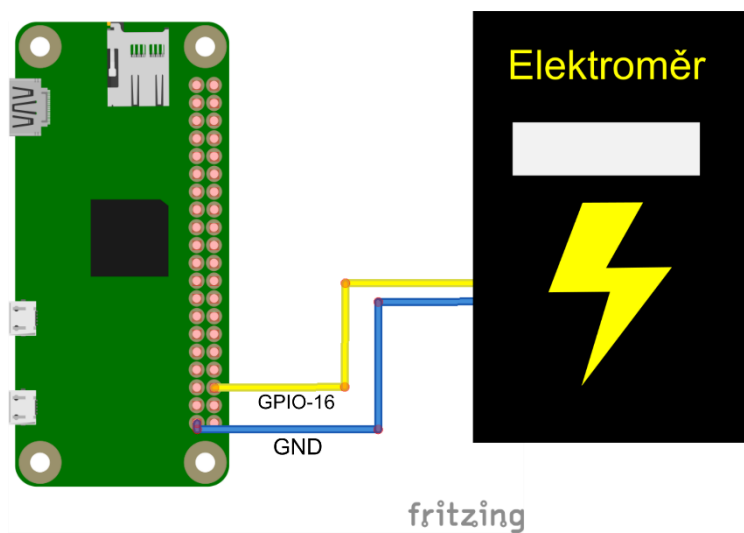
Aby bylo zabráněno nežádoucímu současnému sepnutí relé KM1 a KM2, které by nevyhnutelně vedlo ke zkratu zdroje pro napájení motorů, jsou tyto proti sobě blokovány. Mezi k1 a KM1 je vložen rozpínací kontakt KM2 a mezi k2 a KM2 rozpínací kontakt KM1. Tím je hardwarově zajištěno, že KM1 může sepnout jen v případě, že KM2 je v klidovém stavu a naopak.

Anemometr je připojen na pin 21.



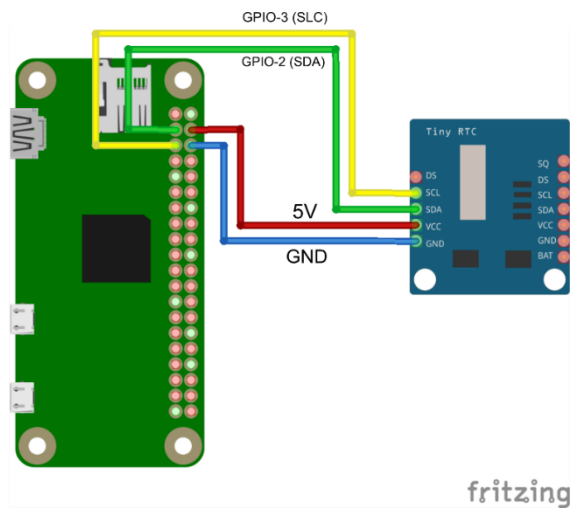
Obrázek 12: Zapojení anemometru

K připojení elektroměru slouží pin 16.



Obrázek 13: Zapojení elektroměru

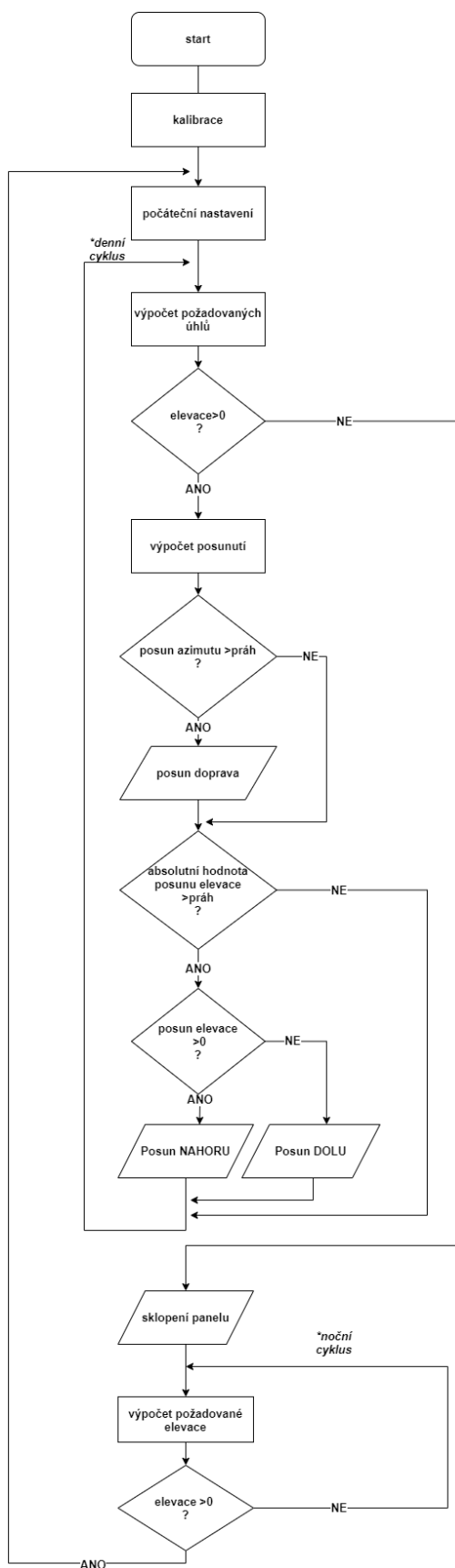
Časový modul musí být připojen na sběrnici I2C a je napájen pomocí 5V.



Obrázek 14: Zapojení RTC

## 6.1.2 Softwarová implementace

Princip řízení natáčení panelu, lze popsat následujícím diagramem:



Obrázek 15: Princip řízení

Při startu se provede kalibrace.

Na začátku dne se panel se natočí výchozí polohy (azimut  $45^\circ$  a elevace  $0^\circ$ ). Následně program přejde do denního cyklu.

Nejprve jsou vypočítány požadované úhly, ze kterých je následně vypočten posun v obou osách. Posun se provede jen v případě, že je větší než stanovené minimum. Tím se zamezí příliš častému spínání relé, které by vedlo k jejich rychlejšímu opotřebení, a hlavně k zvýšení nepřesností, které jsou způsobeny prodlevou mezi sepnutím výstupu a pohybem panelu. Tato prodleva je velmi malá, ale projeví se při každém sepnutí, nepřesnosti se tedy sčítají, proto by při příliš častém spínání přestali být zanedbatelné. Konkrétně bylo minimum nastaveno na  $5^\circ$ . U elevace je ještě třeba zjistit, zda slunce stoupá nebo už začalo klesat. Tento proces se neustále opakuje.

