

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інформаційних технологій та комп'ютерної інженерії

(повне найменування інституту, назва факультету (відділення))

Кафедра обчислювальної техніки

(повна назва кафедри (предметної, циклової комісії))

## Пояснювальна записка

до магістерської кваліфікаційної роботи

магістр

(освітньо-кваліфікаційний рівень)

на тему Система метеорологічних досліджень на базі мінікомп'ютера

Виконав: студент 2 курсу, групи 1КІ-18м

напряму підготовки (спеціальності)

123 Комп'ютерна інженерія

(шифр і назва напряму підготовки, спеціальності)

Осипчук С. В.

(прізвище та ініціали)

Керівник Богомолів С.В.

(прізвище та ініціали)

Рецензент Поплавський А.В.

(прізвище та ініціали)

## РЕФЕРАТ

Магістерська дипломна робота присвячена розробці системи збору метеорологічних даних на базі мінікомп'ютера. Дана система обробляє дані з електронних сенсорів температури, вологості та тиску, передає їх на опрацювання до мінікомп'ютера, який в свою чергу формує всю отриману інформацію в HTML сторінку та розміщує на сервері під своїм керуванням. В залежності від отриманих даних можлива зміна станів інших пристроїв підключених до системи. Під час виконання магістерської кваліфікаційної роботи було проаналізовано основні можливості системи та використання їх у різних сферах, проаналізовано аналоги, виявлено їх недоліки, з урахуванням яких створено нову розробку. Залежно від отриманих даних стан інших пристроїв, підключених до системи, може змінюватися.

Розроблено електричну та структурну схеми. В додатках наведена електрична принципова схема приладу, перелік всіх її елементів і лістинги скриптів для обробки даних та розміщення на власному сервері.

Ключові слова: мінікомп'ютер, метеорологічна система, метеостанція, датчик.

## ABSTRACT

The master's thesis is devoted to the development of a system of meteorological data collection based on a minicomputer. The system processes data from electronic sensors of temperature, humidity and pressure, for processing to a minicomputer, which in turn generates all the received information into an HTML page and places it on a server under its control. Depending on the data obtained, the states of other devices connected to the system may change. During the performance of the master's qualification work, the basic capabilities of the system and their use in different fields were analyzed, analogues were analyzed, their shortcomings were identified, in view of which new development was created. Depending on the data obtained, the states of other devices connected to the system may change.

Electric and structural schemes have been developed. The annexes show the electric principle diagram of the device, a list of all its elements and listing scripts for data processing and placement on its own server.

Key words: minicomputer, meteorological system, metrological system, weather station, sensor.

## ЗМІСТ

ВСТУП.....	8
1 ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ ДО ПОБУДОВИ СИСТЕМ МЕТЕОРОЛОГІЧНИХ ДОСЛІДЖЕНЬ .....	10
1.1 Типи давачів сигналу .....	10
1.2 Огляд типів датчиків вимірювань .....	15
1.3 Огляд існуючих видів метеостанцій .....	18
1.4 Огляд приладів для визначення різноманітних метеорологічних показників .....	21
1.5 Класифікація та огляд видів мінікомп'ютерів .....	31
1.6 Обґрунтування створення системи метеорологічних досліджень на базі мінікомп'ютерів .....	32
1.7 Висновок .....	33
2 ПРОЕКТУВАННЯ СИСТЕМИ МЕТЕОРОЛОГІЧНИХ ДОСЛІДЖЕНЬ.....	34
2.1 Вибір мінікомп'ютера.....	34
2.2 Розробка структурно-функціональної схеми системи метеорологічних досліджень .....	38
2.3 Вибір електронних компонентів .....	39
2.4 Огляд та вибір інтерфейсів підключення .....	44
2.5 Розробка електричної принципової схеми .....	54
2.6 Висновок .....	55
3 МАКЕТУВАННЯ ТА НАЛАШТУВАННЯ СИСТЕМИ МЕТЕОРОЛОГІЧНИХ ДОСЛІДЖЕНЬ .....	56
3.1 Розробка алгоритму функціонування системи .....	56
3.2 Підготовка програмного забезпечення.....	56
3.3 Налаштування програмного забезпечення отримання даних .....	59

					08-23.МКР.012.00.000 ПЗ		
Змн.	Лист	№ докум.	Підпис	Дата			
Розробив	Осипчук С.В.				Літ.	Аркуш	Аркушів
Перевірив	Богомолов С.В.						
Реценз.	Яремчук Ю. Є.				ВНТУ, 1КІ-18м		
Н. контр.							
Затверд.	Мартинюк Т. Б.						
					Система метеорологічних досліджень на базі мінікомп'ютера		

	7
3.4 Налаштування серверу та передачі інформації на сайт .....	63
3.5 Реалізація взаємодії з іншими елементами системи.....	66
3.6 Налаштування роботи з пристроями через інтерфейс 1-wire.....	69
4 ЕКОНОМІЧНА ЧАСТИНА .....	76
4.1 Оцінювання комерційного потенціалу розробки .....	76
4.2 Прогнозування витрат на виконання наукової роботи та впровадження результатів .....	82
4.3 Прогнозування комерційних ефектів від реалізації результатів розробки	87
4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності .	88
4.5 Висновки. ....	93
ВИСНОВКИ .....	94
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	95
ДОДАТОК А .....	97
ДОДАТОК Б - ЕЛЕКТРИЧНА ПРИНЦИПОВА СХЕМА .....	<b>ОШИБКА!</b>
<b>ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.</b>	
ДОДАТОК В - ПЕРЕЛІК ВИКОРИСТАНИХ ЕЛЕМЕНТІВ .....	1012
ДОДАТОК Г - ЛІСТИНГ БІБЛІОТЕКИ РОБОТИ З LCD ДИСПЛЕЄМ.....	1023
ДОДАТОК Д - ЛІСТИНГ БІБЛІОТЕКИ РОБОТИ З ДАТЧИКОМ ВМЕ280..	1078
ДОДАТОК Е - РОБОТА САЙТУ З ОБРОБЛЕНИМИ ДАНИМИ.....	11213
ДОДАТОК Ж - АЛГОРИТМ РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	11415

					08-23.МКР.012.00.000 ПЗ	Арк. 7
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

Актуальність роботи. В даний час великого поширення набувають автоматизовані комп'ютерні системи, які працюють без втручання оператора. Більшість виробництв прагнуть скоротити кількість людських ресурсів та все частіше піддаються комп'ютеризації. Дана тенденція спрямована на спрощення виробництва, зменшення шансів похибки, мінімізації затрат та для збільшення масштабів виробництва.

З початку, на промисловості одноплатні комп'ютери використовувались переважно для обробки та перетворення інформації з комп'ютерної мови на зрозумілу операторам, але з часом все більше роботи стали виділяти на нелюдський, незалежний ресурс. Еволюція даних пристроїв привела їх до появи мінікомп'ютерів – пристроїв великої розрахункової здатності здатних обробляти великі об'єми інформації, комбінувати та систематизувати данні. Найважливішим аспектом мінікомп'ютерів все одно залишається багатофункціональність, тобто вони можуть використовуватись для керування багатьма пристроями, підключеними до однієї системи та реагувати на великий стек команд набагато краще ніж мікроконтролери.

Маючи велику обчислювану здатність, невеликі розміри, можливість роботи з багатьма різними інтерфейсами та властивість працювати на віддалених відстанях автономно, зберігаючи на собі інформацію з можливістю доступу до неї через мережу інтернет, не дивно що вони почали повільно завойовувати ринок.

