

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ, МОЛОДІ ТА СПОРТУ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ “КИЇВСЬКИЙ
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Факультет електротехніки та автоматики

Кафедра відновлюваних джерел енергії

До захисту допущено
Завідувач кафедри
_____ С.О.Кудря
“ ” _____ 2019р.

ДИПЛОМНА РОБОТА
На здобуття ступеня бакалавра

до дипломного проекту освітньо-кваліфікаційного рівня «бакалавр» за напрямком підготовки
6.050701 – "Електротехніка та електротехнології"
на тему: «Вимірювальна система для визначення характеристик інверторів в складі
фотоелектричної станції на базі мікрокомп'ютера Raspberry Pi 3»

Студент групи	<u>ЕД-51</u>	_____	<u>Ленський Денис Вікторович</u>
	(шифр групи)	(підпис)	(ім'я, по-батькові, прізвище)
Керівник роботи			<u>д-р. ф.-м. наук, проф. Гаєвський</u>
			<u>О.Ю.</u>
		(підпис)	(вчена ступінь, вчене звання, ініціали, прізвище)
Рецензент		_____	_____
		(підпис)	(вчена ступінь, вчене звання, ініціали, прізвище)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без відповідних
посилань.

Студент _____
(підпис)

Київ - 2019

ПОЯСНЮВАЛЬНА ЗАПИСКА

До дипломної роботи

**На тему: «Вимірювальна система для визначення характеристик
інверторів в складі фотоелектричної станції на базі мікрокомп'ютера
Raspberry Pi 3»**

Київ – 2019

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет Електроенерготехніки та Автоматики
Кафедра Відновлюваних Джерел Енергії
Рівень вищої освіти – перший (бакалаврський)
Спеціальність 6.050701 «Електротехніка та електротехнології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Кудря С. О.

«___»_____2019 р.

Завдання
на дипломну роботу студенту

Ленському Денису Вікторовичу

1. Тема проекту: «Вимірювальна система для визначення характеристик інверторів в складі фотоелектричної станції на базі мікрокомп'ютера Raspberry Pi 3»

керівник проекту докт. ф.-м. наук. проф. Гаєвський Олександр Юлійович
затверджені наказом по університету від «___»_____2010 р. №_____

2. Строк подання студентом проекту «___»_____2019 р.

3. Вихідні дані до роботи : _____.

4. Зміст пояснювальної записки

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1	Розділ 1 «Архітектура вимірювальної системи для визначення характеристик ФЕС»	10.05.2019-25.05.2019	
2	Розділ 2 «Робота вимірювальної системи та її інтерфейс»	26.05.2019-10.06.2019	
3	Розділ 3 «Охорона праці»	11.06.2019-16.06.2019	
4	Загальні висновки	17.06.2019-18.06.2019	

Студент

Ленський Д.В.

Керівник проекту

Гасвський О.Ю.

Анотація

Метою дипломної роботи є створення і налаштування платформи для збору, обробки, відображення даних та забезпечення віддаленого доступу до них. Завдання дипломної роботи полягало у створенні вимірювальної системи з програмним управлінням для моніторингу характеристик обладнання ФЕС, яка зможе відделно збирати, обробляти, надсилати та відображати дані на ПК користувача. Ці задачі актуальні для тестування та діагностики поточного стану обладнання ФЕС, для аналізу фактичних значень інверторів та фотомодулів в складі фотоелектричної станції.

Для визначення характеристик обладнання ФЕС була розроблена програма на базі одноплатного мікрокомп'ютера Raspberry Pi 3, який, в свою чергу, виконував роль збору, обробки та передачі даних по інтернету до ПК користувача, де він міг відобразити ці дані.

У даній дипломній роботі реалізована система для визначення характеристик інверторів та фотомодулів за допомогою розробленого програмного забезпечення та вимірювального пристрою.

Пояснювальна записка містить 72 сторінок машинописного тексту формату А4, 18 рисунків, 1 таблицю і 28 найменувань за переліком посилань.

Ключові слова: фотомодуль, аналогово-цифровий перетворювач, сонячна енергетика, програма, програмне забезпечення, вимірювальна система, фотоелектрична станція, мікрокомп'ютер.

Annotation

The purpose of the graduate work is to create and configure a platform for collecting, processing, displaying data and providing remote access to them. The task of the graduate work was to create a measurement system with software for monitoring the characteristics of the FES equipment, which can separately collect, process, send and display data on the user's PC. These tasks are relevant for the testing and diagnostics of the current state of the FES equipment, for the analysis of the actual values of inverters and photomodules in the photoelectric station.

To determine the characteristics of the FES equipment, a program developed using the single-board microcomputer Raspberry Pi 3, which, in turn, performed the role of collecting, processing and transmitting data over the Internet to the user's PC, where he could display this data.

In this graduate work system of inverters and photomodules characteristics determination is implemented with the help of the developed software and measuring devices.

The explanatory note contains 72 pages of A4 text, 18 figures, 1 table and 28 references.

Keywords: photomodule, analog-digital converter, solar energy, program, software, measuring system, photovoltaic station, microcomputer.

ЗМІСТ

СПИСОК СКОРОЧЕНЬ.....	9
ВСТУП. ПОСТАНОВКА ЗАДАЧІ.....	10
РОЗДІЛ 1 АРХІТЕКТУРА ВИМІРЮВАЛЬНОЇ СИСТЕМИ ДЛЯ	
ВИЗНАЧЕННЯ ХАРАКТЕРИСТИК ОБЛАДНАННЯ ФЕС.....	13
1.1 Принцип роботи вимірювальної системи та її комплектуючі.....	13
1.2 Датчики та АЦП.....	13
1.3 Мікрокомп'ютер Raspberry Pi.....	14
1.4 Конструкційні особливості Raspberry Pi 3 Model B+.....	16
1.5 Запуск Raspberry Pi, використовуючи micro SD.....	18
1.6 Підключення Raspberry Pi до інтернету.....	19
1.7 Підключення по віддаленому доступу до Raspberry Pi через мережу з використанням SSH з'єднання та X11 forwarding.....	20
1.8 Збереження Raspberry Pi в актуальному стані.....	23
1.9 Спільний доступ до домашньої папки Raspberry Pi з SMB.....	23
РОЗДІЛ 2 РОБОТА ВИМІРЮВАЛЬНОЇ СИСТЕМИ ТА ЇЇ	
ІНТЕРФЕЙС.....	25
2.1 Інтерфейс програми для визначення та обробки характеристик сонячного обладнання.....	25
2.2 Програмний код вимірювальної системи.....	26
2.3 Зовнішній вигляд програми.....	45
РОЗДІЛ 3 ОХОРОНА ПРАЦІ. ОЦІНКА ВПЛИВУ НА НАВКОЛИШНЄ	
СЕРЕДОВИЩЕ.....	49
3.1 Аналіз шкідливих та небезпечних факторів під час використання обладнання фотоелектричної станції напругою 0,4 кВ.....	49
3.2 Заходи щодо приведення умов праці в норму.....	52
3.3 Пожежна безпека.....	59
3.4 Оцінка впливу на навколишнє середовище.....	65

3.5	Можливі впливи при будівництві та експлуатації сонячної станції на навколишнє середовище.....	67
3.6	Оцінка небезпеки під час користування фотоелектричним обладнанням.....	68
	ВИСНОВКИ ТА РЕЗУЛЬТАТИ РОБОТИ.....	69
	СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	70

СПИСОК СКОРОЧЕНЬ

1. МК – мікрокомп'ютер
2. ФЕС – фотоелектрична станція
3. ПК – персональний комп'ютер
4. АЦП – аналогово-цифровий перетворювач
5. ПЗ — програмне забезпечення

ВСТУП. ПОСТАНОВКА ЗАДАЧІ

Скорочення використання традиційних викопних палив диктується не тільки їх обмеженими запасами, але і вимогами щодо зменшення викидів в атмосферу парникових газів. Одним з найбільш перспективних напрямів розвитку світової енергетики в даний час є використання відновлюваних джерел енергії, що знімає ряд проблем, пов'язаних з використанням традиційних палив. Кінець другого тисячоліття характеризується інтенсивним ростом обсягів використання енергії відновлюваних джерел енергії у світі.

Сонце є найбільш ефективним поновлюваним джерелом енергії на нашій планеті. Сонячна енергія – це енергія, отримана від сонця за рахунок використання сонячних батарей. Фотоелектричні елементи в сонячній панелі перетворюють світло в електричний струм, використовуючи фотоелектричний ефект. Концентровані ФЕС використовують лінзи або дзеркала і системи трекінгу, щоб зконцентрувати велику кількість сонячного світла в невеликий пучок. Сонячна панель стрімко перетворюється на недорогу низьковуглецеву технологію для отримання відновлюваної енергії від сонця. Міжнародне Енергетичне Агенство прогнозувало в 2014 році, що до 2050 р сонячна енергія буде складати близько 16% енергії світу[1].

Багато промислових країн встановили велику кількість фотоелектростанцій у своїх мережах, щоб забезпечувати населення альтернативними джерелами енергії замість звичайних. Імпульсний стабілізатор напруги надає переваги з точки зору розміру, ваги, вартості, ефективності та загальної продуктивності. Вони стали прийнятою частиною сучасних електронних гаджетів. Основна концепція імпульсного стабілізатора напруги або SMPS є те, що регулювання здійснюється за допомогою комутації регулятора. При цьому використовується послідовний перемикаючий елемент, що перетворює подачу струму на згладжуючий включений і вимкнений конденсатор. В основному, це пристрій, в якому перетворення енергії і регулювання забезпечується силовими напівпровідниками, які безперервно включаються і вимикаються з високою частотою. У результаті висока ефективність і низький рівень розсіювання тепла[2].

Використання сонячної енергії є найбільш яскравою і екологічною ідеєю живлення сучасних гаджетів і систем. Беручи до уваги цю ідею, імпульсний стабілізатор напруги (SMPS) розроблений для ФЕС, повністю працює на сонячній енергії і так і називається - сонячний SMPS. Правильне і безперебійне постачання електроенергії неможливе в різних частинах світу. Особливо такі країни, як Індія мають велику проблему з електроенергією, де велика її кількість виробляється традиційними методами. Є великі можливості для сонячної енергії в Індії. На сьогоднішній день сонячні енергетичні технології стали більш просунутими, за рахунок чого ми маємо можливість використовувати енергію, яку ми отримуємо від сонця більшу у ще більшій кількості. Основна мета таких систем полягає у збереженні електроенергії та максимальному використанні відновлюваних джерел енергії[3].

Що стосується України, то потенціал використання сонячної енергії в Україні є достатньо високим для широкого впровадження як теплоенергетичного, так і фотоенергетичного обладнання. Середньорічна кількість сумарної сонячної енергії, що надходить на 1 кв. кілометр поверхні території України, становить в межах 1070 кВт·год у північній частині країни та 1400 кВт·год і вище в південній. Для подальшого розвитку та масового виробництва сонячних фотоелементів розроблені модулі батарей на основі напівпровідникового кремнію[4].

Сонячна енергетика в Україні поки не набула широкого господарського використання, проте передумови для цього є. Вона здатна забезпечити економію за рік до 6 млн. тонн умовного палива, потенціал її розвитку становить власна наукова і промислова база, конструкторські бюро, що проектують сонячні колектори, виробництво моно- і полікремнію, нанотехнології, необхідна металопродукція тощо. Проекти фотоенергетики активно впроваджуються з 2010 року. Загальна потужність промислових сонячних станцій, яка вимірюється від сотень кіловат до десятків мегават, зросла з 2015 року по перший квартал 2018 року більше ніж 560 МВт. А інвестиції в економіку України приватного капіталу склали біля 550 мільйонів євро. Це важливо іще й з огляду створення нових робочих місць, окрім того, до місцевих бюджетів надходять податки[5].

Наразі потреба у фотоелектричних станціях зросла у декілька разів, оскільки тарифи на електроенергію стали дуже великими. Велика кількість домогосподарств вже використовують сонячні панелі для економії електроенергії та щоб зменшити використання традиційних викопних палив та, тим самим, покращити екологічний стан планети. Крім того, зелений тариф стимулює приватних осіб до за купування сонячних батарей, адже це хороший спосіб для пасивного заробітку за кошт держави, а їй, в свою чергу, вигідно платити за надану енергію. Але на скільки вигідно буде для приватних господарств встановлювати сонячні панелі після закінчення дії зеленого тарифу і чи продовжиться ця тенденція[6]?

На сьогодні ми бачимо чітку тенденцію до зменшення вартості панелей, інверторів та системних комплектуючих. Разом з тим ми спостерігаємо значне зменшення електроспоживання різних приладів. Сьогодні люди звертають увагу на те скільки споживає той чи інший прилад. Все більше стають популярними електромобілі. Тому очевидно, що ці фактори в перспективі приведуть до того, що буде цілком вигідно встановлювати сонячну електростанцію для покриття власних потреб домогосподарств, підприємств, а не лише просто для заробітку. Проте далеко не всі питання в галузі сонячної енергетики вирішені[7].

Всі сучасні фотоелектричні станції оснащуються системами моніторингу. Автономні ФЕС ідеальні для віддалених від мереж населених пунктів. Однак, для того, щоб впевнитись, що автономна система здатна безперервно

забезпечувати високу продуктивність, її часто потрібно контролювати та тестувати. Але не завжди зручно це робити безпосередньо на самій станції. Через це актуальним є створення систем віртуального моніторингу (збір даних і передача їх через мережу). Ідея відображення отриманих даних через датчики на сайті в Інтернеті або на сервері стають досить популярними завдяки до малопотужних електронних пристроїв і різних методів безпроводного зв'язку. Це стає цілком можливим завдяки малопотужним електронним пристроям. Однак, більшість комерційних рішень віртуального моніторингу, які присутні на ринку, зазвичай дуже дорогі та не завжди відповідають задачам та потребам користувача[8]. Оскільки повне готове рішення даної проблеми настільки об'ємне та обширне, що його реалізацією займаються колективи досвічених розробників, в даній роботі були поставлені задачі дещо меншого масштабу:

- розробка бюджетної системи моніторингу на базі мікрокомп'ютера (виконує роль збору, обробки даних та передачу їх через мережу інтернет)
- створення і налаштування платформи для збору, обробки, відображення даних та забезпечення віддаленого доступу до них
- створення окремих блоків вимірювальної системи на базі мікрокомп'ютера Raspberry Pi 3 для визначення характеристик обладнання ФЕС.

РОЗДІЛ 1

АРХІТЕКТУРА ВИМІРЮВАЛЬНОЇ СИСТЕМИ ДЛЯ ВИЗНАЧЕННЯ ХАРАКТЕРИСТИК ОБЛАДНАННЯ ФЕС

1.1 Принцип роботи вимірювальної системи та її комплектуючі

На рисунку 1.1 ви можете побачити блок-схему вимірювальної системи для визначення характеристик обладнання ФЕС, яка розроблялась у даній роботі. Завдяки PV-джерелам та датчикам температури і радіації зніматимуться аналогові значення, які подаються на входи аналогово-цифрового перетворювача. Він конвертує аналогові значення в цифрові і передає їх через піни на мікрокомп'ютер. Він, у свою чергу, збирає і обробляє дані та, використовуючи можливості вбудованого у мікрокомп'ютер Wi-Fi модуля, Raspberry Pi надсилає їх через веб-сервер на персональний комп'ютер користувача, де він може графічно їх відобразити та проаналізувати за допомогою розробленого ПЗ для обробки даних та інтерфейсу.

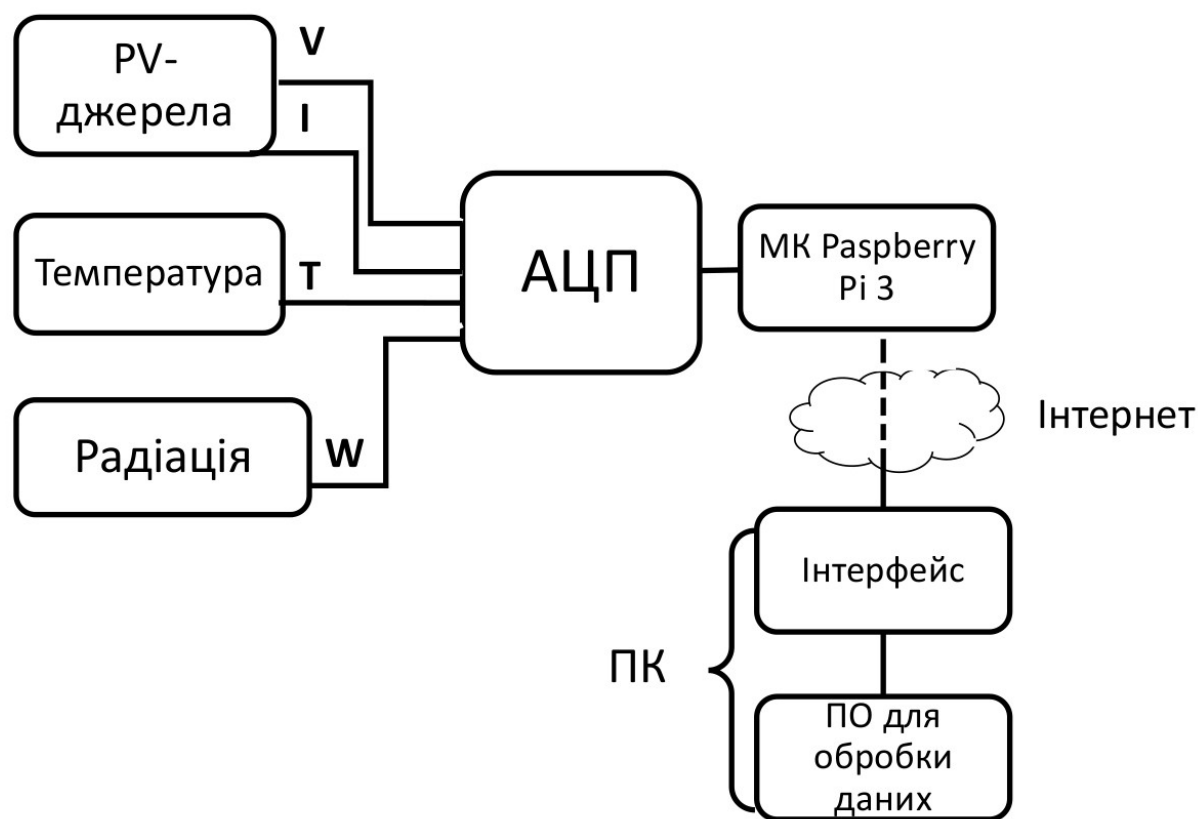


Рисунок 1.1. Блок-схема вимірювальної системи віддаленого доступу

1.2. Датчики та АЦП

Для отримання аналогових значень характеристик обладнання ФЕС я використовував датчики радіації ВН1750 (рис. 1.2) та температури DHT11 (рис 1.3). Напрягу та струм фотомодулів та інверторів можна виміряти за допомогою дільника напруги в схемі.