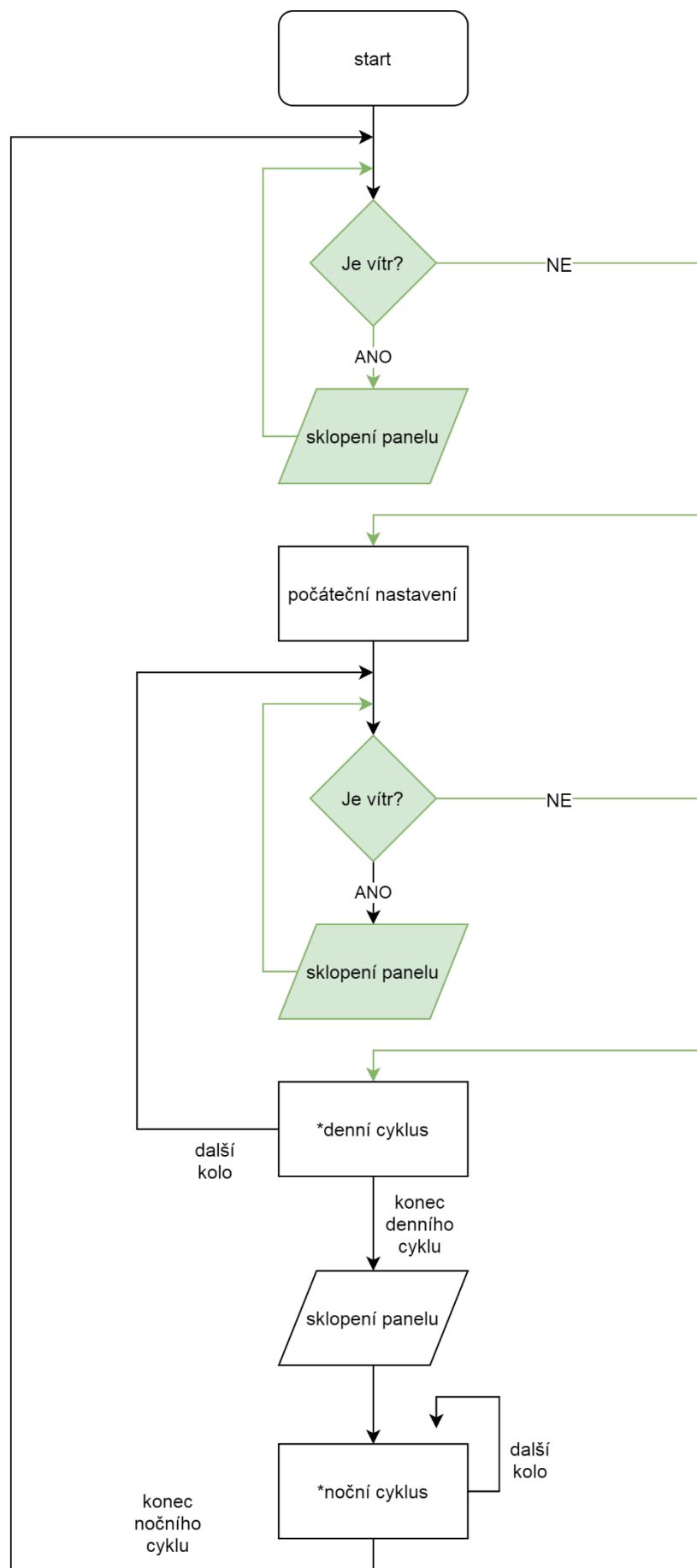
Pokud je elevace záporná, značí to, že slunce již zapadlo za obzor a není třeba pokračovat v natáčení panelu. Panel je na noc sklopen. V noci je pouze hlídána hodnota elevace a jakmile překročí nulu, program opět skočí na začátek a zahájí proces sledování slunce.

Dále je třeba do algoritmu zařadit hlídání větru.

Během dne je třeba kontrolovat, jestli nefouká moc silný vítr a pokud ano sklopit panel do vodorovné polohy. Takto sklopený panel vytváří nejmenší odpor, a tudíž nejlépe odolá foukajícímu větru. Jakmile silný vítr ustane je možné opět začít s natáčením panelu.

Stejně tak je třeba vítr hlídat na začátku programu před zahájením úvodních konfigurací. Večer je panel sklopen preventivně, proto není třeba vítr hlídat během noci.

Hlídání větru je znázorněno na následujícím diagramu, kde je denní a stejně tak i noční cyklus znázorněn jedním blokem (označeno \*).



Obrázek 16: Hlídaní větru

Celý program je rozdělen do následujících souborů, respektive tříd:

**Relays.py** obsahuje pokyny pro pohyb do všech stran a stará se o spínání příslušných relé.

**Calibration.py** obsahuje metody pro nastavení os do koncových poloh a zajišťuje přepočty ze stupňů na čas.

**SolarCalc.py** vypočítá aktuální požadovanou elevaci a azimut (kde se nachází slunce) ve stupních.

**Wind.py** hlídá, jestli je překročen počet otáček anemometru (zda fouká moc silný vítr).

Hlavní skript, který se nachází v souboru **solar\_tracking.py**, pomocí volání metod výše zmíněných tříd vykonává algoritmus popsaný diagramem.

### **Relays.py:**

Pro spínání výstupů je třeba nainstalovat knihovnu RPi.GPIO, kterou bylo nejprve třeba nainstalovat. A kvůli práci s časem knihovnu time:

```
import RPi.GPIO as GPIO
import time
```

Inicializace výstupů pro jednotlivá relé se provádí v konstruktoru:

```
def __init__(self, left=5, right=19, up=13, down=6) :
    self.left=left
    self.right=right
    self.up=up
    self.down=down

    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(self.right, GPIO.OUT)
    GPIO.setup(self.left, GPIO.OUT)
    GPIO.setup(self.up, GPIO.OUT)
    GPIO.setup(self.down, GPIO.OUT)
```

Jak bylo vysvětleno v hardwarové části (viz stránka:18), kvůli přítomnosti optočlenů je otočena logika a relé jsou spínány logickou nulou (tedy False).

Posun doprava:

```
def go_right(self):
    GPIO.output(self.right, False)
```

Posun doprava o:

```
def go_right_by(self, degrees):
    GPIO.output(self.right, False)
    time.sleep(degrees)
    GPIO.output(self.right, True)
    time.sleep(0.1)
```

Stupně, o které se osa otáčí, jsou přepočteny na čas, takže parametr *degrees* je ve skutečnosti v jednotkách sekund.

Zastavení pohybu doprava:

```
def stop_right(self):
    GPIO.output(self.right, True)
```

Stejným způsobem je realizován pohyb pro všechny směry.

### **Calibration.py:**

Protože jsou relé spínány pomocí logické nuly i koncové spínače po přepnutí převádějí na vstup Raspberry tutéž logickou nulu.

Konfigurace vstupů z koncových spínačů:

```
GPIO.setup(17, GPIO.IN, GPIO.PUD_UP) #down
GPIO.setup(22, GPIO.IN, GPIO.PUD_UP) #up
GPIO.setup(23, GPIO.IN, GPIO.PUD_UP) #left
GPIO.setup(24, GPIO.IN, GPIO.PUD_UP) #right
```

Pomocí parametru PUD\_UP je softwarově připojen integrovaný pull-up rezistor, tím je zajištěno, že vstup je aktivován přivedením logické nuly z koncového spínače.

Zjištění, jak dlouho trvá ujet jeden stupeň dolu/nahoru:

```
def elevation(self):
    repeat=True
    repeat2=False
    while (repeat):
        Calibration.relays.go_up()
        if (GPIO.input(22)==False):
            #sepnuty první(horní) koncový spínač
            Calibration.relays.stop_up()
            time.sleep(1)
            begin=time.time() #začátek měření
            repeat=False
            repeat2=True
    while (repeat2):
        Calibration.relays.go_down()
        if (GPIO.input(17)==False):
            #sepnuty druhý(dolní) koncový spínač
            Calibration.relays.stop_down()
            end=time.time() #konec měření
            repeat2=False

    measured_time=end-begin
    one_degree=measured_time/90

    return one_degree
```

Panel se nejprve nastaví do horní výchozí polohy, poté je měřena doba, než dojde k dolnímu koncovému bodu. Celková dráha činí 90°, proto naměřený čas dělíme devadesáti a vyjde nám, jakému času odpovídá jeden stupeň nahoru/dolů.

Obdobným způsobem je na čas přepočten i stupeň doprava/doleva.

### **SolarCalc.py**

Zde je potřeba nainportovat modul datetime pro práci s časem a kalendářem a matematickou knihovnu math pro výpočty:

```
import datetime
import math
```



## Výpočet polohy slunce:

```
#zemepisne udaje
latitude = 48.77 #Latitude-zemepisna sirka
longtitude = 14.98 #Longtitude-zemepisna delka
time_zone = +1 #Time Zone-casove pasmo

#casove udaje
date = datetime.datetime.now() #aktualni datum
midnight = datetime.datetime.combine(date.date(),
    datetime.time()) #pulnoc
minutes=((date - midnight).seconds)/60 #minuty od pulnoci

julian_day = 367 * date.year - int((7 * (date.year +
    int((date.month + 9) / 12.0))) / 4.0) + int((275 *
    date.month) / 9.0) + date.day + 1721013.5 + (date.hour +
    date.minute / 60.0 + date.second / math.pow(60, 2)) / 24.0 -
    0.5 * math.copysign(1, 100 * date.year + date.month -
    190002.5) + 0.5 -time_zone/24 # julianske datum
julian_century = (julian_day-2451545)/36525 #Juliánské století

#vypocty
geom_mean_long_sun = (280.46646 + julian_century * (36000.76983 +
    julian_century * 0.0003032)) % 360#
geom_mean_anom_sun = 357.52911 + julian_century * (35999.05029 -
    0.0001537 * julian_century)

. . .
celý kód lze nalézt v přílohách
```

## Elevace:

```
elevation=90-solar_zenith_angle
```

## Azimut:

```
if (hour_angle>0):

azimuth=(math.degrees(math.acos(((math.sin(math.radians(latitude))
    *math.cos(math.radians(solar_zenith_angle)))-
```

```

math.sin(math.radians(sun_declin)))/(math.cos(math.radians(latitude)
e))*math.sin(math.radians(solar_zenith_angle))))+180) %360

else:
azimuth=(540-
    math.degrees(math.acos(((math.sin(math.radians(latitude))*ma
th.cos(math.radians(solar_zenith_angle)))-
math.sin(math.radians(sun_declin)))/(math.cos(math.radians(latitude)
e))*math.sin(math.radians(solar_zenith_angle))))))%360

```

Jedná se o vzorce ze stránek NOAA (viz kapitola: Astronomické vazby) přepsané do Pythonu.