Мета роботи. Забезпечення опрацювання сигналів давачів з різнотипними інтерфейсами і застосуванням сучасних мінікомп'ютерів.

Наукова новизна. Запропоновано використовувати мінікомп'ютери для обробки інформації системи давачів сигналів для контролю або зміни стану інших різноманітних систем, які підключені до даної системи та можуть вплинути на покази даних давача сигналу.

Об'єктом дослідження є процес отримання інформації за допомогою метеорологічних датчиків через різні інтерфейси, подальша його обробка та використання результатів для коригування роботи іншого обладнання.

Предметом магістерської кваліфікаційної роботи є вдосконалення систем збору метеорологічних даних та організація впливу давачів сигналів на інші системи.

Метою магістерської кваліфікаційної роботи є забезпечення опрацювання сигналів давачів з різнотипними інтерфейсами і застосуванням сучасних мінікомп'ютерів. Задачами магістерської кваліфікаційної роботи є:

- a) провести огляд існуючих систем метеорологічних досліджень та сформулювати основні ідеї створення власної;
- b) здійснити дослідження та сформулювати переваги створення метеостанції на базі мінікомп'ютера;
- c) розглянути основні мінікомп'ютери на ринку та обґрунтувати вибір для даної системи;
- d) здійснити вибір додаткових елементів та побудувати схему електричну принципову ;
- e) розробити алгоритми та програмне забезпечення (ПЗ).

# 1 ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ ДО ПОБУДОВИ СИСТЕМ МЕТЕОРОЛОГІЧНИХ ДОСЛІДЖЕНЬ

## 1.1 Типи давачів сигналу

Генератор сигналів - це будь-який пристрій, який створює електронні сигнали. Існує безліч різних типів з різноманітним призначенням та застосуванням. Функціональні генератори, генератори довільних сигналів і генератори векторних сигналів є загальними типами спеціалізованих генераторів сигналів.

Хвильовий генератор сигналів використовується для генерації електричних форм хвиль у широкому діапазоні сигналів. До поширених типів виходів форм хвиль належать синусоїда, квадратна хвиля, рампа або трикутна хвиля, пульсова хвиля, хвиля серцевого малюнка, пульсові хвилі Гаусса та довільні хвилі. Ці форми хвиль можна вводити у випробувальний ланцюг та аналізувати, щоб переконатися, що пристрій працює належним чином.

Для більшої гнучкості багато форм хвиль можна завантажувати як кроки та розігрувати в послідовності, що включає повтори, стрибки та тригери, щоб визначити порядок та кількість повторень кожного кроку. Використовуючи комбінацію довільних форм хвиль та послідовності, можна досягти надзвичайно складного та динамічного відтворення форми хвилі.

Генератор довільних функцій має заздалегідь заданий перелік форм хвиль або шаблонів, які він може відтворювати. Оператор може змінювати параметри форм хвиль, такі як швидкість відтворення, амплітуду та зміщення, або додавати деякі основні спотворення чи модуляції.

Генератор векторних сигналів спеціалізується на створенні радіочастотних сигналів з аналоговою та цифровою схемами модуляції у таких форматах, як QAM, QPSK, FSK, BPSK та OFDM. Векторні генератори сигналів зазвичай використовуються для тестування чутливості приймача.

Ринок пропонує широкий спектр генераторів сигналів, що охоплюють широкий спектр застосувань: від реплікації сенсорних сигналів до створення RF



та найшвидших високошвидкісних послідовних сигналів даних. Кожен генератор сигналів може створювати практично необмежену кількість сигналів - аналоговий або цифровий, ідеальний або спотворений.

Генератор імпульсів, як і випливає з назви - це форма генератора сигналів, яка створює імпульси. Ці генератори сигналів часто мають форму логічних генераторів імпульсів, які можуть видавати імпульси із змінною затримкою, а деякі навіть пропонують змінний час підйому та падіння.

Імпульси часто потрібні при тестуванні різних цифрових, а іноді і аналогових схем [2].

Для використання деяких датчиків в роботах необхідно додатково застосовувати аналогово-цифрові перетворювачі. Різноманітні області застосування АЦП - це вимірювальні та контрольні системи, промислові приладобудування, системи зв'язку та всі інші сенсорні системи. Класифікація АЦП на основі таких факторів, як продуктивність, швидкість передачі даних, потужність, вартість тощо.

Однією з головних переваг перетворювача АЦП є висока швидкість збору даних навіть на мультиплексованих входах. З винаходом широкого спектру інтегральних мікросхем АЦП (ІС) збір даних від різних датчиків стає більш точним та швидшим. Динамічні характеристики високоефективних АЦП - це покращена повторюваність вимірювань, низьке енергоспоживання, точна пропускна здатність, висока лінійність, відмінне співвідношення сигнал / шум (SNR) тощо.

Практично кожен параметр, що вимірюється навколишнім середовищем, має аналогічну форму, наприклад температуру, звук, тиск, світло тощо. Якщо розглядати систему моніторингу температури, в ній отримання, аналіз та обробка даних про температуру від датчиків є неможливою для цифрових комп'ютерів та процесорів. Тому цій системі потрібен проміжний пристрій для перетворення аналогових температурних даних у цифрові дані для зв'язку з цифровими процесорами, такими як мікроконтролери та мікропроцесори. Аналого-цифровий

перетворювач (АЦП) - це електронна інтегральна схема, яка використовується для перетворення аналогових сигналів, таких як напруги, в цифрову або двійкову форму, що складається з 1s і 0s. Більшість АЦП приймають вхід напруги від 0 до 10В, -5В до + 5В тощо, і відповідно виробляють на цифровий вихід якесь двійкове число. Приклад простого АЦП зображено на рисунку 1.1.

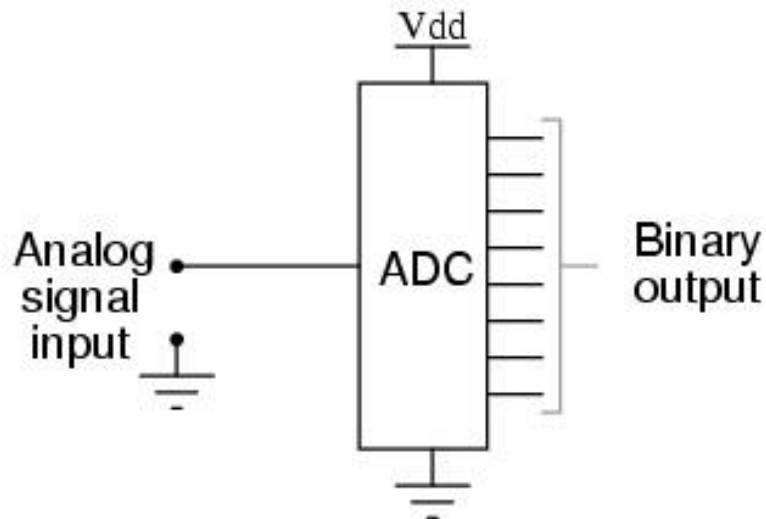


Рисунок 1.1 – Вигляд АЦП

Аналого-цифровий перетворювач відбирає аналоговий сигнал на кожен падаючий або зростаючий край тактового зразка. У кожному циклі АЦП отримує аналоговий сигнал, вимірює та перетворює його в цифрове значення. АЦП перетворює вихідні дані в ряд цифрових значень шляхом наближення сигналу з фіксованою точністю.