Оскільки Raspberry Pi 3 не має аналогових входів, цю задачу буде брати на себе 4-канальний 16-розрядний АЦП ADS1115. Raspberry Pi 3 був обраний як мікрокомп'ютер для збору та обробки даних[9] з 4-х параметрів через 4 канали АЦП: струму, напруги, температури навколишнього середовища, радіації. Raspberry Pi був вибраний здебільшого завдяки тому, що він:

- має відносно низьку ціну
- малий та компактний у розмірах
- продуктивність на тому ж рівні, що у комп'ютерів 2009 року
- має вбудований Wi-Fi модуль

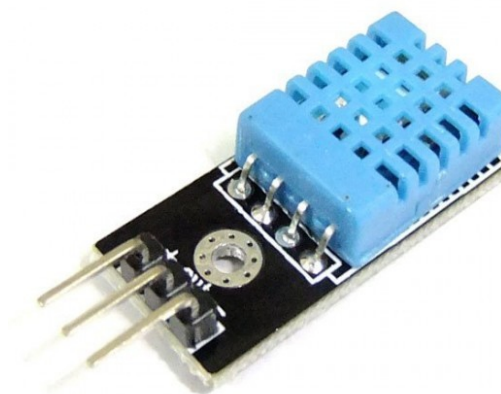


Рисунок 1.2. Датчик радіації BH1750 Рисунок 1.3. Датчик температури DHT11

1.3 Мікрокомп'ютер Raspberry Pi

З моменту виходу комп'ютера Raspberry Pi в лютому 2012 року сотні тисяч людей знайшли новий спосіб програмувати. Сучасні домашні комп'ютери, планшети і телефони, як правило, орієнтовані на надання контенту користувачеві або пасивно, або через ігри та іншу активну діяльність.

Тим не менш, Raspberry Pi повністю перетворює цю концепцію. Ідея полягає в тому, що користувач використовує уяву, а Raspberry Pi стає продовженням їхньої творчості. Raspberry Pi надає просту, недорогу платформу, за допомогою якої можна експериментувати і грати з власними ідеями.

Raspberry Pi - це одноплатний комп'ютер, який ви можете побачити на рисунку 1.4, створений Фондом Raspberry Pi, фонд, сформований основною метою відновлення низького рівня комп'ютерних навичок для дітей у Великобританії. Мета полягала в тому, щоб відродити революцію мікрокомп'ютера з 1980-х років, які дали ціле покоління досвідчених програмістів. Ще до виходу комп'ютера в кінці лютого 2012 року стало зрозуміло, що Raspberry Pi отримала величезну прихильність по всьому світу.

Назва Raspberry Pi було поєднанням бажання створити альтернативну назву комп'ютера (наприклад, Apple, BlackBerry, Apricot) і прагнення оригінальної концепції простого комп'ютера, який можна програмувати за допомогою Python (скорочено Pi).



Рисунок 1.4. Мікрокомп'ютер Raspberry Pi 3 Model B+

Часто запитують: “Чому Python був обраний мовою програмування на Raspberry?”. Річ у тому, що Python є лише одним з багатьох мов програмування, які можна використовувати на Raspberry Pi.

Є багато мов програмування, які можна вибрати, починаючи з високого рівня програмування, наприклад, Scratch, ідучи до традиційного C, аж до BASIC, і закінчуючи машинним кодом Асемблеру. Хороший програміст часто повинен володіти багатьма мовами, щоб мати знати можливості та сильні і слабкі сторони кожної мови, щоб найкращим чином задовольнити потреби бажаної програми. Корисно зрозуміти, як різняться різні мови і техніки програмування намагатися подолати виклик перетворення "що ви хочете" в "те, що ви отримуєте" тому що це те, що ви намагаєтеся зробити пишучи код.

Python був вибраний як проста мова для початку навчання програмувати, вона забезпечує багатий набір інструментів програмування, дозволяє писати складні програми без зайвої метушні. Це дозволяє новачкам поступово дізнаватися про поняття і методи, на яких сучасні мови програмування базуються без необхідності знати все це з самого початку. Це дуже функціональна мова з великою кількістю додаткових бібліотек, які можна швидко імпортувати.

Як і всі мови програмування, Python не ідеальний; такі речі, як додавання пробілу на початок рядка часто порушують ваш код (відступ має велике

значення в Python; він визначає як згруповані блоки коду). Як правило, Python повільний; оскільки він інтерпретується, потрібен час для створення модулів під час запуску програми на мові Python. Це може бути проблемою, якщо вам потрібно негайно реагувати на критичний для подій час. Тим не менш, ви можете попередньо скомпілювати Python або використовувати модулі, написані іншими мовами, щоб подолати це. Він приховує деталі; це є одночасно і перевагою і недоліком. Це добре підходить для початківців, але може бути важко, коли вам доведеться вгадувати такі аспекти, як типи даних, але це в свою чергу змушує вас розглянути всі можливі помилки, що в свою чергу, розвиває ваші навички[10].

Перш ніж використовувати Raspberry Pi, вам знадобиться карта SD з операційною системою або з готовою операційною системою (NOOBS) на ній. Далі буде детально описано типи пристроїв, які можна підключити до Raspberry Pi і головне, як і де їх підключити. Як ви дізнаєтеся пізніше, коли ви запустите Raspberry Pi, ви зможете підключитися дистанційно до нього та використовувати його через мережеве посилання, в цьому випадку вам потрібні лише живлення та мережа з'єднання.

1.4 Конструкційні особливості Raspberry Pi 3 Model B+

Порти, які має Raspberry Pi:

- Дисплей:

Raspberry Pi підтримує наступні два підключення основного дисплея: HDMI та аналоговий вхід. Для досягнення найкращих результатів використовуйте телевизор або монітор, який має з'єднання HDMI, вони відображатимуть дозвіл (1080p), а також забезпечують цифровий аудіовихід. Існує кілька типів з'єднань DVI; деякі підтримують аналог (DVI-A), деякі цифрові (DVI-D), а деякі обидва (DVI-I). Raspberry Pi здатний тільки забезпечити цифровий сигнал через HDMI, так рекомендується використовувати адаптер HDMI до DVI-D (рисунок 1.5). Цьому не вистачає чотирьох додаткових аналогових контактів, таким чином дозволяючи йому вписатися в обидва DVI-D і роз'єми типу DVI-I.



Рисунок 1.5. Підключення HDMI до DVI (адаптер DVI-D)

Якщо ви хочете використовувати старий монітор (з VGA-підключенням), додатково необхідний перетворювач HDMI в VGA. Альтернативний спосіб

відображення полягає у використанні аналогового композитного відео підключення (через роз'єм RCA); це також може бути приєднане до S-Video або європейський адаптер SCART. Однак аналоговий відеовихід має максимальну роздільну здатність - 640 x 480 пікселів, тому він не ідеальний для загального використання. При використанні підключення RCA або входу DVI аудіо має бути забезпечення аналоговим аудіозв'язком.

- Ethernet (тільки модель B):

Якщо ми використовуємо модель Raspberry Pi, можна підключитися до інтернету, використовуючи кабельне підключення, або ж підключитися через Wi-Fi через usb адаптер.

- USB (x1 модель A, x2 B) – для клавіатури та миші:

Raspberry Pi має працювати з більшістю USB клавіатур і мишей. Однак, Ви можете зіткнутися з проблемами, якщо ваш пристрій потребує більше 140 мА, у такому випадку рекомендується використовувати USB-концентратор. Можна також використовувати бездротові миші та клавіатури, які використовують RF Dongles. У разі відсутності живлення, що надходить від джерела живлення, або пристрої поглинають великий струм, може пошкодитися SD карта.

Потужність USB може бути більшою проблемою з ранньою версією моделі B 1 плати, які були доступні до жовтня 2012 року. Вони включали додаткові поліф'юзи на виході USB і спрацьовували, якщо було поглинуто 140 мА. Поліф'юзи можуть зайняти кілька годин або днів, щоб повністю відновитися, таким чином викликаючи непередбачувану поведінку, навіть коли поліпшується ситуація з потужністю.

Debian Linux (на якому базується Raspbian) підтримує багато загальних пристроїв USB, такі як флеш-накопичувачі, жорсткі диски (може знадобитися зовнішнє живлення), камери, принтери, Bluetooth і адаптери Wi-Fi. Деякі пристрої будуть виявлені автоматично, тоді як інші вимагають встановлення драйверів.

- Micro USB живлення:

Raspberry Pi вимагає живлення 5В, який може давати принаймні 700 мА (рекомендовано 1А або більше) з мікро USB з'єднання. Можливе живлення пристрою за допомогою портативних акумуляторних батарей, таких як ті, які підходять для живлення або підзарядки смартфонів. Знову ж таки, переконайтеся, що вони можуть забезпечити 5В при 700 мА або більше. Ви повинні робити всі інші з'єднання з Raspberry Pi перед підключенням живлення. Однак пристрій USB, аудіо та кабель Ethernet можуть бути підключені та вилучені під час роботи мікрокомп'ютера він працює без проблем[11].

Окрім стандартних основних з'єднань, які ви очікуєте побачити на комп'ютері, Raspberry Pi також має ряд додаткових з'єднань.

Вторинні апаратні з'єднання

Кожне з наведених нижче з'єднань надає додаткові інтерфейси для Raspberry Pi:

- P1 13 x 2 піни GPIO: це головні піни GPIO, які використовуються Raspberry Pi для взаємодії безпосередньо з апаратними компонентами.
- P5 8 x2 піни GPIO: вони присутні тільки на платі версії 2.0 (не встановлені штирі).
- P6 reset: цей параметр присутній тільки на платі версії 2.0 (не встановлено жодних контактів). Reset не спрацьовує, коли P6-Pin1 (reset) і P6-Pin2 (GND) з'єднані разом.
- P5 і P6 піни: ми використовуємо P5 і P6 у кнопці керованого вимикання.
- P2/P3 GPU/LAN JTAG: спільна група дій з тестування (JTAG) - це програмований інтерфейс дебагінгу, який використовується для налаштування та тестування процесорів. Необхідний спеціальний пристрій JTAG для використання цього інтерфейсу.
- S5 Direct Camera CSI: це підключення підтримує модуль камери Raspberry Pi
- S2 Direct Display DSI: це з'єднання призначене для підтримки дисплея (адд-он, випущений Raspberry Pi Foundation)[12].

1.5 Запуск Raspberry Pi, використовуючи micro SD

Альтернативою використанню NOOBS є вручну записувати образ операційної системи на карту SD. Хоча спочатку це був єдиний спосіб встановити операційну систему, деякі користувачі все ще надають перевагу цьому способу встановлення операційної системи на Raspberry, у тому числі і я. Це дозволяє підготувати SD карту, перш ніж вона буде використовуватися в Raspberry Pi. Ця micro SD карта також може забезпечити більш легкий доступ до файлів запуску та конфігурації і залишає більше місця для користувача (на відміну від NOOBS, але розділ RECOVERY не включено). За замовчуванням зображення Raspbian складається з двох розділів, BOOT і SYSTEM, які будуть встановлені на 2-Гбайтну SD-карту (рекомендується 4 ГБ або більше). Вам потрібен комп'ютер під керуванням Windows / Mac OS X / Linux (хоча його можна використовувати іншу Raspberry Pi, щоб написати свою картку). Завантажте останню версію операційної системи, яку ви бажаєте використовувати. Виконайте наступні дії, залежно від типу комп'ютера, який плануєте використовувати для запису в SD-карта (файл .img, який вам потрібен, зазвичай стискається, тому перед початком роботи вилучити файл).

Наступні кроки стосуються Windows:

1. Переконайтеся, що завантажено зображення Raspbian, як було описано раніше і ви вилучили його в зручну папку, щоб отримати файл .img.
2. Завантажити Win32DiskImager.exe
3. Запустіть Win32DiskImager.exe
4. Натисніть на значок папки та перейдіть до розташування файлу .img і натисніть кнопку Зберегти.
5. Якщо ви ще цього не зробили, вставте картку SD у пристрій для читання карт і підключіть її на комп'ютер.

6. Виберіть маленьку літеру пристрою, що відповідає вашій SD-карті з випадкового списку. Переконайтеся, що це правильний пристрій (буде перезаписано все, що знаходиться на пристрої під час написання зображення). Літера диска може не відображатися, поки не буде вибрано джерело файлу зображення .img.
7. Нарешті, натисніть кнопку Записати і почекайте, поки програма напише зображення на флеш-картку.
8. Після завершення можна вийти з програми.
9. Тепер ви можете вставити флеш карту у Raspberry Pi і включати її[13].

1.6 Підключення Raspberry Pi до інтернету

Найпростіший спосіб підключення Raspberry Pi до Інтернету - використання вбудованого LAN підключення в моделі B. Якщо ви використовуєте модель Raspberry Pi A, ви можете використовувати адаптер USB-to-LAN. Вам знадобиться доступ до відповідної кабельної мережі, яка буде підключена до Інтернету і стандартний кабель (Cat5e або аналогічний з роз'ємом типу RJ45 для підключення до Raspberry Pi). Багато мереж підключаються та налаштовуються автоматично за допомогою DHCP протоколу, який контролюється маршрутизатором або комутатором. Якщо це так, просто підключіть кабель до мережевого порту на маршрутизаторі або мережевому комутаторі. Якщо ж сервер DHCP недоступний, вам доведеться налаштувати параметри вручну. Ви можете це зробити, виконавши такі дії:

1. Переконайтеся, що три світлодіоди на Raspberry Pi, позначені FDX, LNK і 100, загоряються (індикатор 100 може не світитися, якщо підключено до пристрою швидкістю до 10 Мбіт/с), а в деяких випадках починають блимати. Це покаже що є фізичне підключення до маршрутизатора, а обладнання живиться та функціонує.

2. Перевірте посилання на вашу локальну мережу за допомогою команди ping. По-перше, дізнайтеся IP адреси іншого комп'ютера в мережі (або, можливо, адреси вашого маршрутизатора, часто 192.168.0.1 або 192.168.1.254). Тепер, на терміналі Raspberry Pi, використовуйте команду ping (параметр -c 4 використовується для передачі лише чотирьох повідомлень; в іншому випадку натисніть Ctrl + C, щоб зупинити), щоб перевірити IP-адресу, як зазначено: ping 192.168.1.254 -c 4

3. Перевірте посилання з Інтернету (це не вдасться, якщо ви звичайно підключаєтеся до Інтернету проксі-сервера): ping www.raspberrypi.org -c 4

4. Нарешті, ви можете перевірити посилання назад на Raspberry Pi, відкривши IP-адресу за допомогою hostname -I на Raspberry Pi. Потім можна скористатися командою ping на інший комп'ютер у мережі, щоб переконатися, що він доступний (за допомогою Raspberry IP-адреса Pi замість www.raspberrypi.org). Версія Windows для команди ping виконає п'ять пінгів і зупиниться автоматично, опція -c 4 не потрібна). Якщо

вищевказані тести провалюються, вам потрібно перевірити свої з'єднання, а потім підтвердити правильність налаштування мережі.