### Wind.py:

Hlídaní větru:

```

def is_wind(self):
    impulse = int(0)
    recorded=0
    timeout=time.time()+Wind.MEASURING_TIME
    strong_wind=False

    while(time.time()<timeout):

        if(GPIO.input(21) == False):
            if(recorded == 0):
                impulse = impulse + 1
                recorded = 1
            else:
                recorded=0

        if (impulse>Wind.THRESHOLD):
            strong_wind=True
        else:
            strong_wind=False

    return strong_wind

```

Po daný časový úsek je měřen počet pulzů z anemometru (počet pulzů je přímo úměrný počtu otáček vrtulky). Pokud je překročen práh, funkce vrátí hodnotu True, jinak vrátí hodnotu False.

Protože ve chvíli zaznamenávání impulzu stihne procesor mikropočítače vykonat obrovské množství cyklů, je třeba vytvořit proměnnou *recorded*, která se v prvním cyklu záznamu změní na 1 a po ukončení aktivace vstupu změní opět na 0. [26] Tak je docíleno, že každý impulz bude započítán jen jednou. Tento mechanismus se někdy nazývá hlídání hrany.

### Hlavní program:

Hlídání větru:

```
while True:
    is_strong_wind=wind.is_wind()
    if (is_strong_wind==False):
        break
    else:
        calibration.go_to_up_home()
        actual_elevation=90
```

Pomocí objektu třídy *Wind* je volána funkce *is\_wind()*, pokud ta vrátí True, panel je sklopen. Dokud nevrátí False, program je blokován v tomto místě kódu.

Kalibrace:

```
one_degree_right=calibration.azimuth()
one_degree_up=calibration.elevation()
```

Na začátku jsou provedeny kalibrace azimutu i elevace. Funkce *azimuth()* objektu třídy *Calibration* vrátí, jak dlouho trvá ujet jeden stupeň doprava a funkce *elevation()* vrátí, jak dlouho trvá ujet jeden stupeň nahoru/dolů.

Počáteční konfigurace:

```
calibration.go_to_left_home()
actual_azimuth=45

calibration.go_to_down_home()
actual_elevation=0
```

Následně je panel nasměrován do výchozí pozice a příslušné proměnné indikující pozici panelu jsou nastaveny na odpovídající hodnoty.

Výpočet posunutí:

```
required_azimuth=solar_calc.get_azimuth()
required_elevation=solar_calc.get_elevation()
if (required_elevation<0):
    break

azimuth_shift=required_azimuth-actual_azimuth
elevation_shift=required_elevation-actual_elevation
```

Pomocí metod instance třídy *SolarCalc* je vypočtena aktuální poloha slunce. Odečtením aktuální pozice panelu získáme požadovaný posun.

Posunutí doprava (azimut):

```
if (azimuth_shift>THRESHOLD):
    azimuth_shift_to_time=azimuth_shift*one_degree_right
    relays.go_right_by(azimuth_shift_to_time)
    actual_azimuth=required_azimuth
```

Pokud je požadovaný posun větší než stanovený práh (v našem případě konkrétně 5°), je panel otočen doprava. Požadovaný posun je nejprve přepočten na čas a předán jako parametr metodě *go\_right\_by()*, která patří objektu třídy *Relays*. Nakonec je aktualizována proměnná indikující pozici panelu.

Posunutí nahoru/dolu (elevace):

```
if (abs(elevation_shift)>THRESHOLD):

    if(elevation_shift>0):
        elevation_shift_to_time=elevation_shift*one_degree_up
        relays.go_up_by(elevation_shift_to_time)
        actual_elevation=required_elevation
```

```
else:
    elevation_shift_to_time=elevation_shift*one_degree_up
    relays.go_down_by(elevation_shift_to_time)
    actual_elevation=required_elevation
```

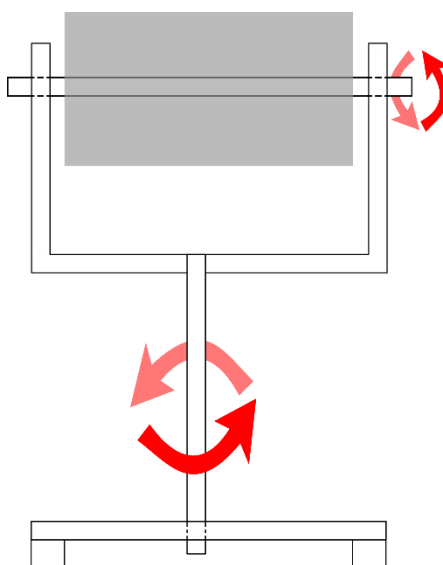
Nastavení elevace je obdobné jako u azimutu, jen je zde situace komplikovanější o to, že se slunce neposouvá stále jedním směrem, ale po polovině dne stoupání, začne opět klesat. Proto je zde třeba hlídat, jestli je posun záporný nebo kladný a podle toho otočit panelem nahoru nebo dolů.

Noční cyklus:

```
while True:
    required_elevation=solar_calc.get_elevation()
    if (required_elevation>0):
        break
    time.sleep(5)
```

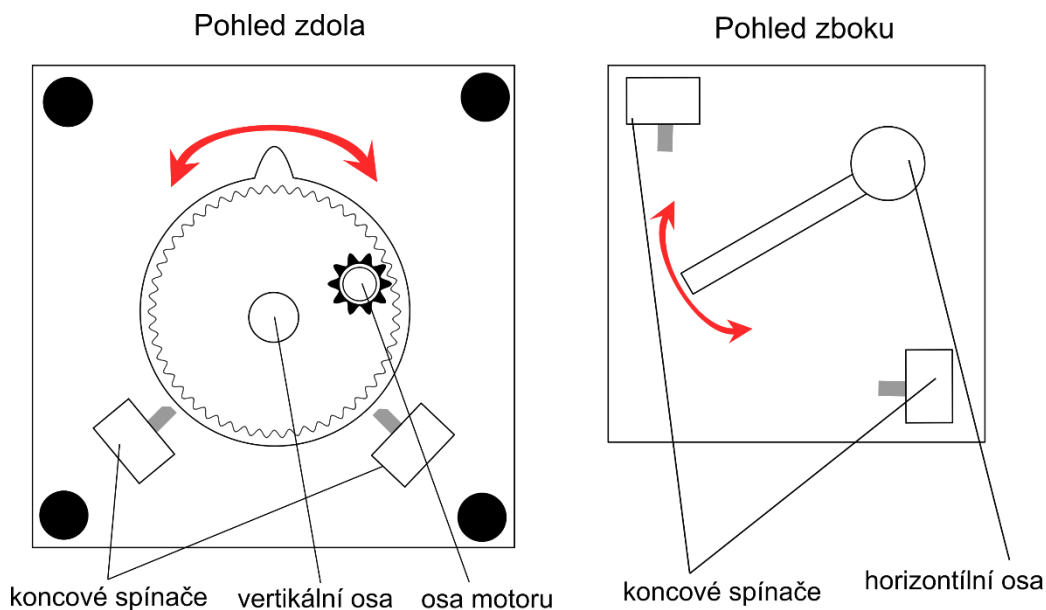
## 6.2 Model

Jak již bylo zmíněno v teoretické části, byla zvolena konstrukce umožňující natáčet panel ve dvou osách. Vertikální osa kolmá na základnu umožňuje otáčení panelu zleva doprava (azimut). K vrchnímu konci této osy je upevněna konstrukce horizontální osy umožňující otáčení zdola nahoru (elevace), s níž je pevně spojen panel.