У АЦП два фактори визначають точність цифрового значення, яке фіксує вихідний аналоговий сигнал. Це рівень квантування або швидкість передачі даних та швидкість вибірки. Нижче, на рисунку 1.2 зображено, як відбувається аналогове цифрове перетворення. Швидкість передачі даних визначає роздільну здатність оцифрованого виводу, і як можна побачити на малюнку нижче, для перетворення аналогового сигналу використовується 3-бітний АЦП.

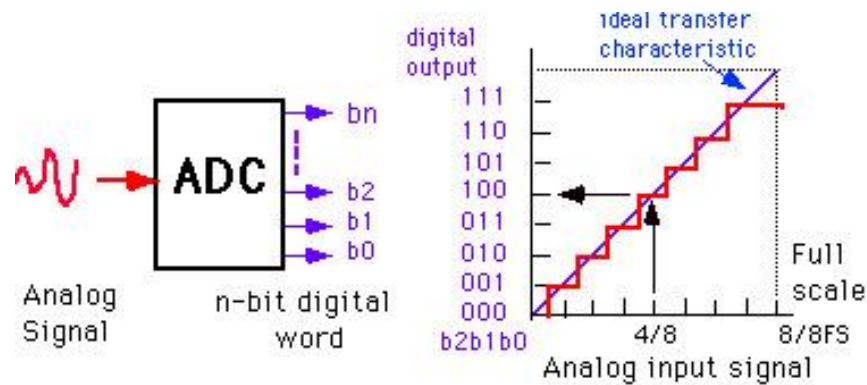


Рисунок 1.2 – Процес перетворення аналогового сигналу

Припустимо, що один вольт сигнал повинен бути перетворений з цифрового за допомогою 3-бітного АЦП, як показано нижче. Таким чином, для отримання виходу 1В доступно загалом  $2^3 = 8$  підрозділів. Отримані результати  $1/8 = 0,125$  В називають мінімальним рівнем зміни або квантування, представленим для кожного поділу як 000 для 0 В, 001 для 0,125, і так само до 111 для 1 В. Якщо збільшити бітові швидкості до 6, 8, 12, 14, 16 і т.д., буде отримано кращу точність сигналу. Таким чином, швидкість передачі бітів або квантування дає найменшу зміну виходу значення аналогового сигналу, що є результатом зміни цифрового подання.

Припустимо, якщо сигнал становить приблизно 0-5В, і використовується 8-бітний АЦП, то двійковий вихід 5В становить 256. А для 3В – 132.

Існує абсолютна ймовірність неправильного подання вхідного сигналу на стороні виходу, якщо він відбирається з вибіркою на іншій частоті, ніж необхідна. Тому ще одне важливе врахування АЦП - це частота вибірки.

Необхідно звернутися до розгляду подвійного перетворювача. У цьому типі перетворювача АЦП порівняння напруга генерується за допомогою інтегральної схеми, яка формується комбінацією резистора, конденсатора та операційного підсилювача. За заданим значенням  $V_{ref}$ , цей інтегратор генерує пилоу форму хвилі на своєму виході від нуля до значення  $V_{ref}$ . При запуску форми хвилі інтегратора відповідно лічильник починає рахувати від 0 до  $2^n - 1$ , де  $n$  - кількість біт АЦП. Коли вхідна напруга  $V_{in}$  дорівнює напрузі форми хвилі, то схема управління фіксує лічильне значення, яке є цифровим значенням

відповідного аналогового вхідного значення. Цей двосхильний АЦП має відносно середню вартість і низьку швидкодію пристрою. Схема подвійного перетворювача наведена на рисунку 1.3.

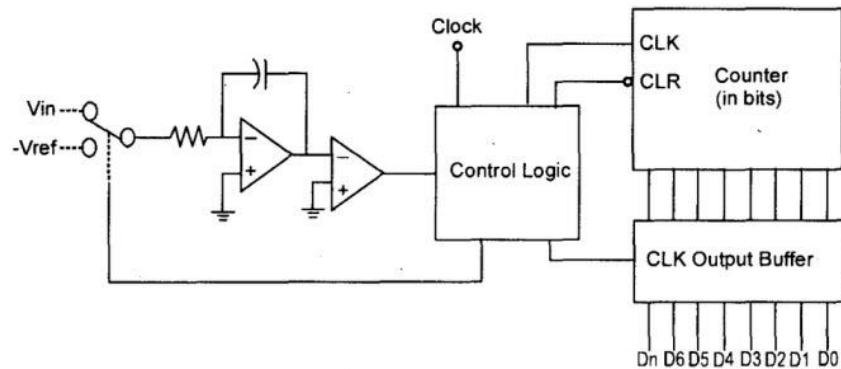


Рисунок 1.3 – Подвійний АЦП

Flash A / D конвертер, який також називають паралельним АЦП, є найбільш широко використовуваним та ефективним АЦП з точки зору його швидкості.

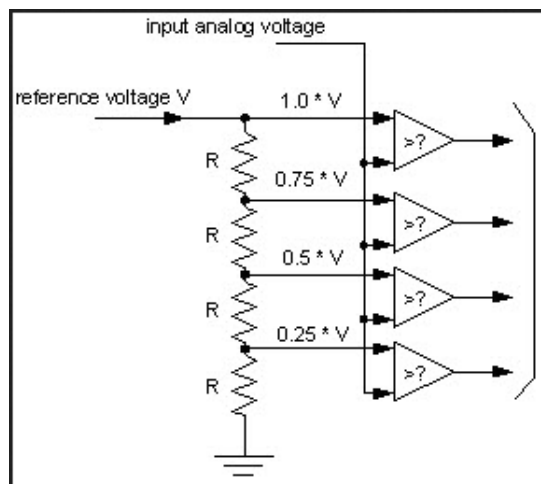


Рисунок 1.4 – FLASH АЦП

Цей аналого-цифровий перетворювач складається з серії компараторів, де кожен порівнює вхідний сигнал з унікальною опорною напругою. На кожному компараторі вихід буде високим, коли аналогова вхідна напруга перевищує опорну напругу. Цей вихід надає декодеру пріоритет для генерації двійкового коду на основі вхідної діяльності вищого порядку шляхом ігнорування інших

активних входів. Цей тип АЦП - це дорогий і швидкісний пристрій, схему якого наведено на рисунку 1.4.

SAR АЦП - це найсучасніший АЦП, який набагато швидший, ніж двонаправлений та флеш-АЦП, оскільки він використовує цифрову логіку, яка перетворює аналогову вхідну напругу до найближчого значення. Схема, що зображена на рисунку 1.5 складається з компаратора, вихідних засувок, реєстру послідовного наближення (SAR) та перетворювача D / A.