1.7 Підключення по віддаленому доступу до Raspberry Pi через мережу з використанням SSH з'єднання та X11 forwarding

SSH (Secure SHell) часто є кращим методом для створення віддалених з'єднань, оскільки він дозволяє робити тільки термінальні з'єднання і зазвичай вимагає менших ресурсів. Додатковою особливістю SSH є можливість передачі даних X11 на сервер X Windows, який працює на вашому комп'ютері. Це дозволяє запускати програми, які зазвичай виконуються на робочому столі Raspberry Pi і вони з'являться у власній Windows системі на локальному комп'ютері, як показано на рисунку 1.6:

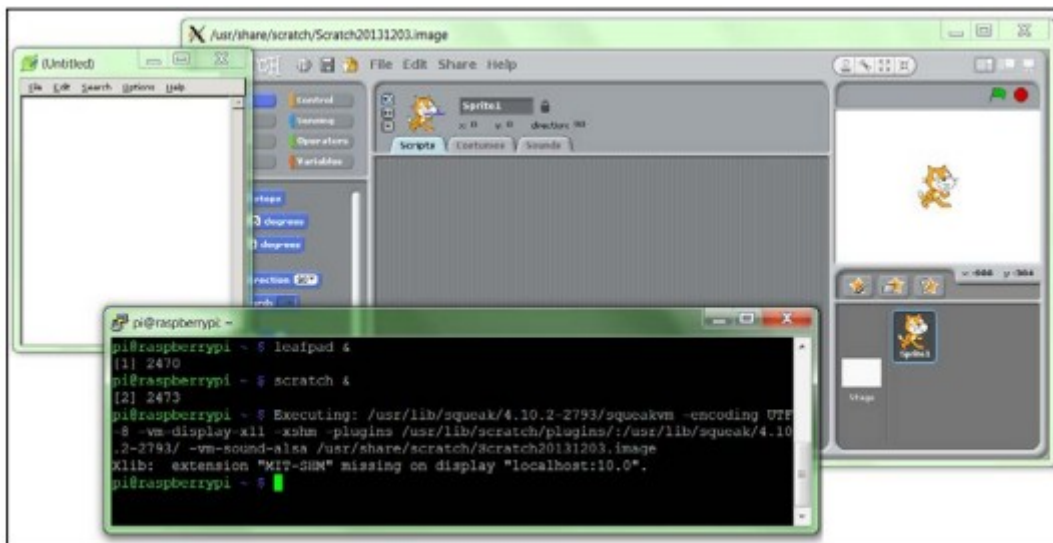


Рисунок 1.6. X Forwarding можна використовувати для відображення програм, які працюють на Raspberry Pi, на комп'ютер Windows.

Якщо ви використовуєте останню версію Raspbian, SSH і X11 Forwarding буде включено за замовчуванням. Linux і OS X мають вбудовану підтримку X11 Forwarding, але якщо ви використовуєте Windows, ви будете потрібно встановити та запустити сервер X Windows на вашому комп'ютері[14]. Завантажте та запустіть xming з сайту Xming. Встановіть xming, дотримуючись кроків встановлення, включаючи встановлення PuTTY, якщо він у вас ще не встановлений. Ви також можете завантажити PuTTY окремо. Далі, ми повинні забезпечити, щоб програма SSH, яку ми використовуємо, дозволила X11 під час підключення. Для Windows ми будемо використовувати PuTTY для підключення до Raspberry Pi. У діалоговому вікні PuTTY перейдіть до Connection | SSH | X11 і поставте галочку в прапорці для пересилання X11, як показано на рисунку 1.7. Якщо ви залишите параметр розташування дисплея X пустим, він прийме типовий сервер 0: 0 таким чином (номер сервера можна підтвердити,

перемістивши ваш курсор на значок Xming у системному треї, коли він запущений):

Введіть IP-адресу Raspberry Pi в налаштуваннях сеансу (ви також можете знайти, що замість цього можна використовувати ім'я хоста Raspberry Pi; ім'я хоста за замовчуванням – raspberrypi). Збережіть налаштування за допомогою відповідного імені RaspberryPi і натисніть кнопку Відкрити, щоб підключитися до вашого Raspberry Pi (рисунок 1.8). Ви, ймовірно, побачите попереджувальне повідомлення, що ви не підключилися до комп'ютера перед цим (це дозволяє перевірити, чи все правильно, перш ніж продовжити).

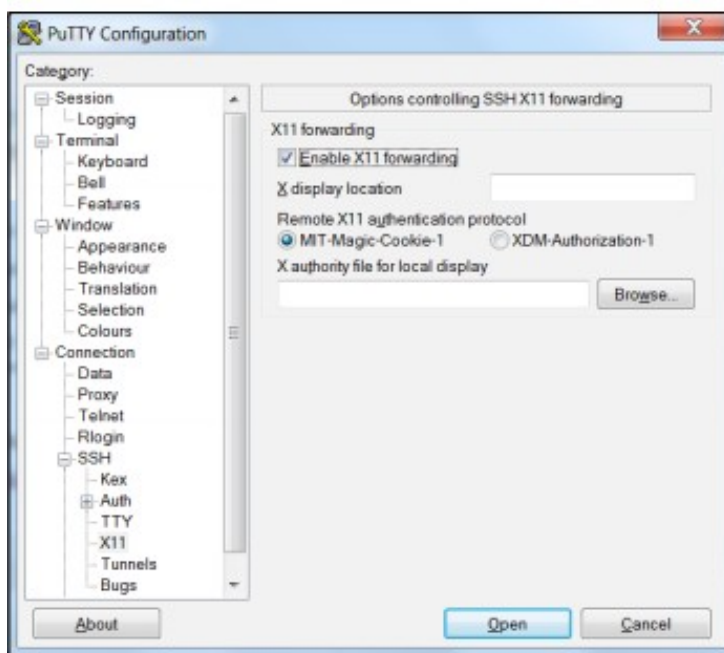


Рисунок 1.7. Увімкнення X11 Forwarding у конфігурації PuTTY

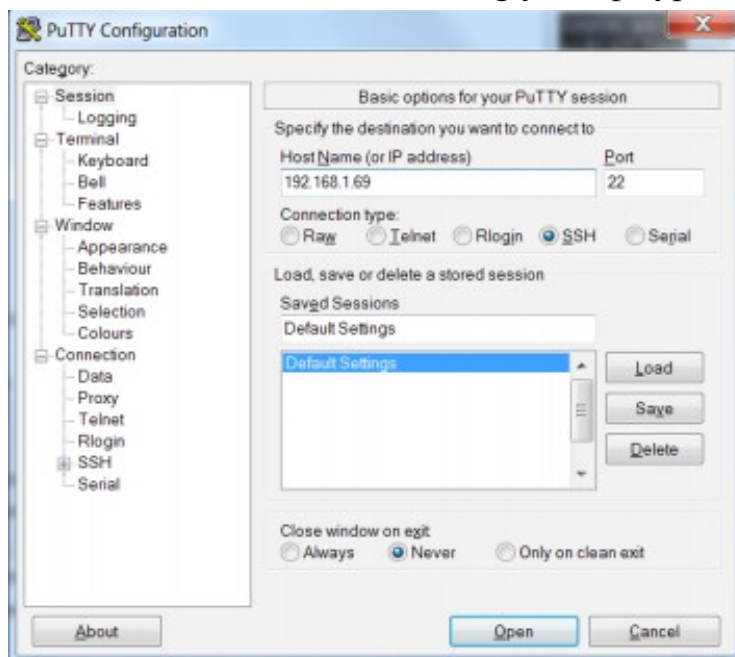


Рисунок 1.8. Відкриття з'єднання SSH з Raspberry Pi за допомогою PuTTY

Для OS X або Linux натисніть на Terminal, щоб відкрити підключення до Raspberry Pi. Щоб підключитися до імені користувача pi за замовчуванням, з IP-адресою 192.168.1.69, використовуйте наступну команду;

Параметр -X дозволяє переадресовувати X11:

```
ssh -X pi@192.168.1.69
```

Якщо все добре, ви повинні бути привітані з запитом на ваш пароль (пам'ятайте за замовчуванням значення для користувача pi – raspberry). Переконайтеся, що у вас працює Xming, запустивши програму Xming з панелі Пуск. Потім у вікні терміналу введіть програму, яка зазвичай працює в Raspberry Pi, такі як блокнот і тд. Зачекайте, поки програма з'явиться на робочому столі комп'ютера (якщо виникає помилка, ви, напевно, забули ключити Xming, запустіть його та повторіть спробу).

X Windows і X11 це те, за допомогою чого Raspberry Pi (і багато інших комп'ютерів на базі Linux) може відображатись і керуватись графічним вікном як частиною робочого столу[15]. Для того, щоб X11 Forwarding працювало через мережеве підключення, нам потрібні як SSH, так і X11 Forwarding включені на Raspberry Pi. Виконайте наступні дії:

1. Щоб увімкнути (або вимкнути) SSH, ви можете отримати доступ до raspi config (просто введіть `sudo raspi-config` з терміналу) і натисніть SSH у меню Додаткові параметри, як показано на рисунку 1.9 (SSH часто вмикається за замовчуванням для більшості дистрибутивів, які дозволяють віддалені з'єднання без необхідності монітора налаштувати його):

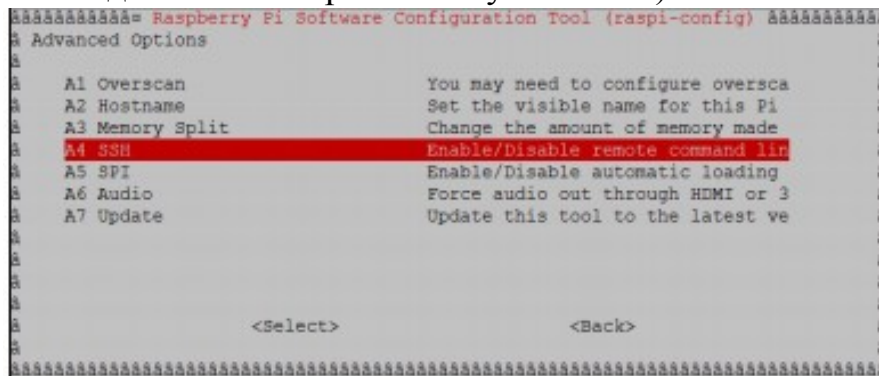


Рисунок 1.9. Меню розширених налаштувань у панелі raspi-config

2. Переконайтеся, що X11 Forwarding увімкнена на Raspberry Pi (знову ж таки більшість дистрибутивів вмикає її за замовчуванням).

3. Використовуйте nano наступною командою:

```
sudo nano /etc/ssh/sshd_config
```

4. Знайдіть рядок у файлі `/etc/ssh/sshd_config`, який керує X11 Forwarding і переконайтеся, що він відповідає "так" (без знака #):

```
X11Forwarding yes
```

5. Збережіть, якщо потрібно, натиснувши Ctrl + X, Y, і Enter і перезавантажте (щоб зміни вступили в силу):

sudo reboot

1.8 Збереження Raspberry Pi в актуальному стані

Образ Linux, який використовує Raspberry Pi, часто оновлюється, включаючи удосконалення і виправлення системи, а також додавання підтримки для нового обладнання або змін до останньої плати. Модулі, які ви встановлюєте, також можуть бути оновлені. Це особливо важливо, якщо ви плануєте використовувати той самий системний образ на іншій платі Raspberry Pi, оскільки у старих зображеннях відсутня підтримка будь-яких змін або альтернативних чіпів ОЗП. Нові прошивки повинні працювати на старих платах Raspberry Pi, але старіша прошивка може не бути сумісною з новішим обладнанням[16]. На щастя, ви не повинні перезавантажувати вашу SD-карту кожного разу, коли з'являється новий реліз, оскільки замість цього можна оновити модулі. Щоб оновити систему, потрібно підключитися до Інтернету. Завжди рекомендується зробити резервну копію зображення перед оновленням (і, як мінімум, скопіювати важливі файли).

Перед запуском будь-яких оновлень або встановлення будь-яких модулів, варто переконатися, що у вас є останній список модулів у сховищі. команда оновлення отримує останній список доступного програмного забезпечення та версії:

```
sudo apt-get update
```

Якщо ви просто хочете отримати оновлення ваших поточних пакетів, оновлення виведе їх все на сьогоднішній день:

```
sudo apt-get upgrade
```

Щоб переконаватися, що ви запускаєте останню версію Raspbian, ви можете запустити dist-upgrade (це може зайняти годину або близько того, залежно від кількості модулів, які мають бути оновлені). Це дозволить виконати всі оновлення і також видалить надлишкові пакети та очистить їх:

```
sudo apt-get dist-upgrade
```

Обидва способи оновлять програмне забезпечення, включаючи прошивку, що використовується під час завантаження та запуску (bootcode.bin і start.elf).

1.9 Спільний доступ до домашньої папки Raspberry Pi з SMB

Якщо ви підключили Raspberry Pi до мережі, ви можете отримати доступ до домашньої папки шляхом налаштування спільного доступу до файлів; це значно полегшує передачу файлів і забезпечує швидке завантаження файлів та є простим та ефективним способом резервного копіювання даних. Блок повідомлень веб-сервера (SMB) - це сумісний протокол спільного використання файлів Windows, OS X і Linux[17]. Переконайтеся, що у вас включена Raspberry Pi і вона підключена до інтернету. Вам також знадобиться інший комп'ютер у тій самій локальній мережі, щоб перевірити чи працює веб-сервер.

По-перше, вам потрібно встановити samba – веб-сервер, який обробляє спільні папки у форматі, що є сумісні з методами обміну Windows, OS X та Linux.

Спершу перевірте ваші оновлення, щоб отримати останній список доступних пакетів:

```
sudo apt-get update  
sudo apt-get install samba
```

Інсталяція потребуватиме близько 20 МБ простору і займе кілька хвилин.

Після завершення установки ми можемо зробити копію файлу конфігурації, щоб дозволити нам відновити її за замовчуванням, якщо буде необхідно:

```
sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.backup  
sudo nano /etc/samba/smb.conf
```

Прокрутіть вниз і знайдіть розділ з іменем аутентифікації; розкоментуйте його (приберіть #), тобто замініть `# security = user` на `security = user`.

Як описано у файлі, це налаштування гарантує, що ви повинні ввести своє ім'я користувача та пароль для Raspberry Pi для доступу до файлів (це важливо для спільних мереж). Знайдіть розділ, що називається Share Definitions і [homes], і змініть `read only = yes`, замість `read only = no`. Це дозволить нам переглядати та записувати файли до спільної папки. Після того, як це зроблено, збережіть і вийдіть, натиснувши `Ctrl + X`, `Y` і `Enter`.

Тепер ми можемо додати `pi` (користувач за замовчуванням) для використання `samba` наступним чином:

```
sudo pdbedit -a -u pi
```

Тепер введіть пароль (ви можете використовувати той самий пароль, що й логін або вибрати інший, але не використовуйте пароль за замовчуванням `raspberrypi`, який буде дуже легко для когось вгадати). Перезапустіть `samba` для використання нового файлу конфігурації наступним чином:

```
sudo /etc/init.d/samba restart
```

Для того, щоб протестувати роботу веб-серверу, вам потрібно знати або ім'я хоста Raspberry Pi (за замовчуванням ім'ям хоста є `raspberrypi`) або його IP-адресу. Ви можете знайти їх обидва наступною командою:

```
hostname
```

Для того, щоб дізнатися IP-адресу додайте `-I` наступним чином:

```
hostname -I
```

На іншому комп'ютері в розділі Мережа введіть адресу `\\raspberrypi\pi` у Наутілусі. Залежно від мережі, комп'ютер повинен знайти Raspberry Pi в мережі, запитає його ім'я користувача та пароль. Якщо комп'ютер не може знайти спільний доступ за допомогою `hostname`, ви можете використовувати IP-адресу безпосередньо, де, наприклад, адресу `192.168.1.69` слід змінити наступним чином: `\\192.168.1.69pi`.

РОЗДІЛ 2

РОБОТА ВИМІРЮВАЛЬНОЇ СИСТЕМИ ТА ЇЇ ІНТЕРФЕЙС

Сьогодні все більше і більше людей використовують відновлювальні джерела енергії з екологічних або з фінансових причин, для себе чи в якості зеленого тарифу. В той же час велика кількість виробників пропонують свої фотомодулі, інвертори та інше обладнання для сонячних систем. Кожен завіряє у паспортних даних певні характеристики того чи іншого обладнання, і споживач, який хоче отримати максимальну вигоду з покупки, завжди дивиться на ціну/якість. Але споживач не знає, чи дійсно фотомодулі будуть виробляти стільки енергії, як завірив виробник, чи дійсно інвертор розрахований на саме ту потужність, про яку каже виробник.

Через це сьогодні є дуже актуальним питання віддаленого моніторингу та тестування обладнання сонячних систем. Саме цю тему я розвинув у своїй дипломній роботі та отримав певний результат. Мова піде про автоматизовану вимірювальну систему характеристик фотомодулів та інверторів на базі мікрокомп'ютера Raspberry Pi 3. Крім того, у своїй роботі я намагався підкреслити, який великий функціонал мають мікрокомп'ютери в наші дні, на їх основі можна з легкістю досліджувати ті проблемні питання, які декілька років тому здавались неможливими у вирішенні.

Отже, у своїй роботі я розкажу про первинне підключення одноплатного мікрокомп'ютера, його налаштування, можливості, та, безпосередньо, про свою програму.

2.1 Інтерфейс програми для визначення та обробки характеристик сонячного обладнання

У своїй роботі я використовував мій ноутбук, одноплатний мікрокомп'ютер Raspberry Pi 3 Model B+, 4-канальний 16-розрядний АЦП ADS1115[18] та невелику кількість електронних компонентів[19]. Інструкція по первинному включенню описана у 1-му розділі.

Код я писав на мові Python. У своєму коді я застосовував велику кількість бібліотек, зокрема, для роботи з АЦП[20] та створення графічного інтерфейсу моєї програми[21]. Програма є повністю автоматизована і сприймає тільки випрямлені значення, оскільки АЦП не підтримує від'ємних значень. Так, можна дослідити ефективність роботи інвертора, ВАХ фотомодуля тощо. Після вводу користувачем певних величин, програма зніме характеристики, занесе ці дані до файлу[22] та побудує графік залежності характеристик[23] від кількості значень, які потрібно було зняти[24]. За допомогою чек-боксів користувач може:

- зберегти величини для наступного вимірювання, щоб знову їх не вводити у програму;
- зберегти побудований програмою графік після того, як характеристики будуть зняті і він будет побудований;

- використовувати для зняття характеристик 1, 2 чи 3 каналу 4-канального АЦП, тобто паралельно можна знімати до 3-х характеристик.