Obrázek 17: Model-pohled zepředu

Obě osy jsou přes převodová kola spojeny s motory. Rozsah otáčení obou os je omezen koncovými spínači.

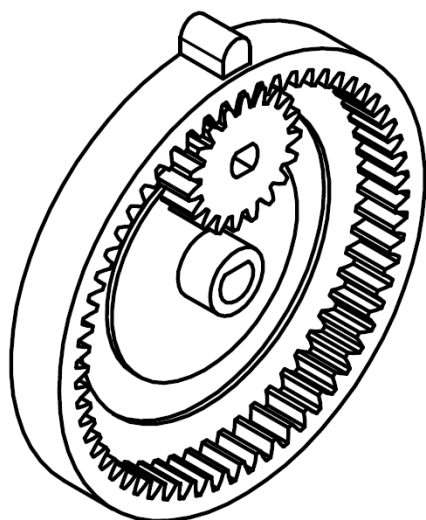


Obrázek 18: Model-pohled zdola a z boku

Podle tohoto návrhu byl vytvořen finální model. Osy jsou vyrobeny z nerezové oceli, ostatní konstrukční díly byly vytištěny na 3D tiskárně.

Základnu modelu tvoří deska o rozměrech 12x12cm. Na spodní části desky se nacházejí převodová kola a koncové spínače. Z toho důvodu je deska v rozích opatřena čtyřmi šrouby, které slouží jako nohy. Středem základny prochází vertikální osa, která je pod základnou spojena se sekundárním převodovým kolem. Na obvodu sekundárního kola je výčnělek, který spíná koncové spínače. S primárním kolem je spojena osa motoru, který je k základně z vrchu přišroubován. Vrchní část modelu tvoří díl ve tvaru U, skrz který prochází horizontální osa. Ta je na jedné straně spojena se sekundárním převodovým kolem (princip převodu je stejný jako u vertikální osy). Na druhé straně je osa zakončena tyčkou kolmou k ní, která slouží ke spínání koncových spínačů. S osou je pevně spojen držák panelu.

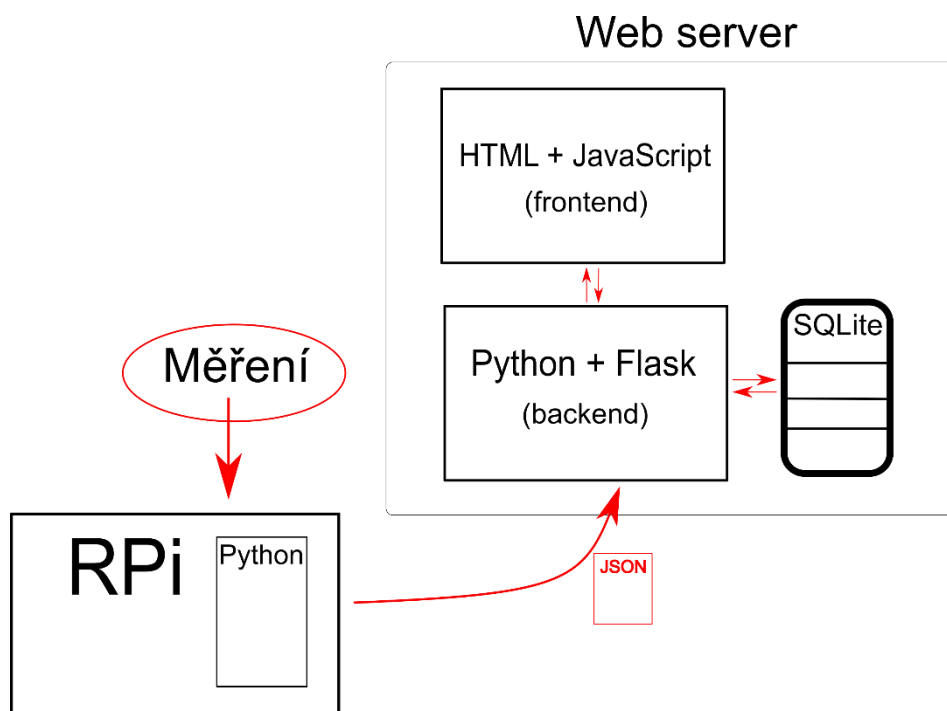
Pro zvýšení přesnosti je žádoucí, aby pohyb os byl co nejpomalejší. I když byly vybrány pomalé motory (12RPM) je třeba pohyb dále zpomalit pomocí převodů.



Obrázek 19: Převod

Sekundární kolo bylo zvoleno s vnitřním ozubením. Důvodem je výrazné zmenšení rozteče mezi hnanou a hnací osou a tím i zmenšení modelu samotného. Toto kolo má 59 a primární kolo 19 zubů. Převodový poměr je tedy  $59/19$ , z toho vyplývá, že pohyb bude zhruba 3krát zpomalen.

### 6.3 Webová aplikace



Obrázek 20: Architektura webové aplikace

Na Raspberry Pi je neustále měřena výroba elektrické energie a v pravidelných časových intervalech je na webový server odeslána informace o přírůstku.

Informace z elektroměru přichází ve formě impulzů, proto je jejich zpracování ze vstupu stejné jako u anemometru (a není jej třeba znovu popisovat).

K odesílání dat na webový server je třeba nainportovat knihovnu requests:

```
import requests
```

Měření přírůstku a odeslání dat na server:

```
if (time.time() > (last_time + 60) or last_time == 0):
    increase = impulse - last #prirustek (celkovy pocet impulzu
                              minus celkovy pri minulem mereni)
    last = impulse
    actual_time = int(time.time())
    last_time = time.time()
    data_from_pi = {'time': actual_time, 'increase': increase}
    try:
        response = requests.post('http://127.0.0.1:5000/api',
                                  json=data_from_pi)
        if response.ok:
            print(response.json())
    except:
        print("chyba na serveru")
```

Každých 60 sekund je na webový server ve formátu JSON odeslána informace o přírůstku impulzů od minulého měření.

Pro testování byl webový server spuštěn na tomtéž mikropočítači, proto je adresa nastavena na localhost. Při reálném využití by mohl webový server běžet kdekoli.

Webová aplikace přijímá data a ukládá je do databáze (SQLite).

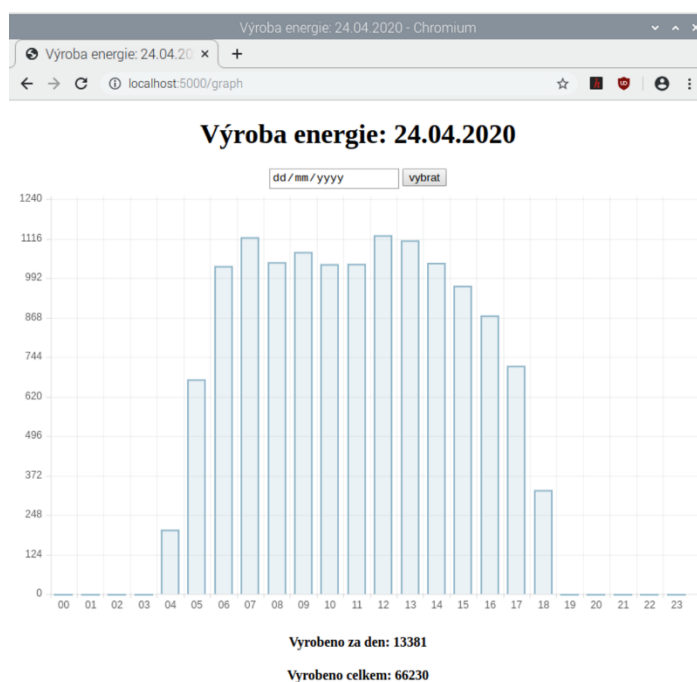
```
@app.route('/api', methods=['POST'])
def index():
    data = request.json
    db.insert_row(data['time'], data["increase"])
```



Uživatel si může přes HTML formulář zvolit libovolný den, který je předán jako parametr funkci `select_day()` třídy `DatabaseHandler`. Vybraná data jsou zpracována a předána šabloně pro vykreslení grafu. Při prvním načtení stránky je automaticky vybrán aktuální datum (obdobná metoda `graph_get()`).

```
@app.route('/graph', methods=['POST'])
def graph_post():
    total=db.count_total()
    date=request.form['date'].split('.')
    selected_day=
        db.select_day(int(date[2]),int(date[1]),int(date[0]))
    #zpracování dat ...
    return render_template('bar_chart.html', title='Výroba
    energie: ' + date[2] + '.' + date[1] + '.' + date[0],
        max=max_day_value + max_day_value / 10, labels=bar_labels,
        values=bar_values, day_total='Vyrobena za den: '
        +str(day_total),total='Vyrobena celkem: '+str(total))
```

Na stránce je kromě grafu, který zobrazuje výrobu v jednotlivých hodinách, také zobrazen součet vyrobené energie za daný den i za celou dobu. Celý výše zmíněný kód i samotnou třídu pro práci s databází lze nalézt v přílohách. Šablona s grafem je ze stránek Ruan Bekker's Blog [27] a využívá JavaScriptovou knihovnu Chart.js pro vykreslování grafů [28].