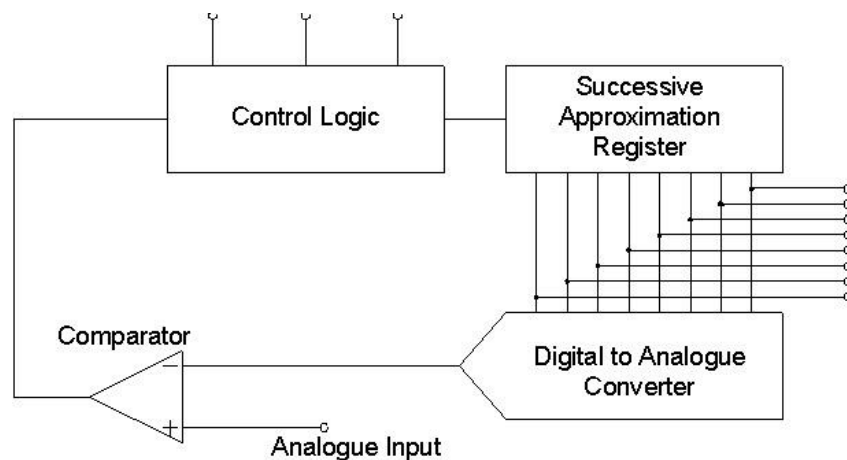


Рисунок 1.5 – SAR АЦП

На початку SAR скидається, і при введенні переходу з низького до високого встановлюється MSB SAR. Потім цей вихід подається на перетворювач D / A, який виробляє аналоговий еквівалент MSB, далі він порівнюється з аналоговим входом  $V_{in}$ . Якщо вихід компаратора має низький сигнал, то MSB буде очищений, інакше MSB буде мати сигнал. Цей процес триває до тих пір, поки всі біти не будуть випробувані, і після Q0 SAR робить паралельні лінії виводу, що містять дійсні дані.

## 1.2 Огляд типів датчиків вимірювань

Автоматизація різноманітних технологічних процесів, ефективне керування різними машинами, агрегатами, механізмами потребує багатьох вимірювань різних фізичних величин.

Можна знайти різні типи датчиків у будинках, офісах, машинах тощо, які працюють на полегшення життя людей, включаючи світло, виявляючи чиюсь присутність, регулюючи температуру в приміщенні, виявляючи дим чи вогонь, роблячи смачну каву, відкриваючи гаражні двері як тільки автомобіль знаходиться біля дверей і багато інших завдань.

Усі ці та багато інших завдань щодо автоматизації можливі завдяки датчикам. Перш ніж перейти розглянути що таке датчик, які існують різні типи датчиків та застосування цих різних типів датчиків, спочатку необхідно розглянути простий приклад автоматизованої системи, який можливий завдяки датчикам (а також багатьох інших компонентів).

Приклад, про який говорилось вище - система автопілотів у літальних апаратах. Практично всі літаки цивільного та військового призначення мають функцію автоматичної системи управління польотом або іноді її називають автопілотом. Приклад взаємодії датчиків наведено на рисунку 1.6.



Рисунок 1.6 – Схема взаємодії датчиків з іншими елементами системи

Автоматична система управління польотом складається з декількох датчиків для різних завдань, таких як контроль швидкості, висоти, положення, стану дверей, наявності перешкод, рівню палива, маневрування та багато іншого. Комп'ютер бере дані з усіх цих датчиків і обробляє їх, порівнюючи їх із заздалегідь розробленими значеннями.

Потім комп'ютер подає керуючий сигнал різним частинам, таким як двигуни, заслінки, руді тощо, які допомагають у плавному польоті. Поєднання датчиків, комп'ютерів та механіки дає можливість керувати літаком в режимі автопілота.

Усі параметри, тобто датчики (які дають вхідні сигнали в комп'ютери), комп'ютери (мізки системи) та механізми (виходи системи, такі як двигуни та заслінки) однаково важливі для створення успішної автоматизованої системи.

Але в цьому підрозділі буде зосереджено на датчиках частини системи та розглянуто різні поняття, пов'язані з датчиками (наприклад, типи, характеристики, класифікація тощо).

Є багато різноманітних визначень, що таке датчик, але зазвичай, можна визначити датчик як вхідний пристрій, який забезпечує вихід (сигнал) щодо певної фізичної величини (вхід). Вигляд основних типів датчиків наведено на рисунку 1.7. [3]

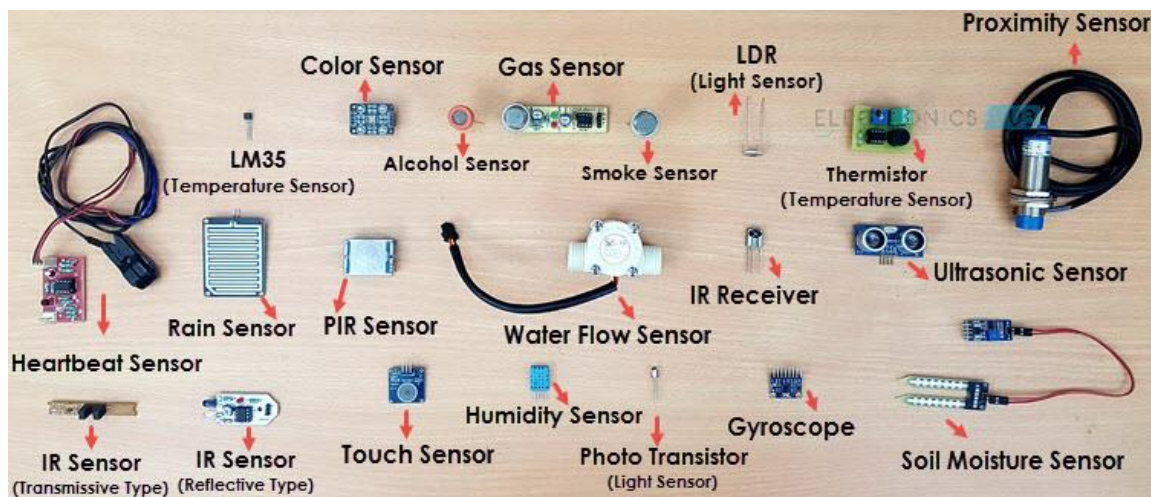


Рисунок 1.7 – Різноманітні датчики

Термін "пристрій введення" у визначенні датчика означає, що він є частиною більшої системи, яка забезпечує введення в основну систему управління (наприклад, на процесор або мікроконтролер).

Ще одне унікальне визначення датчика таке: це пристрій, який перетворює сигнали з однієї енергетичної області в електричну. Визначення датчика можна зрозуміти, якщо буде розглянуто наступний приклад.

Найпростіший приклад датчика - LDR або резистор, залежний від світла. Це прилад, опір якого змінюється залежно від інтенсивності світла, якому він

піддається. Коли світла, що падає на LDR, більше, його опір стає набагато меншим, а коли світла стає все менше, то і опір LDR стає дуже високим.

Можливо підключити цей LDR в дільниці напруги (разом з іншими резисторами) і перевірити падіння напруги на LDR. Ця напруга може бути відкаліброваною до кількості світла, що падає на LDR.

Існує кілька класифікацій датчиків, зроблених різними авторами та експертами. Деякі дуже прості, а деякі дуже складні.

У першій класифікації датчиків вони поділяються на активні та пасивні. Активні датчики - це ті, для яких потрібен зовнішній сигнал збудження або сигнал живлення.

Пасивні датчики, з іншого боку, не потребують зовнішнього сигналу живлення і безпосередньо генерують вихідну характеристику.

Інший тип класифікації заснований на засобах виявлення, використовуваних у датчику. Деякі засоби виявлення - це електричні, біологічні, хімічні, радіоактивні тощо.

Наступна класифікація базується на явищі перетворення, тобто на вході та на виході. Деякі поширені явища перетворення - фотоелектричні, термоелектричні, електрохімічні, електромагнітні, термооптичні тощо.