Користувачеві необхідно буде ввести такі величини, як:

- затримка між зняттям значень характеристик з АЦП;
- затримка між зняттям значень характеристик з АЦП;
- кількість значень, які потрібно зняти;
- масштаб для осі у для першого, другого та третього каналів (за замовчуванням ввести тільки для 1-го каналу, інші якщо користувач буде вимірювати декілька характеристик паралельно).

Також програма містить у собі такі кнопки, як:

- вихід з програми;
- запуск зняття характеристик, занесення даних до файлу та побудова графіку;
- відкриття файлу (файлу з даними або графік, ці файли відкриваються безпосередньо у самій програмі[25]);
- надіслати файл по веб-серверу на інший комп'ютер.

Програма має також чіткий та суворий обробник помилок і виключень. Обробник помилок покаже, чи є помилки у:

- підключенні АЦП до Raspberry Pi;
- правильності введених значень користувачем (мінусові значення затримки, кількості значення для зняття характеристик, букви, пусті поля, недоречні відношення мінімального до максимального значення масштабів по осі у тощо)
- відкритті певного файлу в програмі;
- надсиланні файлу по веб-серверу до іншого комп'ютера.

2.2 Програмний код вимірювальної системи

Нижче приведений програмний код з усіма коментарями для того, щоб запустити мою програму, яка буде здійснювати моніторинг обладнання ФЕС.

Написаний на мові Python:

```
# For operations with csv files
```

```
import csv
```

```
# Import modules for GUI
```

```
import tkinter as tk
```

```
from tkinter import Frame
```

```
from tkinter import Text
```

```
from tkinter import Label
```

```
import tkinter.messagebox as mb
```

```

import tkinter.filedialog as fd
from PIL import Image, ImageTk

# Importing modules used for builing figures
import matplotlib.pyplot as plt
import numpy as np

# System imports
import os
import shutil
import time

# For exceptions
import traceback

# Importing libraries for ADS1115 connection
import board
import busio
import adafruit_ads1x15.ads1115 as ADS
from adafruit_ads1x15.analog_in import AnalogIn

class GUI(Frame):

    # The very beginning of class
    def __init__(self, master=None):
        self.master = master
        self.name_for_save_fig = '1'
        self.remember_val = tk.IntVar()
        self.ymin2 = 0
        self.ymax2 = 0
        self.ymin3 = 0
        self.ymax3 = 0
        self.save_fig = tk.IntVar()
        self.two_chan = tk.IntVar()

```

```

self.three_chan = tk.IntVar()
self.csv_file1 = 'chan1.csv'
self.csv_file2 = 'chan2.csv'
self.csv_file3 = 'chan3.csv'
self.init_window()

# Function for saving figure as pdf/png image
def save(self, name='', fmt='png'):

    try:
        pwd = os.getcwd()
        iPath = './pictures/{}'.format(fmt)
        if not os.path.exists(iPath):
            os.makedirs(iPath)
        os.chdir(iPath)
        plt.savefig('{}.'.format(name, fmt), fmt='png')
        os.chdir(pwd)
        self.text_box.insert(tk.END, "Figure saved\n")

    # Case if error occurs while saving a figure
    except Exception as e:
        print(traceback.format_exc())
        mb.showerror("Save Figure", "Failed to save figure:\n" + str(e))
        return

# Function that helps with opening figures and csv data
def open_file(self):

    try:
        # Call the window which helps to find a file to open
        filename = fd.askopenfilename(filetypes = (("CSV Files", ".csv"), ("PNG Files", ".png"),
        ("PDF Files", '.pdf'), ("All files", "*.*")))

        # Case if opening file is .csv

```

```

if '.csv' in filename:
    # Open and read data of this file
    f = open(filename)
    file_read = f.read()

    # Removing previous data from text_box
    self.text_box.delete(1.0, "end-1c")

    # Output data into text box of program
    self.text_box.insert(tk.END, file_read)

    # Close file
    f.close()

# Case if opening file is .png or .pdf
elif '.png' in filename or '.pdf' in filename:
    # Open image
    img = Image.open(filename)

    # Removing previous data from text_box
    self.text_box.delete(1.0, "end-1c")

    # Make resizing of image and saving it
    basewidth = 400
    wpercent = (basewidth / float(img.size[0]))
    hsize = int((float(img.size[1]) * float(wpercent)))
    img = img.resize((basewidth, hsize), Image.ANTIALIAS)
    os.remove(filename)
    img.save(filename)

    # Open and put new image to program window
    new_img = Image.open(filename)
    render = ImageTk.PhotoImage(new_img)
    self.img_lbl = tk.Button(self.master, image=render, command=self.imgpress)

```

```

        self.img_lbl.image = render
        self.img_lbl.place(x=0, y=0)

# If selected file nor csv neither pdf or png
elif '.' in filename:
    mb.showwarning("Open File", "Type of file is not satisfied by a program\n")

# Case if error occurred while opening file
except Exception as e:
    print(traceback.format_exc())
    mb.showerror("Open File", "Failed to open a file:\n" + str(e))
    return

# Function that sends selected file to shared directory
def send_file(self):

    try:
        # Call the window which helps to find a file to open
        filename = fd.askopenfilename(filetypes = (("CSV Files", ".csv"), ("PNG Files", ".png"),
        ("PDF Files", '.pdf'), ("All files", "*.*")))

        source_file = filename
        destination_folder = '/home/pi/share/'
        # Case if selected file is .csv
        if '.csv' in filename:

            # If file exists remove it
            path_to_csv = destination_folder + self.csv_file1
            if os.path.exists(path_to_csv):
                os.remove(path_to_csv)

            # Moves selected csv file to destination folder
            shutil.move(source_file, destination_folder)

```

```

# Case if selected file is .png or .pdf
elif '.png' in filename or '.pdf' in filename:

    # If file exists remove it
    path_to_png = destination_folder + self.name_for_save_fig + '.png'
    path_to_pdf = destination_folder + self.name_for_save_fig + '.pdf'
    if os.path.exists(path_to_png) and '.png' in filename:
        os.remove(path_to_png)
    elif os.path.exists(path_to_pdf) and '.pdf' in filename:
        os.remove(path_to_pdf)

    # Moves selected file to destination folder
    shutil.move(source_file, destination_folder)

# Removing previous data from text_box
self.text_box.delete(1.0, "end-1c")

# Printing about successfull operation
self.text_box.insert(tk.END, "File has been moved to {}!\n".format(destination_folder))
# Case if error occurred while opening file
except Exception as e:
    print(traceback.format_exc())
    mb.showerror("Open File", "Failed to open a file:\n" + str(e))
    return

# Image press handler, removes image while clicking on it
def imgpress(self):
    self.img_lbl.destroy()
    self.text_box.insert(tk.END, "Image has been removed\n")
    return

# Function that manages data processing
def process_data(self, event=None):

```

```

try:
    # Check for correct checks of buttons
    if int(self.two_chan.get()) == 1 and int(self.three_chan.get()) == 1:
        mb.showwarning("Error", 'So do you want to use 2 or 3 channels of ADS1115? :D')
        return

    # Create connection to ADS1115 via I2C
    self.init_ads1x15()

    # Get values from user input
    tmp1 = self.e1.get()
    tmp2 = self.e2.get()
    tmp3 = self.e3.get()
    tmp4 = self.e4.get()
    tmp5 = 0
    tmp6 = 0
    tmp7 = 0
    tmp8 = 0
    if int(self.two_chan.get()) == 1:
        tmp5 = self.e5.get()
        tmp6 = self.e6.get()

    if int(self.three_chan.get()) == 1:
        tmp5 = self.e5.get()
        tmp6 = self.e6.get()
        tmp7 = self.e7.get()
        tmp8 = self.e8.get()

    # Check for invalid values
    if tmp1.lstrip('-').isdigit() == False or tmp2.lstrip('-').isdigit() == False or tmp3.replace(".",
    "", 1).lstrip('-').isdigit() == False or tmp4.replace(".", "", 1).lstrip('-').isdigit() == False:
        mb.showwarning("Error", 'Fields must be digits and not empty')
        return

```

```

        if int(self.two_chan.get()) == 0 and int(self.three_chan.get()) == 0 and (tmp5 or tmp6 or
tmp7 or tmp8):
            mb.showwarning("Error", 'Fill \'Use 2/3 channels\' before entering y limits for this
channels')
            return
        if int(self.two_chan.get()) == 1 and (tmp5.replace(".", "", 1).lstrip('-').isdigit() == False or
tmp6.replace(".", "", 1).lstrip('-').isdigit() == False):
            mb.showwarning("Error", 'Fields must be digits and not empty')
            return
        if int(self.three_chan.get()) == 1 and (tmp5.replace(".", "", 1).lstrip('-').isdigit() == False
or tmp6.replace(".", "", 1).lstrip('-').isdigit() == False or
tmp7.replace(".", "", 1).lstrip('-').isdigit() == False or tmp8.replace(".", "",
1).lstrip('-').isdigit() == False):
            mb.showwarning("Error", 'Fields must be digits and not empty')
            return

# Assigning values from user input
self.delay = int(self.e1.get())
self.delay_ms = self.delay / 1000
self.measure_nb = int(self.e2.get())
self.ymin1 = float(self.e3.get())
self.ymax1 = float(self.e4.get())

# Check if values from input fields are not empty and have only digits
if self.e5.get() != "" and self.e5.get().replace(".", "", 1).lstrip('-').isdigit() == True:
    self.ymin2 = float(self.e5.get())
if self.e6.get() != "" and self.e6.get().replace(".", "", 1).lstrip('-').isdigit() == True:
    self.ymax2 = float(self.e6.get())
if self.e7.get() != "" and self.e7.get().replace(".", "", 1).lstrip('-').isdigit() == True:
    self.ymin3 = float(self.e7.get())
if self.e8.get() != "" and self.e8.get().replace(".", "", 1).lstrip('-').isdigit() == True:
    self.ymax3 = float(self.e8.get())

# Check for invalid values

```

```

if self.delay <= 0 or self.measure_nb <= 0:
    mb.showwarning("Error", 'Delay and number of measurements cannot be less or equal
zero')
    return
if self.ymin1 >= self.ymax1:
    mb.showwarning("Error", 'ymin1 cannot be greater or equal ymax1')
    return
if self.ymin2 >= self.ymax2 and int(self.two_chan.get()) == 1:
    mb.showwarning("Error", 'ymin2 cannot be greater or equal ymax2')
    return
if (self.ymin3 >= self.ymax3 or self.ymin2 >= self.ymax2) and int(self.three_chan.get()) ==
1:
    mb.showwarning("Error", 'ymin2/ymin3 cannot be greater or equal ymax2/ymax3')
    return

# Removing initial text from text box
self.text_box.delete(1.0, "end-1c")

# Assigning values to x and y
x = np.arange(self.measure_nb)
y1 = []
y2 = []
y3 = []

# Check for invalid limits of y axis
for i in range(0, int(self.measure_nb)):
    if float(self.chan1.voltage) < self.ymin1:
        mb.showwarning("Error", 'ymin1 cannot be greater than measured voltage:
{}'.format(round(self.chan1.voltage, 3)))
        return
    if float(self.chan1.voltage) > self.ymax1:
        mb.showwarning("Error", 'ymax1 must be greater than measured voltage:
{}'.format(round(self.chan1.voltage, 3)))

```

```

        return

    if int(self.two_chan.get()) == 1:
        for i in range(0, int(self.measure_nb)):
            if float(round((self.chan2.voltage * 0.22), 3)) < self.ymin2:
                mb.showwarning("Error", 'ymin2 cannot be greater than measured current:
                {}'.format(round((self.chan2.voltage * 0.22), 3)))
            return
            if float(round((self.chan2.voltage * 0.22), 3)) > self.ymax2:
                mb.showwarning("Error", 'ymax2 must be greater than measured current:
                {}'.format(round((self.chan2.voltage * 0.22), 3)))
            return

    elif int(self.three_chan.get()) == 1:
        for i in range(0, int(self.measure_nb)):
            if float(round((self.chan2.voltage * 0.22), 3)) < self.ymin2:
                mb.showwarning("Error", 'ymin2 cannot be greater than measured current:
                {}'.format(round((self.chan2.voltage * 0.22), 3)))
            return
            if float(round((self.chan2.voltage * 0.22), 3)) > self.ymax2:
                mb.showwarning("Error", 'ymax2 must be greater than measured current:
                {}'.format(round((self.chan2.voltage * 0.22), 3)))
            return

        for i in range(0, int(self.measure_nb)):
            if float(self.chan3.voltage) < self.ymin3:
                mb.showwarning("Error", 'ymin3 cannot be greater than measured voltage:
                {}'.format(round(self.chan3.voltage, 3)))
            return
            if float(self.chan3.voltage) > self.ymax3:
                mb.showwarning("Error", 'ymax3 must be greater than measured voltage:
                {}'.format(round(self.chan3.voltage, 3)))
            return

```

Printing title of table with voltage values

```

with open(self.csv_file1, mode='w') as csv_file1:
    csv_write1 = csv.writer(csv_file1, delimiter=',')
    csv_write1.writerow(['#', 'raw', ' V'])

# Case if user uses 2 channels, creates another file
if int(self.two_chan.get()) == 1:
    with open(self.csv_file2, mode='w') as csv_file2:
        csv_write2 = csv.writer(csv_file2, delimiter=',')
        csv_write2.writerow(['#', 'raw', ' I'])

# Same if user uses 3 channels, creates two another files with data
elif int(self.three_chan.get()) == 1:
    with open(self.csv_file2, mode='w') as csv_file2:
        csv_write2 = csv.writer(csv_file2, delimiter=',')
        csv_write2.writerow(['#', 'raw', ' I'])
    with open(self.csv_file3, mode='w') as csv_file3:
        csv_write3 = csv.writer(csv_file3, delimiter=',')
        csv_write3.writerow(['#', 'raw', ' V'])

# Printing resulting values to file
with open(self.csv_file1, mode='a') as csv_write1:
    for i in range(0, int(self.measure_nb)):
        csv_writer1 = csv.writer(csv_write1, delimiter=',')
        csv_writer1.writerow([(i + 1), self.chan1.value, round(self.chan1.voltage, 5)])
        y1.append(self.chan1.voltage)
        time.sleep(self.delay_ms)

# Case if user wants to use 2 channels, also writing resulting values to another file
if int(self.two_chan.get()) == 1:
    with open(self.csv_file2, mode='a') as csv_write2:
        for i in range(0, int(self.measure_nb)):
            csv_writer2 = csv.writer(csv_write2, delimiter=',')
            csv_writer2.writerow([(i + 1), int(self.chan2.value * 0.22),
round((self.chan2.voltage * 0.22), 5)])

```

```

        y2.append(self.chan2.voltage * 0.22)
        time.sleep(self.delay_ms)

# Same if user uses 3 channels, write data to 2 another files
elif int(self.three_chan.get()) == 1:
    with open(self.csv_file2, mode='a') as csv_write2:
        for i in range(0, int(self.measure_nb)):
            csv_writer2 = csv.writer(csv_write2, delimiter=',')
            csv_writer2.writerow([(i + 1), int(self.chan2.value * 0.22),
round((self.chan2.voltage * 0.22), 5)])
            y2.append(self.chan2.voltage * 0.22)
            time.sleep(self.delay_ms)
    with open(self.csv_file3, mode='a') as csv_write3:
        for i in range(0, int(self.measure_nb)):
            csv_writer3 = csv.writer(csv_write3, delimiter=',')
            csv_writer3.writerow([(i + 1), self.chan3.value, round(self.chan3.voltage, 5)])
            y3.append(self.chan3.voltage)
            time.sleep(self.delay_ms)

# Printing about successfull write to file
self.text_box.insert(tk.END, "Data was successfully written to file
{}\n".format(self.csv_file1))

if int(self.two_chan.get()) == 1:
    self.text_box.insert(tk.END, "Data was successfully written to file
{}\n".format(self.csv_file2))

elif int(self.three_chan.get()) == 1:
    self.text_box.insert(tk.END, "Data was successfully written to file
{}\n".format(self.csv_file2))
    self.text_box.insert(tk.END, "Data was successfully written to file
{}\n".format(self.csv_file3))

# Building a plot

```

```

if int(self.two_chan.get()) == 0 and int(self.three_chan.get()) == 0:
    # Assigning title and labels to axis
    plt.title("V = f(n)")
    plt.xlabel('Number of measurements, n')
    plt.ylabel('Voltage, V')

    # Build plot with grid and set y axis limits
    plt.plot(x, y1)
    plt.grid(True)
    plt.ylim(float(self.ymin1), float(self.ymax1))

elif int(self.two_chan.get()) == 1:
    # Build subplot for each channel data on figure object
    fig = plt.figure(figsize=(10,5))

    # Set the title of subplots and legends, set grid, limits and build plots
    ax1 = fig.add_subplot(211)
    ax1.title.set_text("V = f(n)")
    plt.plot(x, y1)
    plt.xlabel('Number of measurements, n')
    plt.ylabel('Voltage, V')
    plt.grid(True)
    plt.ylim(float(self.ymin1), float(self.ymax1))

    ax2 = fig.add_subplot(212, sharex=ax1)
    ax2.title.set_text("I = f(n)")
    plt.plot(x, y2)
    plt.xlabel('Number of measurements, n')
    plt.ylabel('Current, I')
    plt.grid(True)
    plt.ylim(float(self.ymin2), float(self.ymax2))

    # Correct layout of subplots
    plt.tight_layout()