Obrázek 21: Ukázka webové aplikace

## 7 Testování

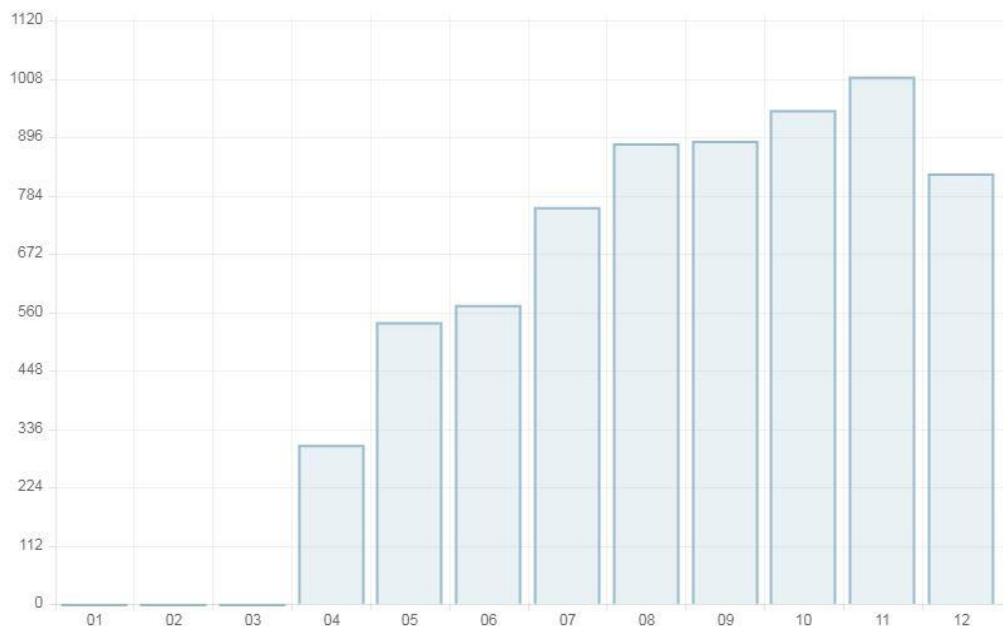
Všechny součásti systému byly testovány průběžně během vývoje. Po dokončení byl systém spuštěn a následně pečlivě pozorován.

Správný výpočet úhlů byl porovnán s výpočty online kalkulátoru NOAA. [29] Čísla v kontrolních výpisech aplikace se od výpočtů kalkulátoru lišila v řádech desetin stupně (což je naprosto zanedbatelné). Natočení panelu bylo porovnáno s vypočtenými úhly, rovněž s výsledky online aplikace SunCalc [30] a samozřejmě pohledem na oblohu. Při testování uvnitř se model natáčel do požadovaných úhlů bez zjevných nepřesností. Při testování venku se projeví mírné nepřesnosti způsobené ztuhlostí modelu při nízkých ranních teplotách.

Jak systém zareaguje na silný vítr, bylo odzkoušeno tak, že práh maximálního počtu otáček v kódu byl přepsán na velmi nízké číslo. Na každé roztočení vrtulky anemometru (rukou) systém zareagoval správně sklopením panelu, chvíli po ustálení bylo vždy znovu zahájeno sledování.

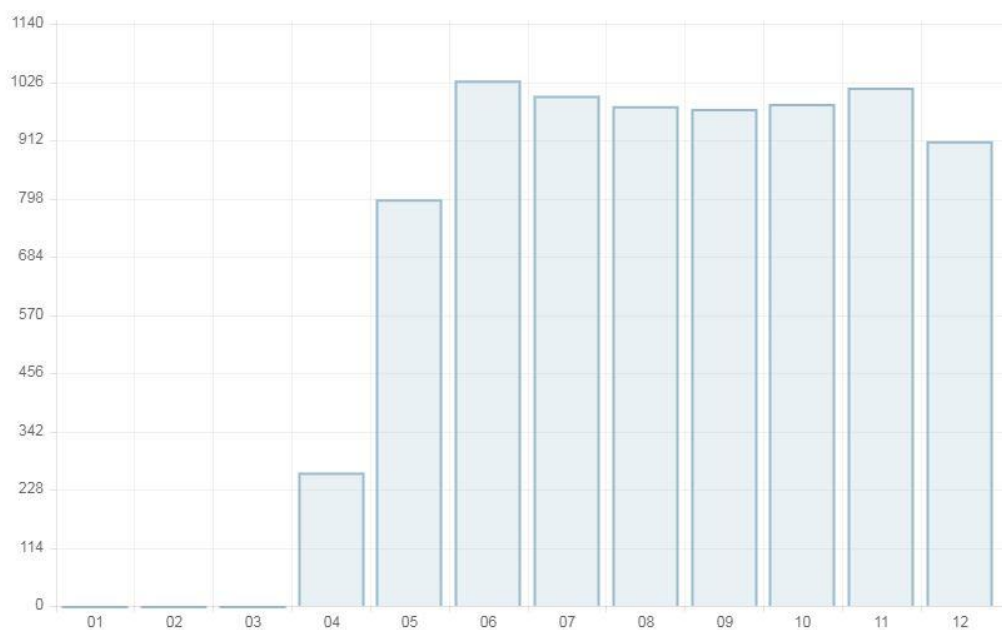
Skutečná elektrárna je vybavena elektroměrem s pulzním výstupem (bezpotencionální kontakt), který odešle jeden pulz při každé vyrobené kWh. U modelu tento elektroměr nelze použít, jelikož vyrobená energie není nikam dodávána a panel vyrábí nízké stejnosměrné napětí (6V). Proto byla za pomoci PLC Siemens Logo vytvořena imitace elektroměru. Lze vycházet z toho, že velikost napětí je přímo úměrná intenzitě slunečního osvětlení. Na analogový vstup PLC je připojeno napětí ze solárního panelu a výstup simuluje výstup elektroměru. PLC je naprogramováno tak, že se zvyšujícím se napětím stoupá četnost pulzů na výstupu. Jeden takto vyrobený pulz sice neodpovídá jednotce kWh, ale počet pulzů je přímo úměrný osvětlení panelu stejně jako u elektroměru.

V dvou po sobě jdoucích jasných dnech bylo provedeno měření. Jelikož byl model umístěn na parapetu, od nějž i okna se světlo odráželo na panel, byly tyto nežádoucí odrazy odstíněny pomocí kartonu. První den byl panel fixně namířen na jih s elevací 45°. Druhý den byl spuštěn program pro sledování slunce. Jak lze vyčíst z následujících grafů, výroba energie při zapnutém sledování slunce byla výrazně vyrovnanější. Rozdíl byl výrazný zejména v ranních hodinách, kdy je největší rozdíl mezi fixním namířením panelu a polohou slunce. Na konci 14 hodiny (12 hodiny GMT) slunce zajde za dům a další měření ztrácí smysl.



Výrobno za den: 6741

Obrázek 22: Výroba energie bez natáčení panelu



Výrobno za den: 7945

Obrázek 23: Výroba energie s natáčením panelu

Poznámka: Webová aplikace využívá čas počítače, na kterém je provozována. Pro účely testování byla v tomto případě spuštěna na Raspberry Pi, na kterém je z důvodu správné funkčnosti aplikace pro sledování slunce nastaven čas GMT. Z tohoto důvodu je čas na grafu posunutý o dvě hodiny vůči místnímu letnímu času.