Остаточна класифікація датчиків - аналогові та цифрові датчики. Аналогові датчики виробляють аналоговий вихід, тобто безперервний вихідний сигнал щодо вимірюваної кількості.

Цифрові датчики, на відміну від аналогових датчиків, працюють з дискретними або цифровими даними. Дані в цифрових датчиках, які використовуються для перетворення та передачі, мають цифровий характер.

### 1.3 Огляд існуючих видів метеостанцій

Метеорологічна станція - станція моніторингу погоди. Вона складається з метеорологічного майданчика, де розміщується більша частина метеорологічного обладнання, і закритого простору, що містить барометр, барограф та інші пристрої для спостереження. Автоматизовані метеостанції встановлюють на в



незаселених районах або в районах сильно віддалених від місць проживання людей. Дані, отримані на метеостанціях, надсилаються зашифрованими до метеорологічних центрів [1].

Розрізняють цифрові й аналогові метеорологічні установки.

На класичних аналогових метеостанціях, яка зображена на рисунку 1.8 зазвичай розміщують прилади призначені для вимірювання[1]:

- температури
- тиску
- вологості повітря
- швидкості вітру
- опадів
- напрямку та сили вітру

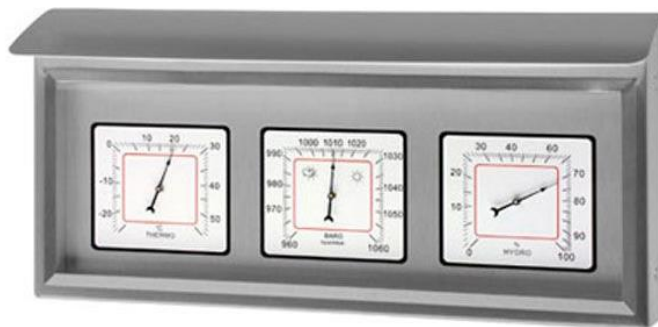


Рисунок 1.8 – Аналогова метеостанція

У вузькому сенсі метеостанцією називають спеціальну установку, яка здійснює метеорологічні спостереження. Найважливішими офіційним метеостанціям світу були призначені пристрої для зняття синоптичних показників. В залежності від кількості отримуваної інформації метеостанції мають конкретну категорію. Метеостанції разом зі стандартними станціями виконують агрометеорологічні спостереження, слугують для визначення розсіяної, сумарної або ж прямої радіації, радіаційний бюджет, ступінь випаровування вологи ґрунту тощо. Найнеобхідніші дані прогнозів погоди надходять із наземних станцій [2].

До цифрових метеостанцій відносять:

— Дорожні метеостанції. Окрім згаданих датчиків, станції вимірювання доріг використовують електронні термометри на поверхні та датчик для вимірювання температури на глибини близько тридцяти сантиметрів (під кришкою), а ще контролер та GPRS модуль для того, щоб передавати дані до спеціалізованих інформаційних центрів. Для надання водіям інформації про погодні умови використовуються інформаційні табло, де відображається температура поверхні та повітря. Крім того на табло можуть з'являтися повідомлення про навколишній стан (мокра дорога, поперечний вітер тощо). [2]

— Лісові метеостанції використовують для уникнення пожеж в лісах. Зазвичай вони отримують живлення від акумуляторів. Установки фіксують кліматичні дані необхідні для аналізу ситуації, такі як вологість дерев, ґрунт та температура навколишнього середовища на різних висотах лісу. Дані обробляються та моделюються карткою протипожежної діяльності, щоб допомогти пожежникам в управлінні можливим запаленням або запобігти водіям від вивітрювання температури поверхні та повітря за допомогою приладових панелей. [2]

— Гідрологічні метеостанції проводять метеорологічні та гідрологічні спостереження за погодою океанів, океанів, річок, озер та заболочених земель. Такі установки розташовані на континентах, плавають на віддалених станціях в морях, крім того є станції спостереження за річками, озерами та водно-болотними угіддями. [2]

— Нещодавно були запущені вітчизняні метеостанції для дому (рисунок 1.9). Засновниками повсякденних установ для вимірювання стану довкілля є звичайні барометри. Функціонал домашньої метеостанції схожий на метеостанцію, вона обробляє лише набагато менше даних, що надходять від одного або декількох датчиків, встановлених поза вікном та в інших приміщеннях. Домашні метеостанції відображають температуру в приміщенні та зовнішню температуру, вимірюють вологість та тиск повітря та формують

прогноз погоди на день на основі даних, отриманих від процесора. Вони працюють як з джерелом живлення, так і зі змінними батареями [3].



Рисунок 1.9 – Побутова домашня метеостанція

Зазвичай для побудови систем метеорологічних досліджень використовують мікроконтролери, персональні комп'ютери або мінікомп'ютери.

#### 1.4 Огляд приладів для визначення різноманітних метеорологічних показників

Для початку розглянемо один із найуживаніших засобів визначення температури навколишнього середовища, а саме термометр.

Термометр - це прилад, який використовується для вимірювання температури або градієнта температури. Цю температуру можна використовувати в промисловості, в метеорології і навіть в медицині.

Цей прилад має дві важливі речі, які допомагають вимірювати температуру. Перший - це датчик, який використовує різні фізичні параметри, такі як розширення різних станів речовини для відчуття зміни, а другий - перетворювач, який перетворює зміну температури в числове число.

Існує дуже велике різноманіття термометрів, враховуючи, якою рідинною вони заповнена. На метеостанціях використовують в основному ртутні термометри.

Термометри, наповнені ртуттю, можуть вимірювати температуру від -35 до 40 градусів Цельсія. Він має масштабний поділ на 0,2 градуса. Але якщо температура нижче -35 градусів Цельсія, то цей термометр не може дати точного результату і стає ненадійним.

Для вимірювання температури нижче -35 градусів Цельсія використовуються наповнені спиртом термометри. Цей тип термометрів може вимірювати від -65 / -75 градусів Цельсія до 20/25 градусів Цельсія.

Термометри використовують різні фізичні параметри, такі як розширення різних станів речовини для вимірювання температури.

Барометр - це прилад, який вимірює атмосферний тиск. Він вимірює тиск в одиницях вимірювання - барах або атмосферах. Існує декілька основних типів барометрів:

- барометри з рідиною;
- барометри без рідини.

Коли барометр показує низький тиск, це означає, що погода дощова, вітряна або хмарна, а коли тиск високий, це означає, що погода швидше за все буде гарною.

Існують різні види барометра, такі як ртутний барометр, анероїдний барометр та цифровий барометр. Серед цих трьох цифровий барометр - останній, який дає найточніші дані.

Анемометр – широко розповсюджений інструмент на вітровій станції. Він використовується для вимірювання швидкості вітру.

Швидкість вітру дає багато інформації про погоду. Наприклад, якщо швидкість вітру висока, це може бути вказівником на початок шторму.

Тож за допомогою анемометра можливо заздалегідь визначити швидкість вітру, яка допомагає захиститися від багатьох стихійних лих.

Висотохмарний проектор – це пристрій, який використовується для вимірювання висоти основи хмари над землею. Проектор зазвичай встановлюється під кутом 90 градусів Цельсія.

Інструмент складається з двох компонентів. Серед яких один - приціл прожектора, а другий - алідада. Алідада - частина кутомірного приладу у вигляді лінійки чи круга, які обертаються всередині лімба навколо однієї з ним осі.