```

```

# Same case with 3 channel input
elif int(self.three_chan.get()) == 1:
    fig = plt.figure(figsize=(12,7))
    ax1 = fig.add_subplot(311)
    ax1.title.set_text("V = f(n)")
    plt.plot(x, y1)
    plt.grid(True)
    plt.xlabel('Number of measurements, n')
    plt.ylabel('Voltage, V')
    plt.ylim(float(self.ymin1), float(self.ymax1))

    ax2 = fig.add_subplot(312, sharex=ax1)
    ax2.title.set_text("I = f(n)")
    plt.plot(x, y2)
    plt.grid(True)
    plt.xlabel('Number of measurements, n')
    plt.ylabel('Current, I')
    plt.ylim(float(self.ymin2), float(self.ymax3))
    ax3 = fig.add_subplot(313, sharex=ax1)
    ax3.title.set_text("V = f(n)")
    plt.plot(x, y3)
    plt.grid(True)
    plt.xlabel('Number of measurements, n')
    plt.ylabel('Voltage, V')
    plt.ylim(float(self.ymin3), float(self.ymax3))
    fig.tight_layout()

# If check button 'Remember values' is False, then remove values from fields
if int(self.remember_val.get()) == 0:
    self.e1.delete(0, tk.END)
    self.e2.delete(0, tk.END)
    self.e3.delete(0, tk.END)
    self.e4.delete(0, tk.END)

```

```

if int(self.remember_val.get()) == 0 and int(self.two_chan.get()) == 1:
    self.e5.delete(0, tk.END)
    self.e6.delete(0, tk.END)

elif int(self.remember_val.get()) == 0 and int(self.three_chan.get()) == 1:
    self.e5.delete(0, tk.END)
    self.e6.delete(0, tk.END)
    self.e7.delete(0, tk.END)
    self.e8.delete(0, tk.END)

# Showing plot on a screen
plt.show(block=False)

# If check button 'Save figure' is True, then save the figure
if int(self.save_fig.get()) == 1:
    self.save(self.name_for_save_fig)

# Case if error occurred while processing data
except Exception as e:
    print(traceback.format_exc())
    mb.showerror("Process Data", "Failed to process data:\n" + str(e))
    return

# Function that handles quit button
def callback(self):
    # Prompts a window if user wants to quit
    if mb.askyesno('Verify', 'Really quit?'):
        exit()
    else:
        mb.showinfo('No', 'Quit has been cancelled')

# Create save connection to ADC via I2C bus
def init_ads1x15(self):

```

```

try:
    # Create the I2C bus
    self.i2c = busio.I2C(board.SCL, board.SDA)

    # Create the ADC object using the I2C bus
    self.ads = ADS.ADS1115(self.i2c)

    # Create single-ended input on channel 0
    self.chan1 = AnalogIn(self.ads, ADS.P0)

    # Case if user wants to use 2 channels, also create single-ended input on channel 1
    if int(self.two_chan.get()) == 1:
        self.chan2 = AnalogIn(self.ads, ADS.P1)

    # Same if user wants 3 channels, also create input on channel 1 and 2
    elif int(self.three_chan.get()) == 1:
        self.chan2 = AnalogIn(self.ads, ADS.P1)
        self.chan3 = AnalogIn(self.ads, ADS.P2)

    # Case if error occurred while connecting ADS1115 to Raspberry Pi
    except Exception as e:
        print(traceback.format_exc())
        mb.showerror("ADS1115 Connection", "Failed to connect ADS1115 with Raspberry Pi:\n"
+ str(e))
        return

# Main function, that makes GUI
def init_window(self):

    try:
        # Assigning title to our window
        self.master.title("Measuring voltage")

        # Create name of entry fields

```

```

tk.Label(self.master, padx=30, text="Delay (in ms):").grid(row=0)
tk.Label(self.master, padx=30, text="Number of measurements:").grid(row=1)
tk.Label(self.master, padx=30, text="Minimum of y axis channel 1:").grid(row=2)
tk.Label(self.master, padx=30, text="Maximum of y axis channel 1:").grid(row=3)
tk.Label(self.master, padx=30, text="Minimum of y axis channel 2:").grid(row=4)
tk.Label(self.master, padx=30, text="Maximum of y axis channel 2:").grid(row=5)
tk.Label(self.master, padx=30, text="Minimum of y axis channel 3:").grid(row=6)
tk.Label(self.master, padx=30, text="Maximum of y axis channel 3:").grid(row=7)

# Create a box, which will behave like console window
self.text_box = tk.Text(self.master, width=50, height=15)
self.text_box.grid(row=10, column=0, columnspan=15, sticky='W', padx=65)
self.text_box.insert("end-1c", "Enter values\n")

# Add scrollbar to the Text widget
self.scroll = tk.Scrollbar(self.master)
self.scroll.config(command=self.text_box.yview)
self.text_box.config(yscrollcommand=self.scroll.set)
self.scroll.grid(row=10, column=1, columnspan=15, sticky='NS')

# Create check buttons
tk.Checkbutton(self.master, text="Remember values",
variable=self.remember_val).grid(row=0, column=2, sticky=tk.NS, padx=35)
tk.Checkbutton(self.master, text="Save figure after", variable=self.save_fig).grid(row=1,
column=2, sticky=tk.NS, padx=35)
tk.Checkbutton(self.master, text="Use 2 channels", variable=self.two_chan).grid(row=2,
column=2, sticky=tk.NS, padx=30)
tk.Checkbutton(self.master, text="Use 3 channels", variable=self.three_chan).grid(row=3,
column=2, sticky=tk.NS, padx=30)

# Create entry fields
self.e1 = tk.Entry(self.master)
self.e2 = tk.Entry(self.master)
self.e3 = tk.Entry(self.master)
self.e4 = tk.Entry(self.master)

```

```

self.e5 = tk.Entry(self.master)
self.e6 = tk.Entry(self.master)
self.e7 = tk.Entry(self.master)
self.e8 = tk.Entry(self.master)

# Assigning to fields basic values
self.e1.insert(10, "100")
self.e2.insert(10, "20")
self.e3.insert(10, "1")
self.e4.insert(10, "5")
self.e5.insert(10, "0")
self.e6.insert(10, "0")
self.e7.insert(10, "0")
self.e8.insert(10, "0")

# Make a grid of entry fields
self.e1.grid(row=0, column=1)
self.e2.grid(row=1, column=1)
self.e3.grid(row=2, column=1)
self.e4.grid(row=3, column=1)
self.e5.grid(row=4, column=1)
self.e6.grid(row=5, column=1)
self.e7.grid(row=6, column=1)
self.e8.grid(row=7, column=1)

# Binding Enter button as if user could press Apply button
self.e1.bind("<Return>", self.process_data)
self.e2.bind("<Return>", self.process_data)
self.e3.bind("<Return>", self.process_data)
self.e4.bind("<Return>", self.process_data)
self.e5.bind("<Return>", self.process_data)
self.e6.bind("<Return>", self.process_data)
self.e7.bind("<Return>", self.process_data)
self.e8.bind("<Return>", self.process_data)

```

```

        # Create two buttons which will call funcs if user press them
        tk.Button(self.master, text='Quit', command=self.callback).grid(row=8, column=0,
sticky=tk.W, pady=10, padx=60)
        tk.Button(self.master, text='Apply!', command=self.process_data).grid(row=8, column=1,
sticky=tk.W, pady=2, padx=60)
        tk.Button(self.master, text='Open file', command=self.open_file).grid(row=8, column=2,
sticky=tk.W, pady=2, padx=60)
        tk.Button(self.master, text='Send file!', command=self.send_file).grid(row=10, column=2,
columnspan=10, sticky=tk.W, pady=0, padx=60)

    # Case if error occurred while creating a GUI window
    except Exception as e:
        print(traceback.format_exc())
        mb.showerror("Window creation", "Failed to create a GUI window:\n" + str(e))

    # Destroy Tkinter object
    self.master.destroy()
    exit()

# Create a Tkinter object
root = tk.Tk()

# Creating an object of program
program = GUI(root)

# Show window
root.mainloop()

```

2.3 Зовнішній вигляд програми

На рисунку 2.1 ви можете побачити як виглядає моя програма:

The screenshot shows a window titled "Measuring voltage" with a standard macOS-style title bar (red, yellow, green buttons). The window contains several input fields and checkboxes. On the left, there are labels for "Delay (in ms):", "Number of measurements:", "Minimum of y axis channel 1:", "Maximum of y axis channel 1:", "Minimum of y axis channel 2:", "Maximum of y axis channel 2:", "Minimum of y axis channel 3:", and "Maximum of y axis channel 3:". To the right of these labels are corresponding input fields with values: 100, 20, 1, 5, 0, 0, 0, and 0. Further right, there are four checkboxes: "Remember values", "Save figure after", "Use 2 channels", and "Use 3 channels". At the bottom left, there is a "Quit" button. In the center bottom, there is an "Apply!" button. On the right side, there is an "Open file" button. At the bottom right, there is a "Send file!" button. A large text area with the placeholder "Enter values" is located below the "Quit" button. A vertical scrollbar is visible on the right side of the text area.

Parameter	Value
Delay (in ms):	100
Number of measurements:	20
Minimum of y axis channel 1:	1
Maximum of y axis channel 1:	5
Minimum of y axis channel 2:	0
Maximum of y axis channel 2:	0
Minimum of y axis channel 3:	0
Maximum of y axis channel 3:	0

Buttons: Quit, Apply!, Open file, Send file!