## 8 Závěr

Podářilo se sestavit a zprovoznit řídící jednotku a model fotovoltaického panelu, který je tato řídící jednotka schopna obsluhovat. Tak bylo dokázáno, že za pomoci mikropočítače Raspberry Pi lze sestavit funkční řídící systém, který natáčí panel směrem ke slunci na základě času.

Webová aplikace zobrazuje informace o výrobě elektrické energie v závislosti na čase.

Jako možný rozvoj tohoto projektu se dá uvažovat tvorba propracovanější centralizované webové aplikace, která umožní tvorbu soukromých uživatelských účtů a přiřazování jednotlivých zařízení/soustav zařízení k těmto účtům a případné další funkcionality.

## Použitá literatura

- [1] BOXWELL, Michael. Solar electricity handbook: a simple, practical guide to solar energy- designing and installing solar PV systems electric systems [eBook]. Eleventh edition 2017. 70 Olton Boulevard East Birmingham B27 7NG United Kingdom: Greenstream Publishing, 2017 [cit. 2020-02-22]. ISBN 978-1-907670-67-1.
- [2] MLA STYLE. The Nobel Prize in Physics 1921: Albert Einstein. NobelPrize.org [online]. Nobel Media AB 2020 [cit. 2020-02-22]. Dostupné z: <https://www.nobelprize.org/prizes/physics/1921/summary/>
- [3] A History of Solar Cells: How Technology Has Evolved. <https://www.solarpowerauthority.com/> [online]. [cit. 2020-02-22]. Dostupné z: <https://www.solarpowerauthority.com/a-history-of-solar-cells/>
- [4] Jak funguje fotovoltaický neboli solární panel? EON: solar [online]. 7. 5. 2018 [cit. 2020-02-22]. Dostupné z: <https://www.eon-solar.cz/blog/1-jak-funguje-fotovoltaicky-neboli-solarni-panel>
- [5] Solární Otočná elektrárna FS. Vojik-avs.cz [online]. [cit. 2020-02-22]. Dostupné z: <http://www.vojik-avs.cz/index.php?id=otocnae&lang=cz>
- [6] Polohovací jednotky. Prosolar.net [online]. [cit. 2020-02-22]. Dostupné z: <http://www.prosolar.net/polohovaci-jednotky.php>
- [7] LUDT, Billy. What is a solar tracker and how does it work? Solarpowerworldonline.com [online]. 16.1.2020 [cit. 2020-02-23]. Dostupné z: <https://www.solarpowerworldonline.com/2020/01/what-is-a-solar-tracker-and-how-does-it-work/>
- [8] ZIPP, Kathie. How does a solar tracker work? Solarpowerworldonline.com [online]. 4.4.2013 [cit. 2020-02-23]. Dostupné z: <https://www.solarpowerworldonline.com/2013/04/how-does-a-solar-tracker-work/>
- [9] Solar Calculation Details. Esrl.noaa.gov [online]. [cit. 2020-02-24]. Dostupné z: <https://www.esrl.noaa.gov/gmd/grad/solcalc/calcdetails.html>
- [10] MEEUS, Jean. Astronomical algorithms. Richmond, Va.: Willmann-Bell, c1991. ISBN 09-433-9635-2

- [11] About our agency. NOAA.gov: National Oceanic and Atmospheric Administration [online]. [cit. 2020-02-24]. Dostupné z: <https://www.noaa.gov/about-our-agency>
- [12] Jean Meeus. En.wikipedia.org [online]. [cit. 2020-02-24]. Dostupné z: [https://en.wikipedia.org/wiki/Jean\\_Meeus#cite\\_note-MAM-2](https://en.wikipedia.org/wiki/Jean_Meeus#cite_note-MAM-2)
- [13] Declination Angle. Pveducation.org [online]. 2019 [cit. 2020-03-02]. Dostupné z: <https://www.pveducation.org/pvcdrom/properties-of-sunlight/declination-angle>
- [14] Elevation Angle. Pveducation.org [online]. 2019 [cit. 2020-03-02]. Dostupné z: <https://www.pveducation.org/pvcdrom/properties-of-sunlight/elevation-angle>
- [15] Azimuth Angle. Pveducation.org [online]. 2019 [cit. 2020-03-02]. Dostupné z: <https://www.pveducation.org/pvcdrom/properties-of-sunlight/azimuth-angle>
- [16] Raspberry Pi For Dummies [eBook]. 3rd. 111 River Street, Hoboken, NJ 07030-5774,; John Wiley & Sons, Inc., 2017 [cit. 2020-02-26]. ISBN 978-1-119-41200-7.
- [17] 10 nejlepších využití pro Raspberry Pi. Nej10.cz [online]. [cit. 2020-02-26]. Dostupné z: <https://www.nej10.cz/10-nejlepsich-vyuziti-pro-raspberry-pi/>
- [18] UPTON, Eben a Gareth HALFACREE. Raspberry Pi: uživatelská příručka. Brno: Computer Press, 2013. ISBN 978-802-5141-168.
- [19] VÍTEK, Jan. Windows 10 na Raspberry Pi snadno a rychle. Svethardware.cz [online]. 13.2.2019 [cit. 2020-02-26]. Dostupné z: <https://www.svethardware.cz/windows-10-na-raspberry-pi-snadno-a-rychle/48680>
- [20] Raspberry Pi Zero WH. Rpishop.cz [online]. [cit. 2020-02-27]. Dostupné z: <https://rpishop.cz/raspberry-pi-zero/685-raspberry-pi-zero-wh-4250236816296.html>
- [21] HATTERSLEY, Lucy. Raspberry Pi 4, 3A+, Zero W - specs, benchmarks & thermal tests. MagPi.raspberrypi.org [online]. Raspberry Pi Trading, 2019 [cit. 2020-03-02]. Dostupné z: <https://magpi.raspberrypi.org/articles/raspberry-pi-specs-benchmarks>
- [22] Motor DC 12V 14RPM S30K s Převodovkou Vysoký točivý Moment. Arduino-shop.cz: IOT Výstupní periférie [online]. [cit. 2020-02-27]. Dostupné z: <https://arduino-shop.cz/arduino/1506-motor-dc-12v-14rpm-s30k-s-prevodovkou-vysoky-tocivy-moment.html>

- [23] Downloads. Raspberrypi.org [online]. [cit. 2020-02-28]. Dostupné z: <https://www.raspberrypi.org/downloads/>
- [24] Raspberry Pi Starter Kit Lesson 2: Introduction of Raspberry Pi GPIO. OSOYOO.com [online]. 26.6.2017 [cit. 2020-03-05]. Dostupné z: <https://osoyoo.com/2017/06/26/introduction-of-raspberry-pi-gpio/>
- [25] 4 Channel 5V Relay Module. Sunfounder.cc [online]. 20.3.2017 [cit. 2020-03-05]. Dostupné z: [http://wiki.sunfounder.cc/index.php?title=4\\_Channel\\_5V\\_Relay\\_Module](http://wiki.sunfounder.cc/index.php?title=4_Channel_5V_Relay_Module)
- [26] MÖHWALD, Albert. Měření radioaktivity snímáním pulsů GM trubice prostřednictvím Raspberry Pi [online]. Praha, 2016 [cit. 2020-03-23]. Dostupné z: [http://stretch.fs.cvut.cz/2016/sbornik\\_2016/a-CD-stretch2016/pRAHA-zBOROVSK%C3%81\\_Mohwald.pdf](http://stretch.fs.cvut.cz/2016/sbornik_2016/a-CD-stretch2016/pRAHA-zBOROVSK%C3%81_Mohwald.pdf) . Gymnázium Christiana Dopplera.
- [27] Graphing Pretty Charts With Python Flask and Chartjs. Ruan Bekker's Blog: From a Curious mind to Posts on Github [online]. 14.12.2017 [cit. 2020-04-15]. Dostupné z: <https://blog.ruanbekker.com/blog/2017/12/14/graphing-pretty-charts-with-python-flask-and-chartjs/>
- [28] Chart.js. Chartjs.org [online]. [cit. 2020-04-22]. Dostupné z: <https://www.chartjs.org/docs/latest/#installation>
- [29] NOAA Solar Calculator: Find Sunrise, Sunset, Solar Noon and Solar Position for Any Place on Earth. Global Monitoring Laboratory: Earth System Research Laboratories [online]. [cit. 2020-05-19]. Dostupné z: <https://www.esrl.noaa.gov/gmd/grad/solcalc/>
- [30] SunCalc. AGAFONKIN, Vladimir. Suncalc.net [online]. 2009 [cit. 2020-05-22]. Dostupné z: <http://suncalc.net>