Прожектор має 430 Вт, які розміщені в захищеній коробці, де також є два дзеркала. Ці дзеркала встановлені для отримання променя світла високої інтенсивності, який робить світлову пляму на основі хмари.

Радіозонд- це прилад, який працює від акумулятора і використовується для вимірювання атмосферних параметрів. Параметри включають температуру, тиск, напрямок вітру, а також швидкість.

Він працює за допомогою гумового або латексного балона, заповненого гелієм або водневим газом, який вимірює параметри і передає їх радіоприймачеві. Радіозонди можуть працювати на частоті від 403 МГц до 1680 МГц. У більшості радіозондів є радіолокаційні відбивачі.

Барограф - це різновид барометра, який представляє його результат у графічній формі. Він використовується для вимірювання барометричного тиску, тобто атмосферного тиску. Цей інструмент включає в себе комп'ютер, який дає точний результат тиску у графічній формі, який повідомляє про зміни, що відбуваються в атмосфері.

Також є прилад, який виявляє блискавку, яка утворюється під час грози. Існує три типи детектора блискавки з урахуванням кількості присутніх антен. Це наземна система, система на основі мобільних пристроїв та космічна система. Детектори блискавок не працюють поодиночі, а працюють разом з погодними радарми. Індикатор показує електричну активність, а радіолокатор показує опади, і разом вони дають повний результат.

Електричні датчики вологості називаються гігрометрами. За принципом дії, гігрометри діляться на [5]:

— Ємнісні. Ємнісні гігрометри, в найпростішому випадку, являють собою конденсатори з повітрям в якості діелектрика в зазорі. Відомо, що у повітря діелектрична проникність безпосередньо пов'язана з вологістю, а зміни вологості діелектрика призводять і до змін у ємності повітряного конденсатора.

— Резистивні. Такий датчик включає в себе два електроди, які нанесені на підкладку, а поверх на самі електроди нанесений шар матеріалу, який відрізняється досить малим опором, сильно, проте, що змінюється в залежності від вологості.

— Термісторні. Термісторний гігrometer складається з пари однакових термісторів. Один з включених в схему термісторів розміщують у герметичній камері з сухим повітрям. А інший - у камері з отворами, через які в неї надходить повітря з характерною вологістю, значення якого потрібно виміряти. Термістори з'єднують за мостовою схемою, на одну з діагоналей моста подається напруга, а з іншого діагоналі зчитують покази.

— У випадку, коли напруга на вихідних клеммах дорівнює нулю, температури обох компонентів рівні, отже однакова і вологість. У разі, коли на виході буде отримано не нульове напруга, то це свідчить про наявність різниці вологостей в камерах. Так, за значенням отриманого при вимірюванні напруги визначають вологість.

— Оптичні. Цей вид датчиків найбільш точний. В основі роботи оптичного датчика вологості — явище пов'язаної з поняттям «точка роси». У момент досягнення температурою точки роси, газоподібна і рідка фази - умови термодинамічної рівноваги. Так, якщо взяти скло, і встановить в газоподібному середовищі, де температура в момент дослідження вище точки роси, а потім почати процес охолодження даного скла, то при конкретному значенні температури на поверхні скла почне утворюватися водяний конденсат, це водяний пар стане переходити в рідку фазу. Дана температура якраз і буде точкою роси[6].

— Електронні. Принцип роботи електронного датчика вологості повітря

заснований на зміні концентрації електроліту, що покриває собою будь-електроізоляційний матеріал. Існують такі прилади з автоматичним підігрівом з прив'язкою до точки роси.

Температура точки роси нерозривно пов'язана і залежить від таких параметрів як вологість і тиск в навколишньому середовищі. В результаті, маючи можливість вимірювання тиску і температури точки роси, вийде легко визначити і вологість. Цей принцип є основою для функціонування оптичних датчиків вологості.

Найпростіша схема такого датчика складається з світлодіода, що світить на дзеркальну поверхню. Дзеркало ж відбиває світло, змінюючи напрямок, і направляючи на фотодетектор. В даному випадку дзеркало можна підігрівати або охолоджувати за допомогою спеціального пристрою регулювання температури високої точності. Часто таким пристроєм виступає термоелектричний насос. Звичайно ж, на дзеркало встановлюють датчик для вимірювання температури.

Часто точка роси вимірюється над концентрованим розчином хлориду літію, який є дуже чутливим до мінімальних змін вологості. Для максимальної зручності такий гігрометр часто додатково обладнують термометром. Цей прилад володіє високою точністю і малою похибкою. Він здатний вимірювати вологість незалежно від температури навколишнього середовища[7].

Принцип роботи анемометра полягає у виявленні змін у деяких фізичних властивостях потоку або в впливі потоку на механічний пристрій рідини. Анемометр може вимірювати повну швидкість, швидкість у площині та швидкісну складову в певному напрямку. Крім того, сучасні анемометри можуть вимірювати напрям вітру, об'ємний потік, вологість, температуру і тиск, в залежності від моделі. Це перетворює анемометри на портативні метеостанції [5].

В залежності від способу вимірювання та типу приймального пристрою анемометри поділяють на ряд типів [4]:

— Поворотний (крилові, чашкові). Тиск повітря на внутрішній поверхні чашки перевищує тиск на зовнішній поверхні, внаслідок чого лезо обертається.

Вісь ножа з'єднана з вимірювальним механізмом, який обчислює кількість обертів за певний проміжок часу

— Теплові. Принцип роботи теплового анемометра полягає в вимірюванні температури пластини або нитки, яку вітер дме. Залежно від швидкості вітру потрібна різна енергія, щоб підтримувати температуру постійною. Тобто швидкість вітру можна визначити за температурою плити.

— Ультразвукові (акустичні). Ґрунтується на вимірюванні швидкості звуку між передавачем та приймачем в залежності від швидкості вітру.

— Оптичний (лазерний доплерівський). Працює за принципом частоти відбитого або розсіяного світла рухомим об'єктом (ефект Доплера) на швидкість цього об'єкта.

Датчики, що використовуються для виявлення швидкості руху предмета чи транспортного засобу, називаються датчиком швидкості. Існують різні типи датчиків для виявлення швидкості, такі як датчики швидкості колеса, спідометри, LIDAR, РЛС наземної швидкості, журнали пітометра, доплерівські радары, індикатори швидкості руху повітря, пітотрубки тощо.

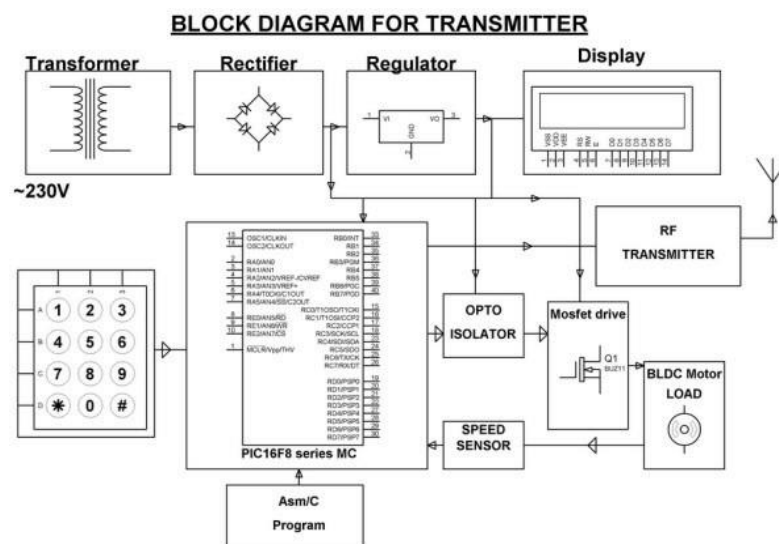


Рисунок 1.10 – Використання датчика руху на базі PIC

Проект на основі мікроконтролерів PIC для синхронізації швидкості декількох двигунів у галузях, що використовують бездротові технології, є



типовим застосуванням датчика швидкості. Один з декількох двигунів у цій галузі розглядається як головний двигун, який виконує роль передавача, а решта двигунів, що діють як приймачі, будуть слідувати швидкості основного двигуна. Основними двигунами та приймачами, що використовуються в таких проектах, є двигуни BLDC, які керуються за допомогою ШІМ-керування в режимі радіочастотного бездротового зв'язку. Приклад організації датчика швидкості зображено на рисунку 1.10.