Checkboxes: Remember values, Save figure after, Use 2 channels, Use 3 channels

Text area: Enter values

Рисунок 2.1. Початковий вид програми

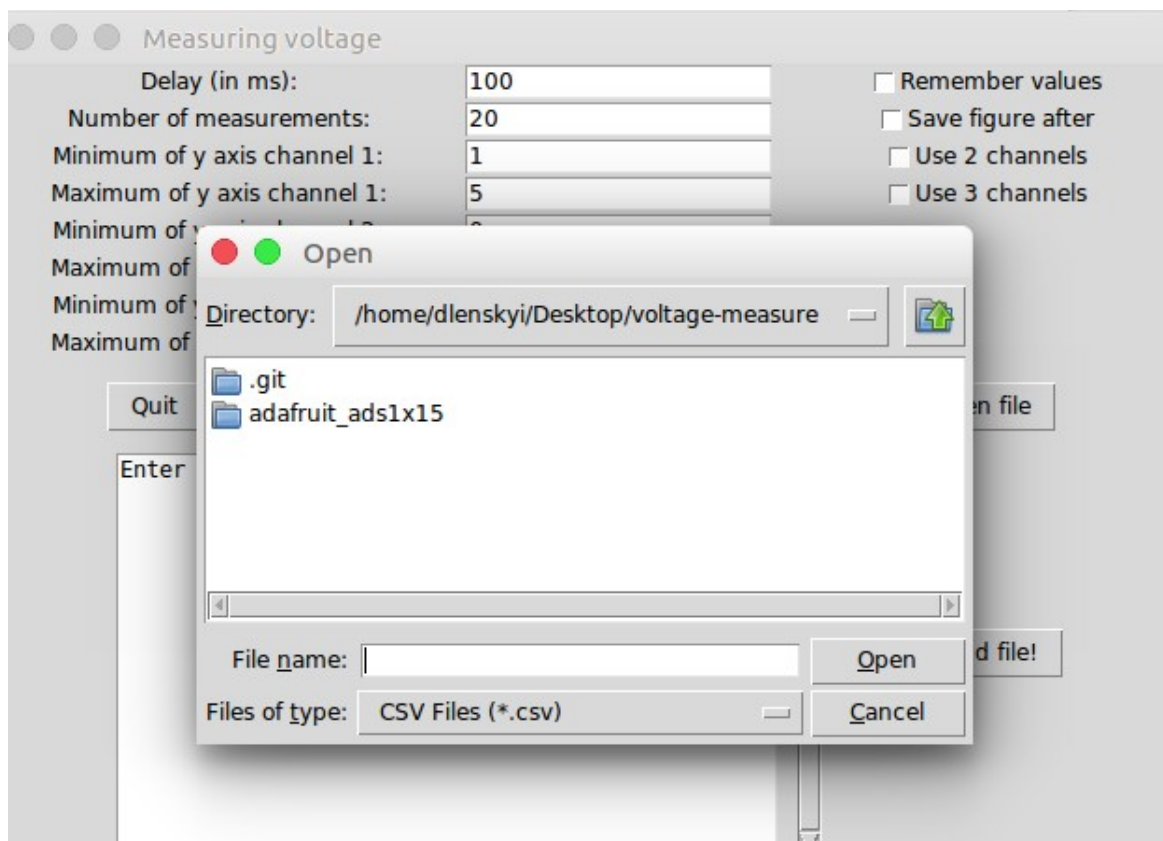


Рисунок 2.1, аркуш 2. Відкриття файлу

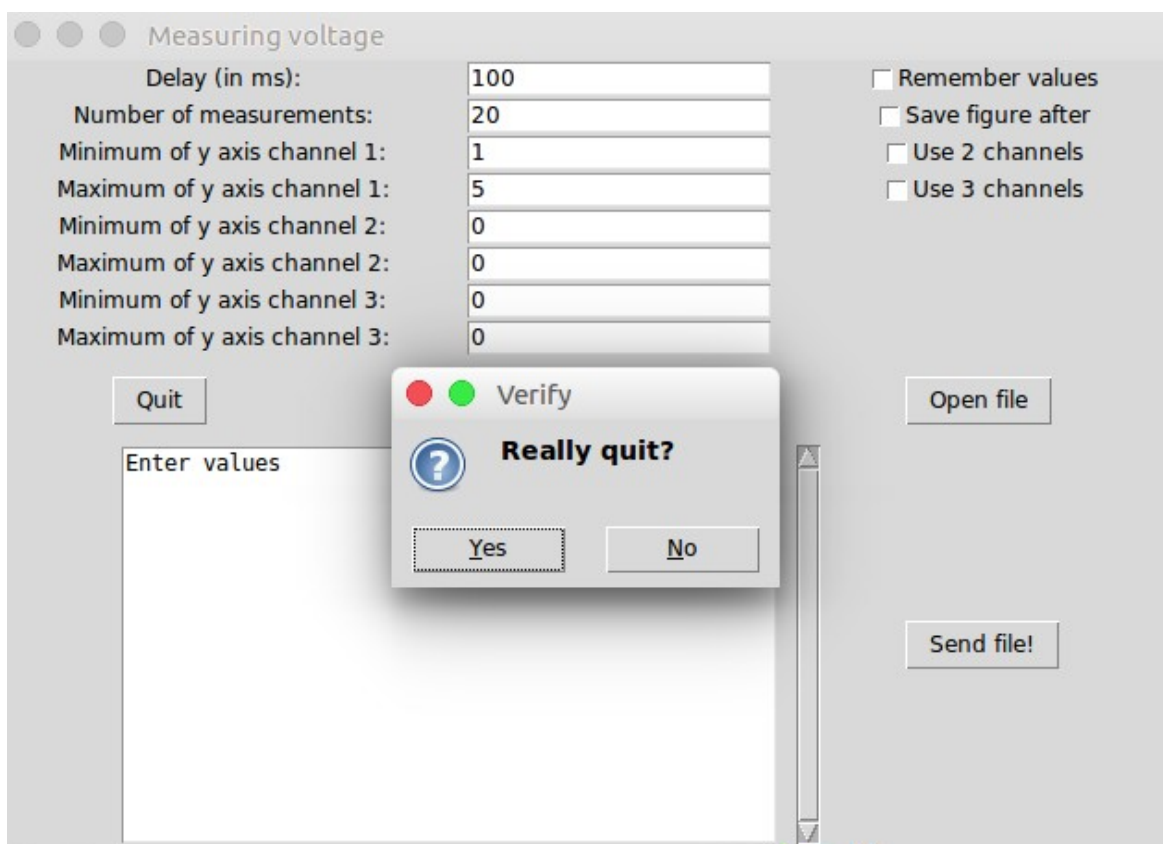


Рисунок 2.1, аркуш 2. Запит вийти з програми

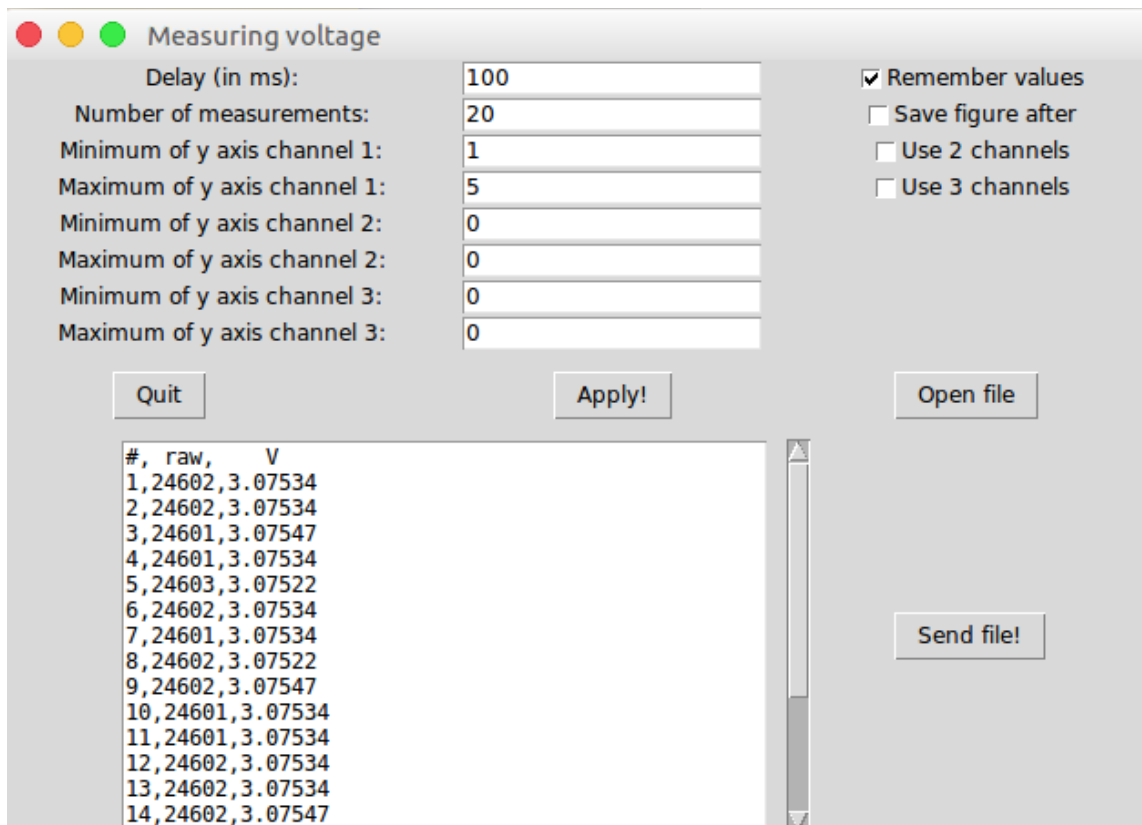


Рисунок 2.1, аркуш 3. Успішне відкриття файлу з даними та їх відображення

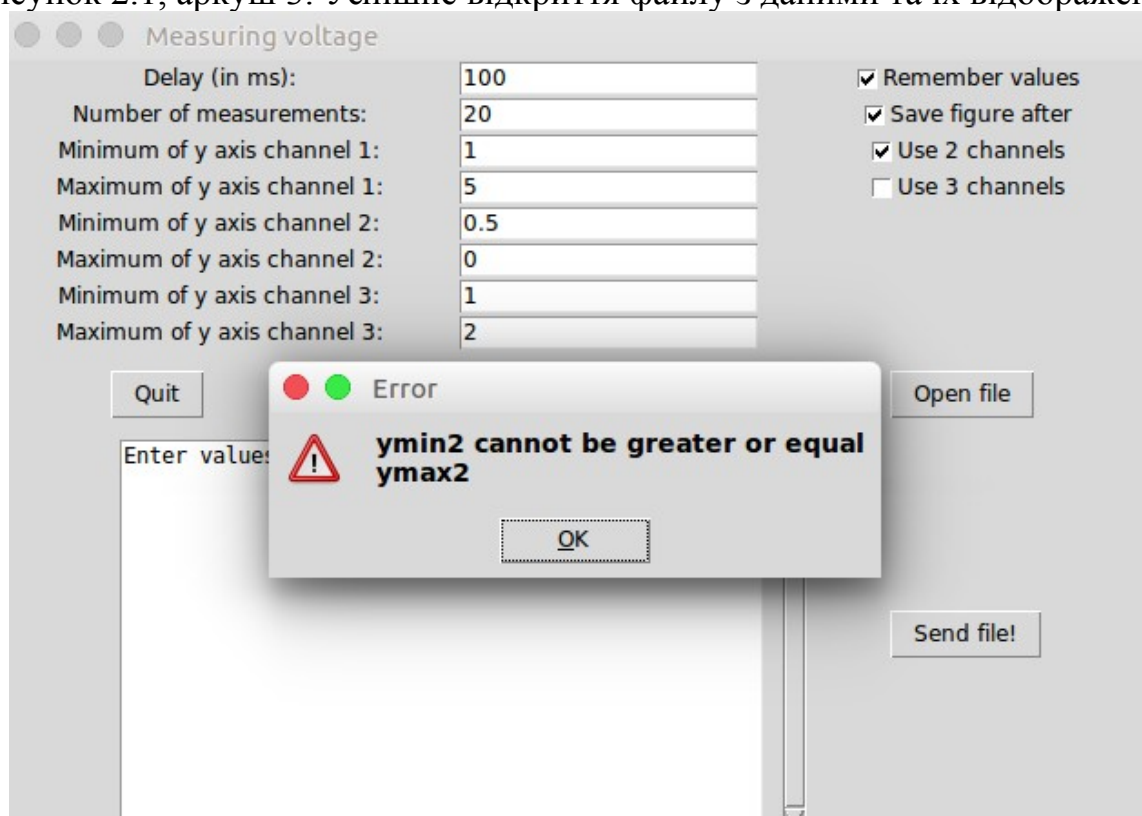


Рисунок 2.1, аркуш 3. Обробник помилок попереджає, що мінімальне значення у не має бути більше, ніж максимальне

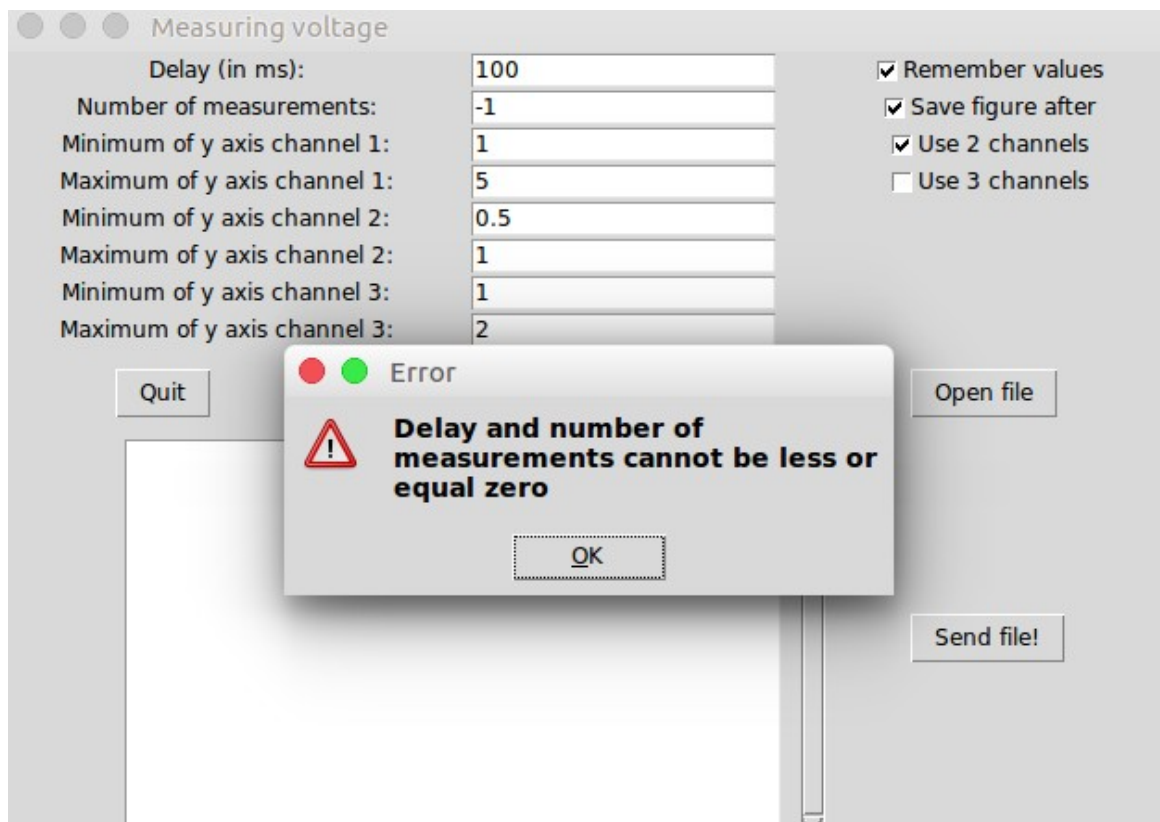


Рисунок 2.1, аркуш 4. Обробник помилок попереджає про невалідність введених значень користувачем

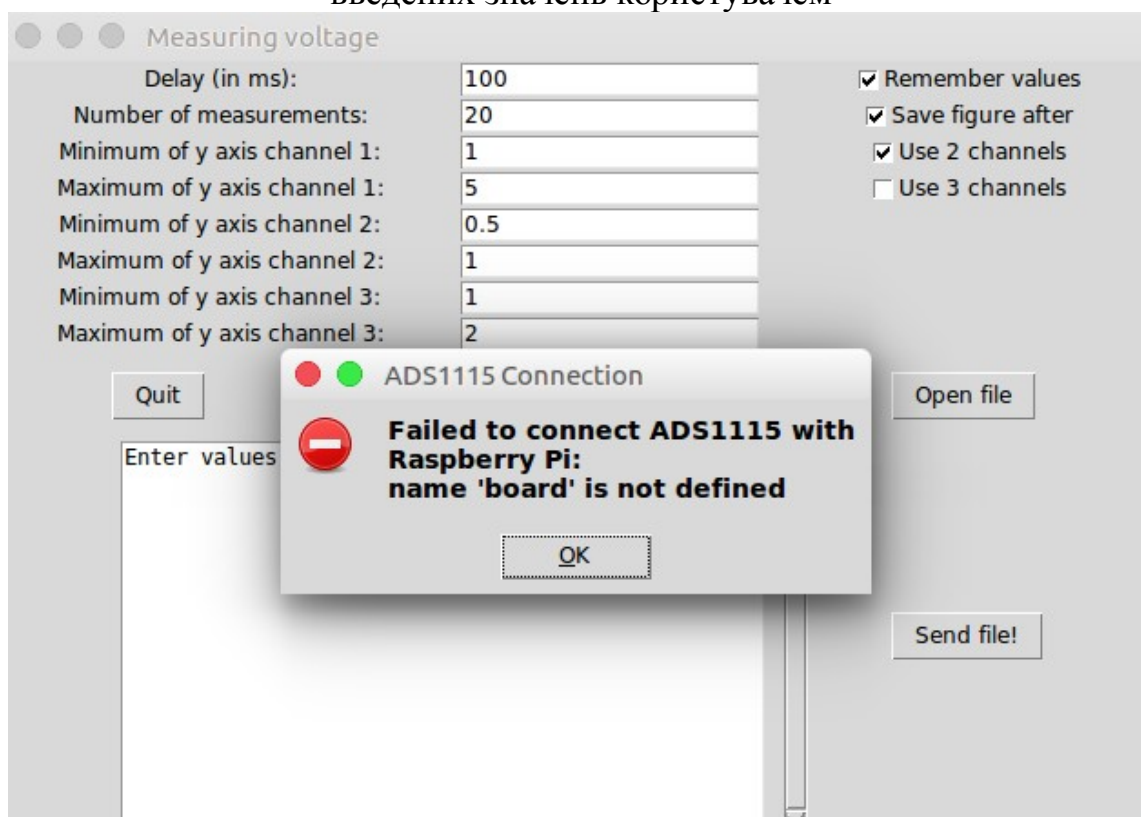


Рисунок 2.1, аркуш 4. Попередження про некоректне підключення АЦП до Pi

РОЗДІЛ 3

ОХОРОНА ПРАЦІ. ОЦІНКА ВПЛИВУ НА НАВКОЛИШНЄ СЕРЕДОВИЩЕ

3.1 Аналіз шкідливих та небезпечних факторів під час використання обладнання фотоелектричної станції напругою 0,4 кВ

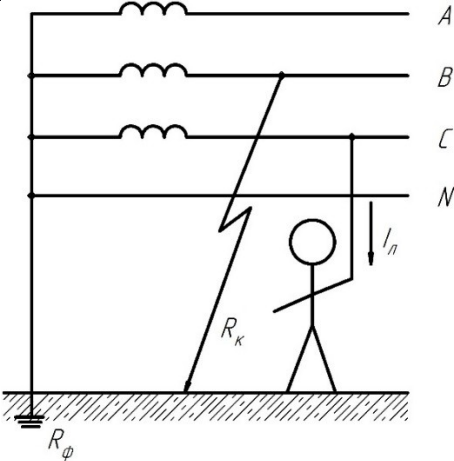
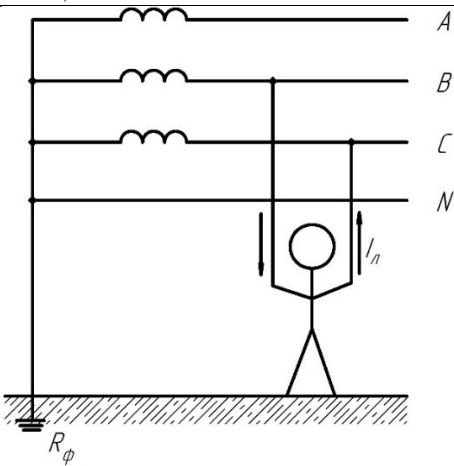
Головним небезпечним фактором під час використання фотоелектричного обладнання станції загальною напругою 0,4 кВ є напруга, що зумовлює проходження струму через людське тіло в результаті дотикання до струмовідних частин електроустановок. Нижче представлено можливі варіанти приєднання людини до електричного кола. Результати розрахунків приведено в таблиці 3.1.

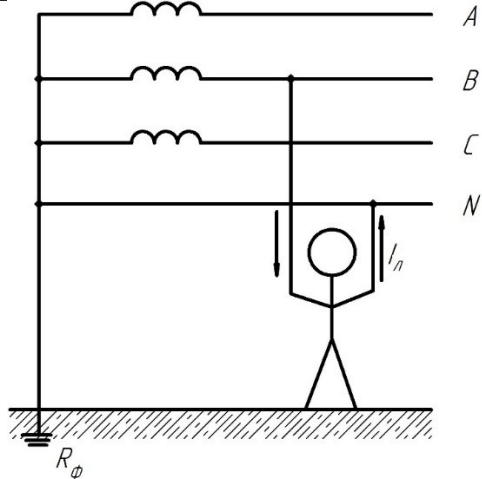
Так як сонячні панелі електростанції будуть розташовані на даху, небезпечним фактором буде виконання працівниками робіт на висоті. Роботи по встановленню сонячної електростанції проводяться на даху 21 корпусу КПП ім. Ігоря Сікорського, тобто там будуть розміщені всі фотопанелі.

Таблиця 3.1 – Струм, що проходитьиме через людське тіло, в момент дотикання її до струмовідних частин

Варіанти приєднання людини до електричного кола	Схема приєднання	Формула та розрахунок струму, що проходить через людину в момент дотикання, А
1	2	3
Однофазний прямий дотик до струмовідних частин з наругою мережі 0,4 кВ з глухозаземленою нейтраллю при нормалньому режимі		$I_n = \frac{U_\phi}{R_n} = \frac{220}{1500} = 0,146 \text{ (A)}$

Продовження таблиці 3.1

Однофазний прямий дотик до струмовідних частин з напругою мережі 0,4 кВ з глухозаземленою нейтраллю при аварійному режимі		$I_n = \frac{U_{np}}{R_n} = \frac{2 \cdot U_n}{3 \cdot R_n} = 0,169 (A)$
Двофазний прямий дотик до струмовідних частин з напругою мережі 0,4 кВ		$I_n = \frac{U_n}{R_n} = 0,38 (A)$

Прямий дотик до нульового та фазного дотів з напругою мережі 0,4 кВ		$I_n = \frac{U_\phi}{R_n} = 0,22 (A)$
--	--	---------------------------------------

Величини що були використані в розрахунках:

$$U_\phi = \frac{U_n}{\sqrt{3}} = \frac{380}{\sqrt{3}} = 220 (B) \quad - \text{ фазова напруга електромережі};$$

$$U_n = 380 (B) \quad - \text{ лінійна напруга електромережі};$$

$R_n = 1,5 (k\Omega)$ – опір людини при однофазному дотику, $R_n = 1 (k\Omega)$ – опір людини при двофазному дотику.

Як висновок, можна сказати, що під час використання обладнання СЕС присутні такі небезпечні фактори:

- ураження електричним струмом при дотиканні людиною до струмоведучих частин електричного обладнання;
- робота на висоті.

Аналізуючи вищезазначені розрахунки з таблиці 3.1, можна сказати, що приєднання людини до електричної мережі напругою 0,4 кВ небезпечне для її

життя, так як розрахункові струми в рази перевищують допустимі значення у всіх приведених випадках.

3.2 Заходи щодо приведення умов праці в норму

3.2.1 Засоби захисту від електричного струму мережі за нормального функціонування електрообладнання

Перед тим як ввести в обслуговування та використання електрообладнання сонячної станції напругою 0,4 кВ проводиться контроль ізоляції, для того зробити обладнання придатним для подальшого використання. Вимірювання опору ізоляції проводиться за допомогою мегаомметра випробовувальною напругою 1 кВ, але при цьому опір кожного електричного устаткування не має бути меншим, ніж 500 кОм.

На корпус інвертора необхідно наносити знак безпеки «Обережно! Електрична напруга!». Позначення кожної фази системи змінного струму здійснюється літерами, цифрами та відповідним кольором:

- шини фази А (L1) – жовтим кольором;
- шини фази В (L2) – зеленим кольором;
- шини фази С (L3) – червоним кольором;
- нейтраль (N) – повздовжніми лініями жовтого та зеленого кольорів.