## Seznam obrázků

Obrázek 1: Fotovoltaický jev [4] .....	5
Obrázek 2: Pozice slunce 2 [1] .....	6
Obrázek 3: Pozice slunce [1] .....	6
Obrázek 4: Deklinace Slunce [13] .....	9
Obrázek 5: Elevace slunce [14] .....	10
Obrázek 6: Azimut [15] .....	10
Obrázek 7: Porovnání spotřeby RPi [21] .....	13
Obrázek 8: Raspberry Pi GPIO piny [24] .....	16
Obrázek 9: Zapojení relé-modulu .....	17
Obrázek 10: Vnitřní zapojení reléového modulu [25] .....	18
Obrázek 11: Schéma relé logiky .....	19
Obrázek 12: Zapojení anemometru .....	20
Obrázek 13: Zapojení elektroměru .....	20
Obrázek 14: Zapojení RTC .....	20
Obrázek 15: Princip řízení .....	21
Obrázek 16: Hlídání větru .....	23
Obrázek 17: Model-pohled zepředu .....	31
Obrázek 18: Model-pohled zdola a z boku .....	32
Obrázek 19: Převod .....	33
Obrázek 20: Architektura webové aplikace .....	33
Obrázek 21: Ukázka webové aplikace .....	35
Obrázek 22: Výroba energie bez natáčení panelu .....	37
Obrázek 23: Výroba energie s natáčením panelu .....	37

## Seznam příloh

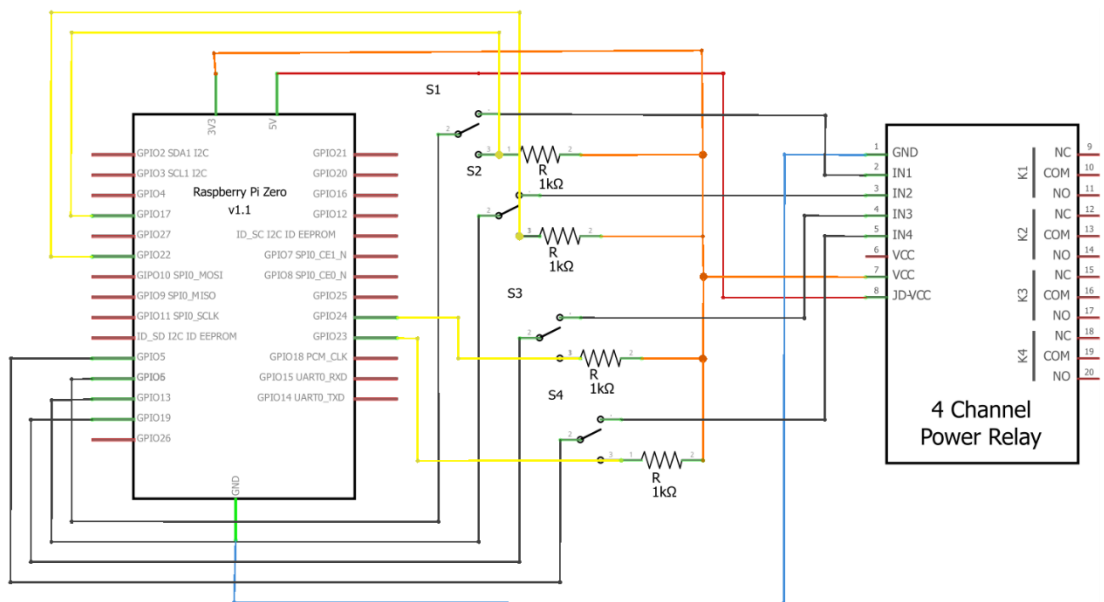
A Schéma zapojení .....	43
B Součástky modelu .....	44
C Fotografie zařízení .....	46
D Zdrojové kódy .....	48