Опорний об / хв задається кожному валу двигуна, на якому встановлений ПЧ-датчик, і закритий контур отримується подачею цього виходу до контролера в ланцюзі. Повна швидкість відобразатиметься на дисплеї, а необхідну швидкість роботи всіх двигунів можна отримати, ввівши потрібний відсоток за допомогою клавіатури. Цей введений відсоток узгоджується з робочим обертом, підтримуючи відповідну потужність постійного струму до двигуна з автоматичним регулюванням виходу імпульсу ширини мікроконтролера.

Таким чином, змінюючи швидкість передачі двигуна, можливо змінити швидкість роботи всіх двигунів за допомогою цієї технології.

Існують різні типи датчиків, які використовуються для вимірювання температури, такі як датчики температури контактного типу, датчики температури безконтактного типу. Вони знову поділяються на механічні датчики температури, такі як Термометр та Біметал. Електричні датчики температури, такі як термістор, термопара, термометр опору. Приклад використання термометра наведено на рисунку 1.11.

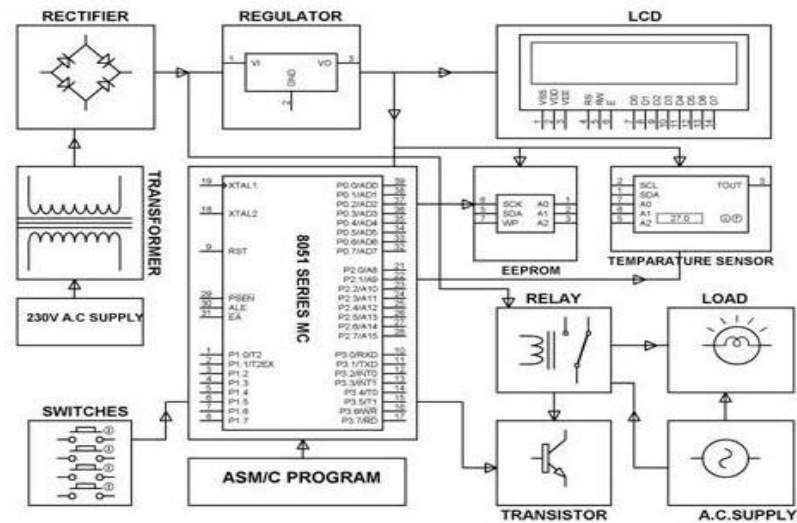


Рисунок 1.11 – Приклад використання датчика температури

Конструкція промислового регулятора температури для регулювання температури пристроїв, що використовуються в промислових областях, є одним з часто використовуваних практичних застосувань датчика температури. У цій схемі IC DS1621 використовується цифровий термометр як датчик температури, термостат, який забезпечує 9-бітові показання температури. В основному схема складається з мікроконтролера 8051, EEPROM, датчика температури, РК-дисплея та інших компонентів.

РК-дисплей використовується для відображення температури в діапазоні від -55 градусів до + 125 градусів. EEPROM використовується для зберігання заздалегідь заданих параметрів температури користувачем через мікроконтролер серії 8051. Реле, контакт якого використовується для навантаження, приводиться в дію мікроконтролером за допомогою драйвера транзистора.

Електронний датчик, що використовується для вимірювання інфрачервоного випромінювання світла з об'єктів у його полі зору, називається датчиком PIR або датчиком піроелектрики. Кожен об'єкт, який має температуру вище абсолютного нуля, випромінює теплову енергію у вигляді випромінювання, випромінюваного на довгих хвилях інфрачервоної хвилі, невидимих для людського ока, але їх можна виявити за допомогою електронних пристроїв спеціального призначення, таких як PIR-детектори руху. Принцип роботи датчика

наведено на рисунку 1.12.

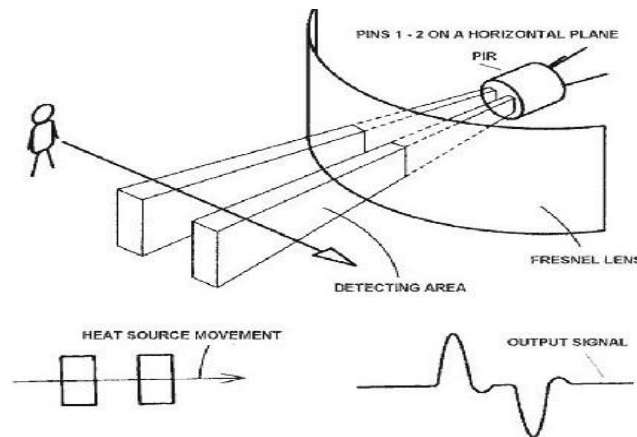


Рисунок 1.12 – Принцип роботи PIR

Сам PIR-датчик розділений на дві половини, чутливі до ІЧ світла, і коли предмет потрапляє в поле зору датчика, тоді позитивна диференціальна зміна буде вироблятися між двома половинами з перехопленням першої половини датчика PIR. Аналогічно, якщо об'єкт покине поле зору, тоді буде вироблятися негативна диференціальна зміна. PIR або пасивний інфрачервоний датчик названий пасивним, оскільки він не випромінює енергії та випромінювання для виявлення випромінювання. Існують різні типи датчиків, які використовуються для виявлення руху, і ці датчики PIR класифікуються на основі кута (широкої області), над яким вони можуть визначати рух об'єктів, таких як кути 110 градусів, 180 градусів і 360 градусів. Принцип використання датчика наведено на рисунку 1.13.

Система автоматичного відкривання дверей - це типове застосування датчиків PIR, призначених для автоматичних операцій закривання та відкривання дверей на основі руху кузова біля дверей. Система автоматичного відкривання дверей на основі датчиків PIR складається, головним чином, з датчика PIR, мікроконтролера 8051, СІД водія, дверного двигуна.