Шини зі сторони змінного струму розташовують так: А-В-С – зверху до низу (горизонтальне розташування); А – найбільш віддалена, В – посередині, С – найближча (вертикальне розташування).

У випадку аварійної ситуації, захист працюючих на СЕС забезпечується заземленням електричного обладнання. Розрахунки заземлення приводяться в п.п. 3.2.2.

Щоб забезпечити безпечне використання електричного обладнання сонячної станції використовують електричні захисні засоби, що приведені в таблиці 3.2.

Таблиця 3.2 – Засоби захисту на СЕС

№	Назва	Тип	Кількість
Основні електричні захисні засоби			
1	Штанга для проколювання кабелю	ППК-10	1
2	Ізолюючі кліщі	К-1000	1
3	Рукавички діелектричні	Зі швами	2 комп.
4	Показчик напруги	УНН-1Ш	1
Додаткові електричні захисні засоби			
5	Калоші діелектричні	-	1 комп.
6	Килим діелектричний	-	1
7	Окуляри захисні	-	1

3.2.2 Розрахунки захисного заземлення

Пробій фази на корпус електричного устаткування створює коротке замкнення, що тягне за собою вимкнення електричного устаткування від мережі.

Заземлення – провідник, або їх сукупність, що з'єднані між собою, та перебувають в безпосередньому контакті з землею через проміжне провідне середовище.

Заземляти обладнання, що має струмоведучі частини та напругу 380/220 В з частотою 50 Гц, необхідно. Грунт – чорнозем, питомий опір якого визначається при умові сухого ґрунту, що має значення 40 Ом*м. Заземлюючим контуром береться прямокутник розмірами 5х15 м. В якості вертикальних стрижнів обираємо кутову сталь №4 розміром 40х40 мм. Було обрано стрижні стандартної довжини – 2,5 м, а відстань між ними становитиме 5 м. Глибина закладання стрижнів – 0,6 м. Природних заземлювачів поблизу нема.

Відповідно до нормативних значень опору заземлюючого пристрою в електроустаткуванні напругою 380/220 В, максимально допустиме значення опору становить не більше 4 Ом. Визначимо необхідний опір штучного заземлення:

$$R_w = \frac{R_n \cdot R_3}{R_n - R_3} \quad (5.1)$$

Так як природній заземлювач відсутній, можна сказати, що опір штучного заземлення має таке значення:

$$R_w = R_3 = 4 \text{ Ом}$$

Питомий опір землі розраховується за формулою:

$$\rho = \rho_{\text{сух}} \cdot \psi \quad (5.2)$$

де, $\rho_{\text{сух}}$ – питомий опір сухого ґрунту;

ψ – кліматичний коефіцієнт опору ґрунту.

$$\rho = 40 \cdot 1,32 = 52,8 \text{ Ом} \cdot \text{м}$$

Опір розтікання струму одного вертикального заземлювача визначається за формулою:

$$R_B = 0,366 \frac{\rho}{l_B} \left(\lg \frac{2 \cdot l_B}{d} + \frac{1}{2} \lg \frac{4 t_B + l_B}{4 t_B - l_B} \right) \quad (5.3)$$

де, d – діаметр стрижня заземлення, в разі використання матеріалів не круглого розрізу, знаходиться еквівалентний діаметр:

$$d_{\text{екв}} = 0,95 \cdot b \quad (3.4);$$

b – довжина сторони еквівалентного заземлювача;

l_B – довжина стрижня заземлювача;

t_B – відстань від середини вертикального стрижня до поверхні ґрунту:

$$t_B = t_0 + \frac{l_B}{2} \quad (5.5)$$

t_0 – відстань від початку заземлювача до поверхні ґрунту.

За формулою 3.4 знаходимо еквівалентний діаметр:

$$d_{\text{екв}} = 0,95 \cdot 0,004 = 0,0038 \text{ м}$$

За формулою 3.5 знаходимо відстань від середини стрижня до поверхні ґрунту:

$$t_B = 0,6 + \frac{2,5}{2} = 1,85 \text{ м}$$

З формули 3.3 визначаємо опір розтікання струму для одного вертикального заземлювача:

$$R_B = 0,366 \frac{52,8}{2,5} \left(\lg \frac{2 \cdot 2,5}{0,038} + \frac{1}{2} \lg \frac{4 \cdot 1,85 + 2,5}{4 \cdot 1,85 - 2,5} \right) = 17,59 \text{ Ом}$$

Приблизна кількість стрижнів:

$$n' = \frac{R_B}{R_{ш}} = \frac{17,59}{4} = 4 \text{ шт.} \quad (5.6)$$

З розмірів контуру визначаємо кількість заземлюючих стрижнів:

$$n' = \frac{P}{a} = \frac{40}{5} = 8 \text{ шт.} \quad (5.7)$$

Отже, обираємо 8 заземлюючих стрижнів.

Коефіцієнт використання вертикальних заземлюючих стрижнів, що розміщені по контуру (8 штук), дорівнює $\eta_B = 0,78$.

Розрахунок опору розтікання струму для декількох заземлюючих стрижнів:

$$R_{роз}^B = \frac{R_B}{n' \cdot \eta_B} = \frac{17,59}{8 \cdot 0,78} = 2,82 \text{ Ом} \quad (5.8)$$

Опір розтікання струму по з'єднуючій смузі знаходиться за формулою:

$$R_{зоп} = 0,366 \cdot \frac{\rho}{l_{зоп}} \cdot \lg \frac{2 \cdot l_{зоп}^2}{t_{зоп} \cdot b} \quad (5.9)$$

де, $l_{зоп}$ – довжина смуги (рівна периметру прямокутника заземлення – 40 м);

$t_{зоп}$ – відстань від середини смуги до поверхні.

$$R_{зоп} = 0,366 \cdot \frac{52,8}{40} \cdot \lg \frac{2 \cdot 40^2}{0,7 \cdot 0,04} = 2,44 \text{ Ом}$$

Опір розтікання струму в з'єднуючій смузі, враховуючи коефіцієнт екранування:

$$R_{роз}^{зоп} = \frac{R_{зоп}}{n_{зоп} \cdot \eta_{зоп}} \quad (5.10)$$

де, $n_{зоп}$ – кількість горизонтально розміщених стрижнів;

$\eta_{зоп}$ – коефіцієнт використання горизонтально розміщених стрижнів, що знаходяться по периметру контуру:

$$R_{роз}^{зоп} = \frac{2,44}{1 \cdot 0,6} = 4,1 \text{ Ом}$$

Еквівалентний опір заземлення в ґрунті розраховується за формулою:

$$R = \frac{1}{\frac{1}{R_{роз}^{зоп}} + \frac{1}{R_{роз}^B}} = \frac{1}{\frac{1}{4,1} + \frac{1}{2,82}} = 1,67 \text{ Ом} (5.11)$$

Отримане значення з формули 3.11 задовольняє дану нерівність:

$$R = 1,67 \leq 4$$

Отже, даний розрахунок було проведено правильно.

3.2.3 Розрахунок влаштування вимкнення захисного апарату

Потужність сонячної електростанції, що знаходиться на даху 21 навчального корпусу КПІ ім. Ігоря Сікорського, складає 72000 Вт, номінальна напруга мережі – 220 В. Номінальний струм автоматичного вимикача, за умови, що струм вимикача має бути більше максимального струму.

З розділу 2 відомо, що було обрано 3 інвертора потужністю по 25 кВт. З їх характеристик видно, який максимально можливий струм вони можуть видавати в мережу. Значення даного струму дорівнює 44,2 А. Тобто потрібно обрати вимикач за струмом, які в сумі видають 3 інвертори, тобто 132,6 А. На рисунку 3.1 зображена часово-струмова характеристика вимикання автоматів різних типів.

Характеристики В, С, D

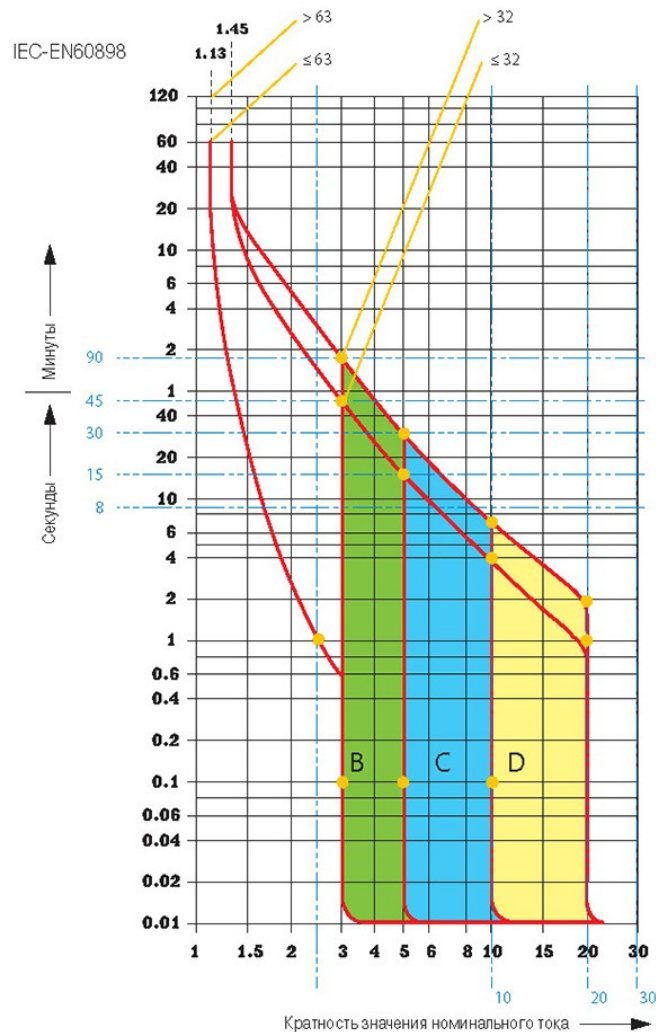


Рисунок 3.1 – Криві відключення різних типів автоматичних вимикачів

Приймаючи до уваги час спрацювання автоматичного вимикача (0,1 с) та область використання (освітлювальні мережі, установки з помірними пусковими струмами) обираємо автоматичний вимикач типу С. Межі спрацювання магнітного роз'єднувача типу С – $5-20 \cdot I_{\text{ном}}$. Такий тип вимикачів є універсальним.

В якості автоматичного вимикача було обрано вимикач АВВ типу Formula A2B 250 TMF з номінальним струмом 40А.

Струм короткого замкнення, при якому відбувається відключення автоматичного вимикача визначається за формулою:

$$I_{\text{вим}} = I_{\text{доп}} \cdot K_{\text{вим}} \quad (5.12)$$

де, $I_{\text{вим}}$ – допустимий номінальний струм автоматичного вимикача (40А);

$K_{\text{вим}}$ – коефіцієнт кратності автоматичного вимикача.

$$I_{\text{вим}} = 160 \cdot 7 = 1120 \text{ А}$$

Переріз фазного провідника обирається за умови максимального допустимого нагрівання $I_{\text{доп}} > I_{\text{мах}}$, де $I_{\text{доп}}$ – довготривало допустимий струм з умов нагрівання. В якості провідників обираємо:

- фазовий провідник – ВВГ 1х16;
- захисний провідник – ВВГ 1х10.

Струм короткого замкнення розраховується за формулою:

$$I_{\text{кз}} = \frac{U_{\phi}}{\frac{Z_{\text{тр}}}{3} + R_{\phi} + R_{\text{нп}}} \quad (5.13)$$

де, $Z_{\text{тр}} = 0,36 \text{ Ом}$;

R_{ϕ} – активний опір фазного провідника;

$$R = \rho_a \cdot \frac{L}{d} \quad (5.14)$$

ρ_a – активний питомий опір жили для фазного та нульового провідника ($1,75 \cdot 10^{-3} \text{ Ом} \cdot \text{мм/м}$);

L – довжина кабелю фазного провідника (100 м для фазного провідника);

d – переріз кабелю;

$$R_{\phi} = 1,75 \cdot 10^{-3} \cdot \frac{100}{16} = 0,0109 \text{ Ом};$$

$R_{\text{нп}}$ – активний опір нульового провідника;

$$R_{\text{нп}} = 1,75 \cdot 10^{-3} \cdot \frac{20}{10} = 0,0035 \text{ Ом};$$

Струм короткого замкнення:

$$I_{\text{кз}} = \frac{220}{\frac{0,36}{3} + 0,0109 + 0,0035} = 1636 \text{ А}$$

Величина струму короткого замкнення більша, ніж величина струму, при якому відбувається відключення автоматичного вимикача. Отже, автоматичний вимикач обрано правильно[26].

3.3 Пожежна безпека

Під час використання фотоелектричного устаткування напругою 0,4 кВ, пожежу можуть створити, такі матеріали: полівінілхлоридна або паперова кабельна ізоляція, ізолюючі матеріали електрообладнання, лако-фарбове покриття електроустановок.

Викликати пожежу можуть такі фактори: коротке замкнення, струмове перевантаження проводів чи кабельної лінії, великі перехідні опори при звуженні на шляху протікання струму, несправність електроустановок, неправильне дотримання працівниками пожежної безпеки на даному об'єкті.

Для уникнення пожежної ситуації необхідно:

- дотримуватись правил пожежної безпеки, всім хто знаходиться на даному об'єкті;
- підтримувати справність захисного відключення електроустановок від мереж.

В якості пожежної сигналізації використовується охоронна сигналізація типу УОТС-11. В ній присутні теплові та димові датчики.

Передбачається стаціонарне розведення водного пожежогасіння з пожежними кранами в ящиках, що розміщуються на стіні.

Головні засоби для гасіння пожежі розташовуються на пожежному щиті: багор, ящик з піском, лом, сокира, щільна тканина, лопата, порошковий вогнегасник типу ОУ-5, вуглекислий вогнегасник типу ОХП-10.

Вимоги пожежної безпеки до будівель та об'єктів різного призначення.

Вимоги пожежної безпеки до будівель та споруд адміністративного призначення:

1. Забезпечення протипожежного стану в адміністративних будівлях та приміщеннях здійснюються відповідно до III розділу даних Вимог.

2. Всі, без виключення, двері евакуаційного призначення та інших виходів завжди мають бути справними та не замкнутими, і відчинятися в напрямку виходу персоналу з будівлі.

3. В адміністративних приміщеннях забороняється використовувати електрообігрівачі.

4. Паперове сміття та ємності для паперових відходів регулярно виносити за межі приміщення в спеціальні місця.

Вимоги пожежної безпеки до центрів з обчислювальною технікою:

1. Пожежна безпека приміщень з обчислювальною технікою та комп'ютерами забезпечується відповідно до Правил охорони праці під час експлуатації комп'ютерних машин та електронно-обчислювальної техніки зазначених в НПАОП 0.00-1.28-10.

2. Приміщення, де знаходяться комп'ютерна та електронно-обчислювальна техніка, мають бути обладнані спеціальними протипожежними засобами.

3. Електричне живлення комп'ютерних та обчислювальних машин повинне мати автоматичне аварійне вимкнення та блокування подачі електричної енергії під час припинення функціонування охолоджувальної системи приміщення.

4. На випадок пожежі вентиляційна система приміщень, обладнаних комп'ютерною та електронно-обчислювальною технікою, повинна мати блокувальний пристрій, що забезпечить її вимкнення.

4.5.1. Проведення заходів щодо пожежної безпеки повинно відповідати вимогам Закону України «Про пожежну безпеку», НАПБ А.01.001 «Правила безпечної безпеки в Україні», ГКД 34.03.303 «Правила пожежної безпеки на підприємствах, організаціях та компаніях енергетичної галузі України» та

деяким іншим галузевим нормативним актам, що описують питання пожежної безпеки.

4.5.2. Проведення заходів щодо пожежної безпеки є основною частиною діяльності на підприємствах усіх працівників. Відповідальність покладається на керівників та на уповноважених осіб.

На будь-якому об'єкті, в залежності від кількості працівників та пожежонебезпечних об'єктів, є необхідність створення служби пожежної безпеки, або призначена уповноважена особа, що буде відповідати за пожежну безпеку.

4.5.3. Експлуатація та улаштування устаткування або будь-яких споруд мають відповідати вимогам зазначеними в нормативних документах з пожежної безпеки.

4.5.4. Необхідність встановлення протипожежного режиму та інструкцій з пожежної безпеки, незалежно від типу об'єкта, враховуючи пожежну небезпеку:

- загальнооб'єктна;
- для окремих цехів, лабораторій, спеціальних приміщень тощо;
- використання системи водопостачання, приладів виявлення та гасіння вогню;
- робота з устаткуванням та приладами, а також проведення робіт будь-якого виду мають бути пожежобезпечними.

Інструкції щодо пожежної безпеки проводяться та розробляються на підставі діючих законів та правил, а також з огляду на особливості споруди, процесів роботи та експлуатації. На додаток до цього, має бути складений план гасіння пожежі, що повинен розроблятися разом зі спеціалістами пожежної охорони Міністерства надзвичайних ситуацій. Усі документи мають бути затверджені в необхідному порядку.

4.5.5. Кожен, хто працює на підприємстві, має обов'язково знати та виконувати вимоги правил пожежної безпеки та встановлений протипожежний

режим на підприємстві. Кожен працівник повинен слідкувати за своїми діями та зупиняти дії інших працівників у разі небезпеки загоряння та пожежі.