# Přílohy

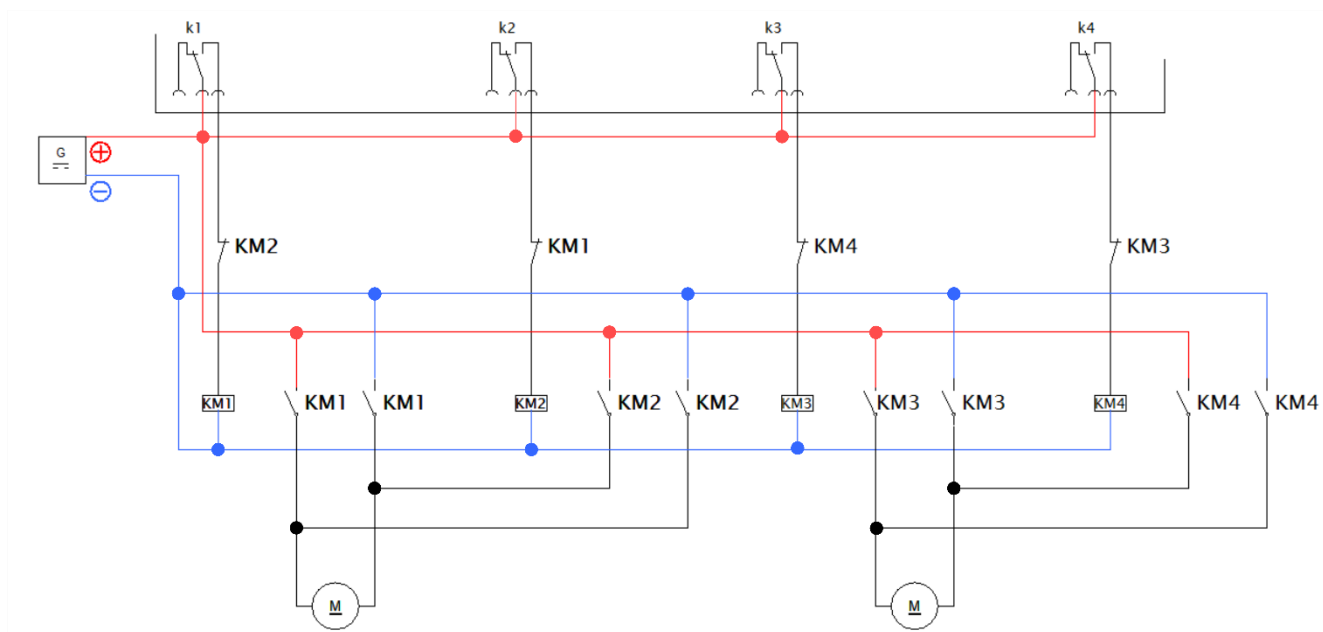
## Příloha A

### Schéma zapojení



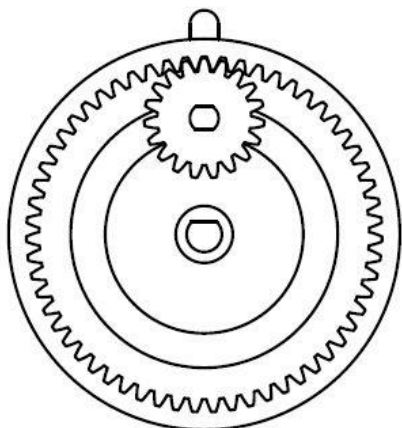
fritzing

zapojení relé modulu

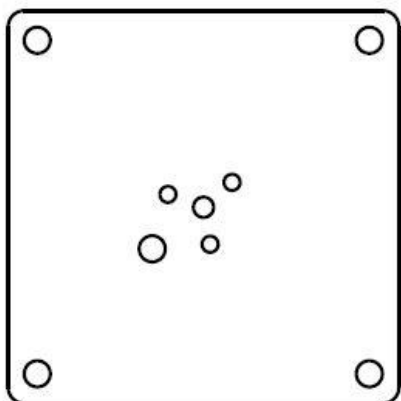


zapojení relé

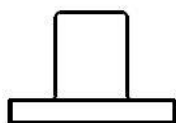
Příloha B  
Součástky modelu



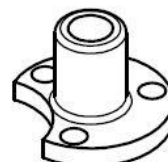
*Převod*

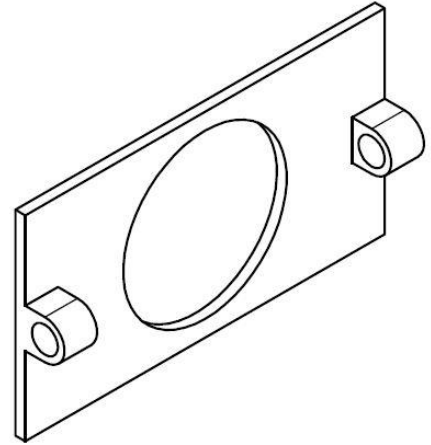
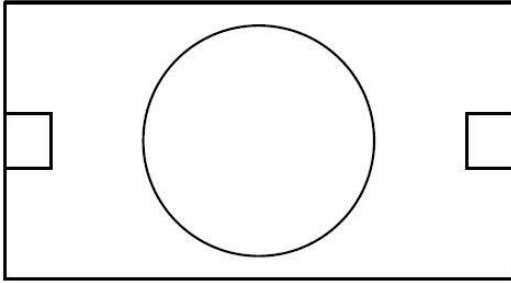
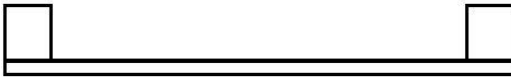


*Základna*

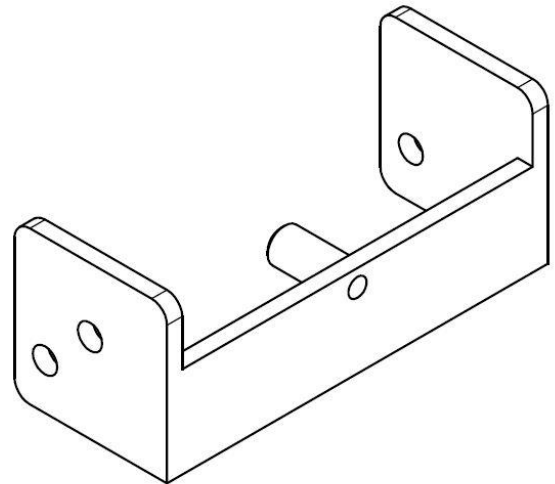
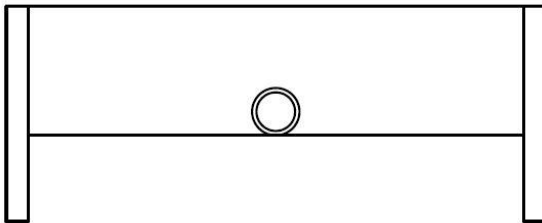
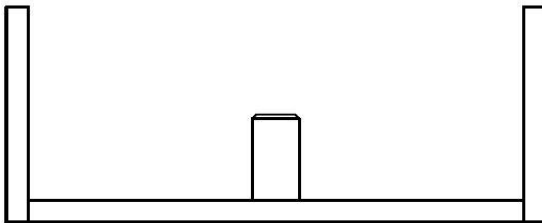
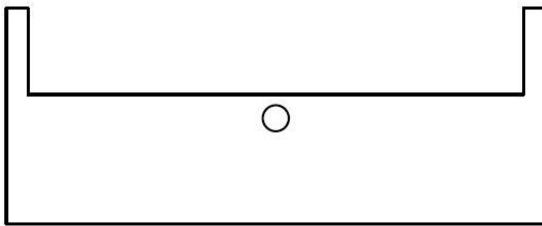


*Futro*





*Držák panelu*

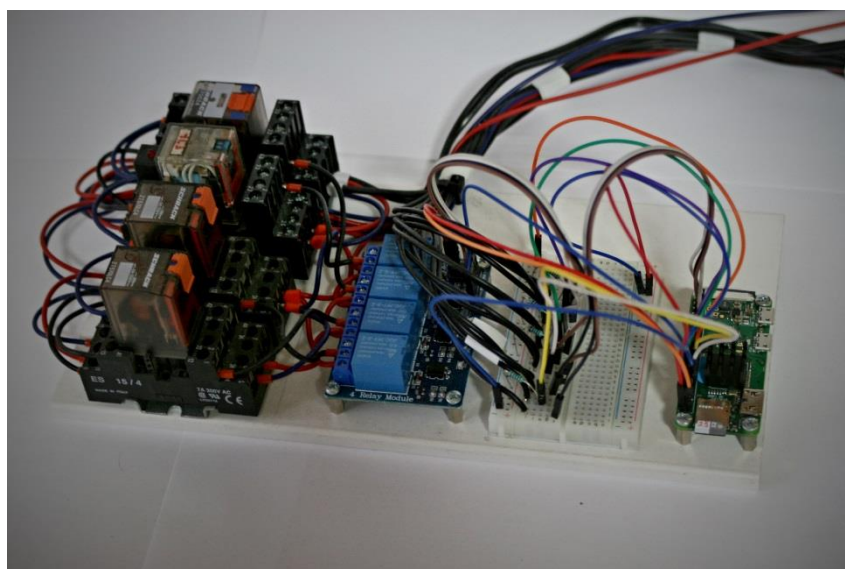


*Konstrukce-U*

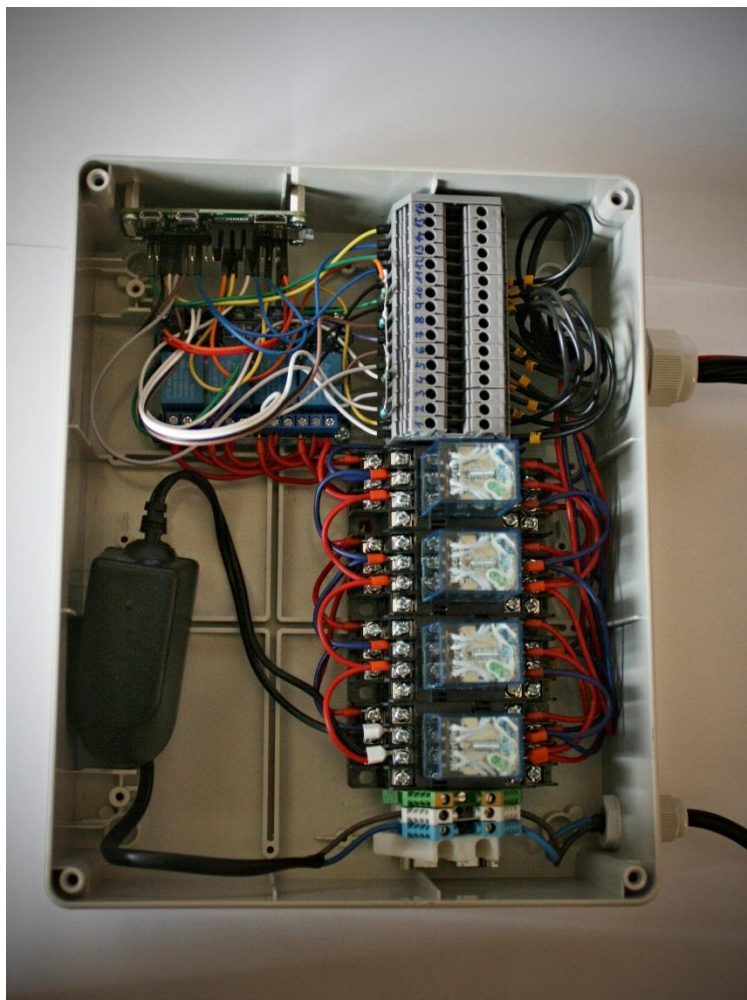
Příloha C  
Fotografie modelu a řídicí jednotky



*Model*



*Prototyp řídicí jednotky*



*Řídicí jednotka-vnitřek*



*Řídicí jednotka-krabice*

## Příloha D

### Zdrojové kódy

Kompletní zdrojové kódy jsou umístěny ve veřejném repozitáři na GitHubu.

Řídicí systém:

<https://github.com/stropak/solar-tracking>

Webová aplikace:

<https://github.com/stropak/web-app>