4.5.6. Згідно з вимогами нормативних документів, працівники мають проходити протипожежний інструктаж, поновлювати та модернізувати знання з пожежної безпеки, проходити тренування з пожежної безпеки, та перевіряти знання правил пожежної безпеки.

Період проведення та обсяг тренувань визначаються нормативними документами. Враховують необхідність навичок пожежогасіння, в залежності від типу роботи, що виконують працівники.

4.5.7. При виникненні пожежі головною особою, що відповідає за гасіння вогню на своєму підрозділі, є уповноважена особа, яка визначається інструкцією з пожежної безпеки.

Ця особа, по приїзду пожежного підрозділу, повинна інформувати про особливості об'єкта та про дії, які були вжиті для гасіння пожежі. Повинна бути надана інформація про уповноважену за пожежогасіння особу, подати дозвіл на гасіння пожежі на об'єкті та максимально допомагати власними силами та додатковими засобами.

4.5.8. На підприємствах, де кількість працівників перевищує 50 осіб, рішенням робітників підприємства створюють спеціальні комісії з пожежної безпеки, обирають керівника, іноді формують волонтерські пожежні угруповання, які згідно з нормативними актами, виконують дану роботу.

4.5.9. Установки виявлення пожежі та її гасіння, обслуговуються спеціальним персоналом, який пройшов тестування, згідно з інструкціями.

Даний персонал, що відповідає за протипожежні установки, визначаються указом на підприємстві.

Засоби, що в першу чергу використовуються для гасіння пожежі, повинні бути постійно в робочому стані, на випадок загоряння на підприємстві. Вони обслуговуються відповідно до інструкцій, що надані в нормативних документах.

4.5.10. Відповідальність, за організацію протипожежної безпеки, несуть:

- керівництво – за загальний стан об'єкта та організацію протипожежних заходів та вимог, також несуть відповідальність за добровольчі угруповання з пожежної безпеки;
- технічні керівники – за роботу комісії з пожежно-технічного стану, технічний стан протипожежних засобів та автоматичних систем пожежогасіння, за виконання нормативних вимог з протипожежної організації та підготовку робочого персоналу;
- інженери та керівники підрозділів – за стан закріплених за ними приміщень та ділянок, з протипожежної системи, а також зв підготовку робочого персоналу.

4.5.11. Кожна ситуація з загорянням повинна розслідуватись спеціальною комісією щодо встановлення причини, збитків підприємства, тих хто був винен у виникненні пожежі. Ця комісія повинна розробляти протипожежні заходи для інших непошкоджених об'єктів підрозділів відповідно до ГКД 34.20.801 «Інструкція зі службового розслідування, первинного обліку та аналізу пожеж, які сталися на об'єктах».

Вимоги щодо пожежної безпеки:

1. До будь-яких будівель завжди повинен бути вільний доступ.
2. На підприємстві, у всіх приміщеннях мають бути таблички, де написано про місцезнаходження засобів з гасіння пожежі.
3. Для будь-яких споруд мають бути визначені категорії вибухової та пожежної небезпеки за ОНТП 24-86, а також визначені класи зон за нормативним документом ДНАОП 0.00.1-32-01 «Правила будови електроустаткування»
4. Працівники зобов'язані вміти використовувати вогнегасники та пожежними кранами, що розташовані в приміщенні.

5. Вентиляційні проходи, електрощитові та технічні кімнати не мають використовуватись для зберігання старого обладнання, меблів та інших предметів.

6. Електрощитові приміщення, вентиляційні проходи, двері горищ повинні бути замкнені. У випадку необхідності, на них встановлюють таблички з вказаним місцем зберігання ключів.

7. Перевірка справності вогнегасників та засобів гасіння пожежі повинні перевірятися не рідше одного разу на рік, та складати акт про їх перевірку.

8. Приміщення, де знаходиться електрообладнання, комп'ютери та обчислювальна техніка, оснащуються спеціальними вогнегасниками (вуглекислотними чи порошковими), з урахуванням, що на кожні 20 кв. м ємність вогнегасника – 3 л. Після закінчення роботи всі електротехнічні прилади повинні вимикатися від електричної мережі.

9. В приміщенні забороняється розводити багаття та відкритий вогонь.

Діяльність працівників у разі виникнення загоряння:

1. Виявивши пожежу або перші ознаки загорання кожен робітник зобов'язаний:

- одразу повідомляти про надзвичайну ситуацію в службу пожежної безпеки за телефоном «101», сказавши при цьому адресу де знаходиться об'єкт, кількість поверхів в будівлі, місце виникнення загорання, чи є на місці пожежі люди, а також власне прізвище та ім'я;

- по можливості, вжити заходи щодо евакуації персоналу та цінних матеріалів та речей, гасити пожежу наявними вогнегасниками та іншими засобами, що призначені для гасіння пожежі;

- повідомити власника підприємства, або керівника, про виникнення пожежі.

2. Повідомлений про виникнення пожежі керівник, або власник, повинен:

- впевнитись у тому, що була викликана служба пожежної безпеки;

- впевнитись у тому, що весь персонал проінформований про пожежну небезпеку;
- за необхідності, вимкнути живлення струмоприймачів та вентиляції;
- у разі загрози здоров'ю та життю персоналу, необхідно організувати евакуацію, та вивести людей із зони ризику тих осіб, які не займаються ліквідацією пожежної ситуації;
- проконтролювати правильність дії персоналу, який ліквідує пожежу, згідно Правил техніки безпеки;
- забезпечити зустріч служби пожежної безпеки, допомогти їм знайти місце пожежі, та надати допомогу при її ліквідації.

3. Забезпечити безперешкодний доступ рятувальній службі, до їх прибуття на місце.

4. У разі наявності потерпілих, надавати першу медичну допомогу, викликати карету швидкої допомоги за номером «103». Якщо нема можливості викликати лікаря, то вжити заходи для переміщення потерпілих до найближчого медичного закладу[27].

3.4 Оцінка впливу на навколишнє середовище

3.4.1 Візуальний вплив (на ландшафт)

Дана сонячна станція розташовується на даху 21 корпусу КПП ім. Ігоря Сікорського, тому впливу на ландшафт не має. Видно дану станцію буде тільки з будівель вищих за даний корпус або з висоти пташиного польоту

3.4.2 Електромагнітне випромінювання

Сонячна станція як самостійний об'єкт не чинить істотного впливу у вигляді електромагнітного випромінювання високого рівня. Джерела впливу, що мають розглядатися – це проєктована кабельна лінія та трансформаторна підстанція, до якої відбувається підключення СЕС.

3.4.3 Вплив на землекористування

Так як дана фотоелектрична станція розміщена на даху, вона не має ніякого впливу на землекористування, бо займає площу даху будівлі, що вже займає певну земельну площу.

3.4.4 Вплив на флору та фауну

Занепокоєння відносно впливу сонячної станції на об'єкти фауни пов'язані з ризиком для рукокрилих та птахів при перельоті над ділянкою фотоелектричної станції та створенням відблисків сонячного світла від поверхні фотопанелей. Так як сонячна електростанція розташовується на даху, вплив на тварин та рослини, що живуть на поверхні та під землею мінімальний та має сенс при прокладанні кабелю до трансформаторної підстанції.

На рисунку 3.2 зображена карта міграції перелітних птахів.



Рисунок 3.2 – Шляхи міграції птахів поблизу розташування сонячної електростанції

Ділянку розміщення СЕС оминають причорноморсько-азовський (мартин, крячки), дніпровський (журавель сірий, чернеть морська та чубата) та поліський північноширотний (білолоба гуска, лебідь-шипун, крижень) коридори міграції птахів, залишаючи ділянку сонячної станції в стороні[28].

3.5 Можливі впливи при будівництві та експлуатації сонячної станції на навколишнє середовище

3.5.1 Клімат та мікроклімат

Вплив сонячної станції на клімат та мікроклімат відсутній

3.5.2 Вплив на повітряне середовище

Під час будівництва СЕС вплив на повітряне середовище виникає в основному під час будівництва, коли забруднюючим фактором є будівельні техніка та механізми. Також це має сенс при виготовленні самих сонячних панелей, а під час самої експлуатації – вплив відсутній.

3.5.3 Вплив на водяне середовище

Вплив на водяне середовище відсутнє, так як станція знаходиться на даху, в неї немає виходу до відкритих водойм та до підземних водяних джерел, а також на станції нема санітарних приміщень, так як всі вони розташовані в корпусі.

3.5.4 Вплив на ґрунти

Вплив на ґрунти має сенс при прокладанні кабельної лінії до трансформаторної підстанції. В процесі прокладання кабелю, викопується траншея глибиною до 1 метра та шириною приблизно пів метра, а найближча підстанція знаходиться біля навчального корпусу №20 КПІ ім. Ігоря Сікорського на відстані 183 м від місця встановлення сонячної станції. Тобто об'єм ґрунту, який необхідно викопати становить 108,5 м³. Після прокладання кабелю ґрунт засипається назад та засівається покривною рослинністю.

3.5.5 Влив на рослинний і тваринний світ, заповідні зони

Сонячна станція будується на даху навчального корпусу 21 КПІ ім. Ігоря Сікорського. Дана будівля побудована на земельній ділянці, що не віднесена до об'єктів природно-заповідного фонду країни.

3.5.6 Вплив на соціальне середовище

При будівництві та експлуатації сонячної електростанції створюються додаткові робочі місця та проходить розвиток альтернативної енергетики.

У випадку будівництва сонячної електростанції на даху навчального корпусу №21 КПІ ім. Ігоря Сікорського, це призводить до переведення корпусу до деякої енергонезалежності.

3.5.7 Вплив на техногенне середовище

Вплив сонячної станції на пам'ятки архітектури, культури та історії, зони рекреації, культурного ландшафту – відсутній, так як СЕС розташовується на даху навчального корпусу.

3.6 Оцінка небезпеки під час користування фотоелектричним обладнанням

В даному розділі приводиться аналіз шкідливих та небезпечних факторів під час користування фотоелектричним обладнанням потужністю 72 кВт, показано можливі ситуації попадання людини під дію електричного струму та оцінена небезпека. Щоб зробити умови праці безпечнішими, проводять спеціальні профілактичні заходи, наприклад: використання знаків безпеки «Обережно! Електрична напруга!», маркування деяких частин обладнання, дотримання відповідного колоруювання та розміщення неізольованих струмовідних частин. Додатково для убезпечення умов праці було проведено розрахунок заземлення та приведені правила пожежної безпеки.

Шкідливий вплив на навколишнє середовище – мінімальний.

ВИСНОВКИ ТА РЕЗУЛЬТАТИ РОБОТИ

В ході дипломної роботи було досягнуто початкової мети. Мною було:

- розроблено бюджетну систему моніторингу на базі мікрокомп'ютера Raspberry Pi, який виконує роль збору, обробки та передачі даних через мережу інтернет
- створено і налаштовано платформу для збору, обробки, відображення даних та забезпечено віддалений доступ до них.
- розроблено інтуїтивно зрозумілий інтерфейс програми для ПК користувача

Хоча в подальшому дана система моніторингу може бути вдосконалена, вже зараз вона демонструє великі можливості та широкий функціонал моніторингової системи віддаленого доступу, використовуючи мікрокомп'ютер Raspberry Pi 3. Всі необхідні дані збираються за допомогою Raspberry Pi і відображаються графічно на персональному комп'ютері. Для створення графічного інтерфейсу користувача (GUI) на комп'ютері була залучена бібліотека Tkinter мови Python. Крім того, розроблена система не тільки може контролювати і вимірювати необхідні дані, але і будувати певні залежності величин, які згодом можна дистанційно аналізувати. Ця система моніторингу має перевагу в гнучкості збору даних. Крім того, це віддалений моніторинг, дані з мікрокомп'ютера Raspberry Pi можуть передаватися по веб-серверу на персональний комп'ютер. Система може розглядатися як система моніторингу з низькими витратами і може використовуватися для інших застосувань.

Зокрема, були вироблені навички роботи з мікрокомп'ютером Raspberry Pi 3 Model B+, досліджено роботу 4-канального 16-розрядного АЦП ADS1115, покращено знання написання коду мовою Python.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Р.А.Розелін, П.Суматі Сонячна енергія для безпроводних сенсорних мереж в програмах моніторингу навколишнього середовища — Міжнародний журнал науково-дослідницьких публікацій, том 4, випуск 8, серпень 2014 року ISSN 2250-3153
2. Ш. Менделл і Д. Сінх Збір та обробка даних фотомодулів у реальному часі, використовуючи мікрокомп'ютер Raspberry Pi і подальший моніторинг напруги сонячної панелі — Міжнародний журнал приладів та систем керування (IJCS) Том 7, № 3, липень 2017
3. А. Аль-Дахуд, М. Фезарі, Ф.Белхучет Моделювання та симуляція моніторингової системи на віддаленому доступі використовуючи Raspberry Pi для сонячних панелей та інверторів 2014 — Перша Міжнародна Конференція Інформаційних Систем, 978-0-7695-5198-2/14
4. A. Gaevskii. “Method for determining parameters of PV modules in field conditions”, 2019 IEEE 6th International Conference on Energy Smart Systems, Kyiv, Ukraine, 2019.
5. Р. П. Шинді, Й. Н. Гатлавар Автоматизована система моніторингу та панель керування для сонячних систем використовуючи безпроводну сенсорну мережу — Міжнародний журнал досліджень та відкриття технологій, Спеціальна державна конференція “ACGT 2015”, том 3, E-ISSN: 2321-9637, питання 3, стр.47-51, лютий 2015
6. Гаєвський О.Ю., Делев Д.С., Чорномурко М.О., Бодняк В.В. Ефективність однофазного інвертора фотоелектричної станції та гармонічні спотворення при різних рівнях навантаження// Енергетика: економіка, екологія, технологія. – 2017.-№7, с.28-34.
7. С. Фердуш, К. Лі Конструкція безпроводних систем сенсорних мереж за допомогою Raspberry Pi і Arduino для моніторингу ФЕС - Президія комп'ютерних наук, том 34, с. 103-110, 2014.

8. М. Ф. Саннер Python: мова програмування для інтеграції та розробки програмного забезпечення — Журнал молекулярної графіки та моделювання, том 17, с. 57-61, 1999.
9. Р. О. Томас, К. Раджасекаран Дистанційне керування за допомогою Raspberry Pi — Міжнародний журнал нових технологій в галузі комп'ютерних наук та електроніки (IJETCSE) ISSN: 0976-1353, том. 8, випуск 1, с. 186-189, квітень 2014.
10. Програмування мовою Python
<https://www.tutorialspoint.com/python/>
<https://www.w3schools.com/python/>
<https://docs.python.org/3/tutorial/>
11. Конструкційні особливості Raspberry Pi
<http://www.cnx-software.ru/2016/02/29/raspberry-pi-3-board-is-powered-by-broadcom-bcm2827-cortex-a53-processor-sells-for-35/>
<https://www.opennet.ru/opennews/art.shtml?num=43400>
12. Знайомство с GPIO портами Raspberry Pi <http://kip-world.ru/pervoe-znakomstvo-s-gpio-v-raspberry-pi.html>
13. Первинне включення Raspberry Pi 3
<https://habr.com/ru/post/149890/>
14. Налаштування SSH з'єднання між Raspberry та комп'ютером
<https://www.raspberrypi.org/magpi/ssh-remote-control-raspberry-pi/>
15. Підключення по віддаленому доступу до Raspberry Pi через комп'ютер використовуючи додаток PUTTY
<https://www.ssh.com/ssh/putty/windows/>
16. Налаштування веб-сервера Samba
<https://www.raspberrypi.org/magpi/samba-file-server/>
17. Оптимізація роботи мікрокомп'ютера Raspberry Pi 3
<https://romantelychko.com/blog/1611/>

18. Вимірювання аналогових значень за допомогою Raspberry
<https://www.raspberrypi.org/forums/viewtopic.php?t=57480>
19. Бібліотека для АЦП ADS1x15
https://github.com/adafruit/Adafruit_CircuitPython_ADS1x15
20. Підключення АЦП ADS1115 до мікрокомп'ютера Raspberry Pi
<https://learn.adafruit.com/adafruit-4-channel-adc-breakouts/python-circuitpython>
21. Знайомство з бібліотекою для графічного інтерфейсу Tkinter
https://www.python-course.eu/python_tkinter.php
<https://pythonprogramming.net/python-3-tkinter-basics-tutorial/>
22. Робота за даними файлів
<https://realpython.com/python-csv/>
23. Знайомство з бібліотекою для побудови графіків з MATLAB оболонкою matplotlib
<https://matplotlib.org/3.1.0/tutorials/index.html>
24. Знайомство з бібліотекою NumPy для проведення розрахунків на мові Python <https://www.guru99.com/numpy-tutorial.html>
25. Допоміжна бібліотека Pillow для збереження та обробки зображень
<https://pypi.org/project/Pillow/2.2.1/>
26. Закон України «Про альтернативні джерела енергії», редакція від 16.07.2015 [Електронний ресурс] / Режим доступу:
<http://zakon3.rada.gov.ua/laws/show/555-15>. – С 1.
27. Носенко Ю. Сучасні сонячні технології / Юрія Носенко // Життєве середовище. - №18(241) – 2012. – С 1.
28. Третьяков О. Охорона праці: навч. посібник / О. В. Третьяков, В. В. Зацарний, В. Л. Безсонний ; ред. К. Н. Ткачук. - К. : Знання, 2010. - 168 с.
– Тема 7.