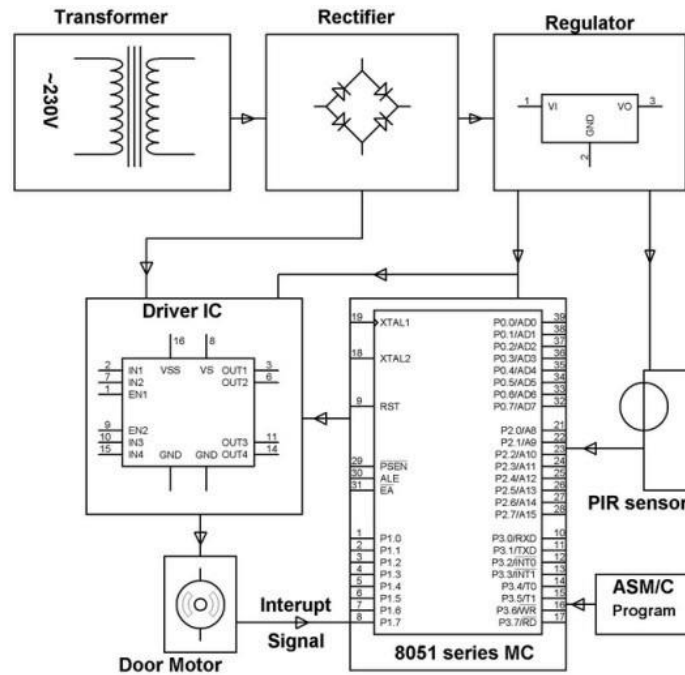


Рисунок 1.13 – Приклад організації пристрою на основі PIR

Якщо поруч з дверима спостерігається рух тіла, то інфрачервоне випромінювання, яке випромінюється з корпусу, призведе до того, що датчик видасть чутливий сигнал, який подається на мікроконтролер. Потім двигун дверей вмикається і керується мікроконтролером через ІЧ драйвер. Таким чином, якщо хтось наблизиться до дверей, мікроконтролером буде надіслана команда для відкривання дверей і встановлена часова затримка для автоматичного закривання дверей.

Принцип дії ультразвукового датчика схожий на сонар або радіолокатор, в якому інтерпретація відлуння радіо- або звукових хвиль для оцінки атрибутів цілі шляхом генерування високочастотних звукових хвиль (близько 40 кГц). Перетворювач, що використовується для перетворення енергії в ультразвукові або звукові хвилі з діапазоном вище діапазону слуху людини, називається ультразвуковим перетворювачем.

Вимірювання відстані на недоступних ділянках є типовим застосуванням ультразвукових датчиків. Схема, яка зображена на рисунку 1.14, складається з ультразвукового модуля, РК-дисплея та мікроконтролера. Ультразвуковий модуль

пов'язаний з мікроконтролером, і цей ультразвуковий перетворювач складається з передавача та приймача.

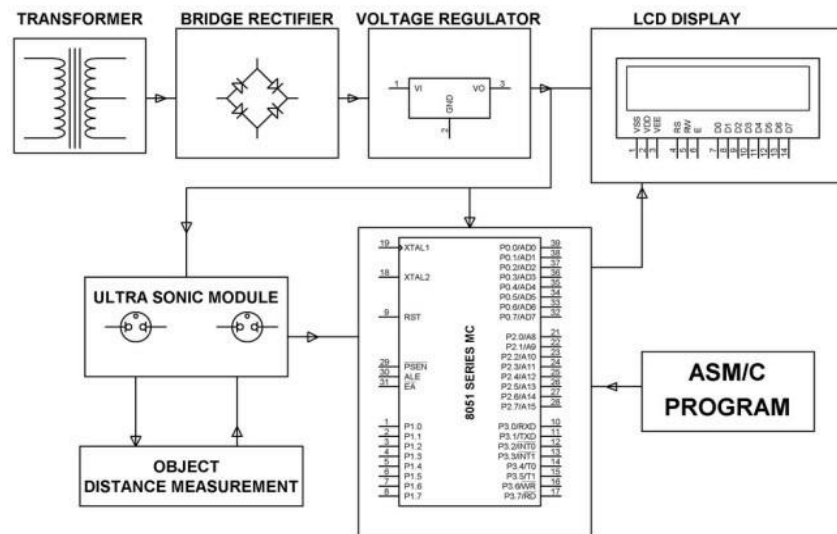


Рисунок 1.14 – Схема використання ультразвукового датчика

Хвилі, що передаються перетворювачем, повертаються знову після того, як хвилі відбиваються від об'єкта. Швидкість звуку вважається для обчислення часу, необхідного для надсилання та прийому хвиль. Відстань обчислюється виконанням програми на мікроконтролері, а потім вона відображається на РК-дисплеї.

### 1.5 Класифікація та огляд видів мінікомп'ютерів

На даний момент є досить великий вибір мінікомп'ютерів. До основних з них, які вже затвердили свою надійність на ринку відносять [8]:

- Raspberry Pi
- Orange Pi
- FriendlyARM
- PINE64
- ODROID
- Banana Pi
- Onion

## — Beagle Bone

Всі вони розрізняються за чіпом керування, розміром пам'яті, операційною системою (Linux, OpenElec & XBMC, RetroPie, RISC OS, Firefox OS, Plan 9, Android, Pipboy, Windows 10 IoT та інші), влаштованими портами, наявністю дистанційного керування (Bluetooth, Wi-Fi, Інфрачервоний порт), вхідною напругою, частотою дискретизації, оперативною пам'яттю та іншими. Від відповідних характеристик залежатиме і його ціна, тому необхідно з розумом ставитись до вибору необхідних. Також всі вони можуть бути різних розмірів, що необхідно враховувати при виготовленні корпусу для приладу [8].

### 1.6 Обґрунтування створення системи метеорологічних досліджень на базі мінікомп'ютерів

На даний момент в основному використовують мікроконтроллери для створення систем метеорологічних досліджень, але кількість прихильників мінікомп'ютерів сильно зростає. Це пов'язано з більшою гнучкістю налаштування, можливістю виведення інформації дистанційно чи навіть у вигляді сайту, запустивши сервер на мінікомп'ютері або надсилання інформації на сервер через мережу.

Також мінікомп'ютери мають значно більшу потужність та пам'ять, що дозволяє зберігати інформацію всередині, компонувати її та надсилати детальну статистику за довгий період часу. Важливим недоліком являється лише його вартість в порівнянні з мікроконтролерами.

Крім того є можливість підключати велику кількість периферії та датчиків, можливе налаштування в графічному режимі. Набагато простіше відбувається і прошивка мінікомп'ютера, для цього необхідно лишень підключити до звичайного комп'ютера чи ноутбука за допомогою USB або Ethernet порта (в залежності від марки).

Також, використовуючи I2C шину можна підключити декілька датчиків послідовно, для вимірювання даних в різних приміщеннях або на різних ділянках. Це допоможе отримувати інформація та обраховувати її для,

приміром включення поливу рослин за низької вологості в приміщенні.

Крім того за необхідності є можливість підключати сенсори за допомогою інших інтерфейсів, незалежно. Необхідно буде лише доопрацювати скрипти обробки даних.

Як бачимо, для трудомісткої роботи значно краще виглядає мінікомп'ютер, а якщо ж потрібно лише оформити простий вихід тільки актуальної інформації, то доцільно використовувати мікроконтроллер.

## 1.7 Висновок

У даному розділі було розглянуто існуючі види систем метеорологічних досліджень, наведено їх класифікацію. Було наведено та описано прилади для визначення метеорологічних показників. Зображено як аналогові, так і електронні види кожного з них.

Також було наведено загальну класифікацію мікрокомп'ютерів, описано їхні основні характеристики та відмінності між різними моделями, залежність ціни від обраних характеристик. Обґрунтовано важливість вибору правильної комплектації та наведено переваги відносно мікроконтролерів. До недоліків належить лише більша вартість яка нівелюється ширшим функціоналом та можливістю прийому даних з великої кількості датчиків.

## 2 ПРОЕКТУВАННЯ СИСТЕМИ МЕТЕОРОЛОГІЧНИХ ДОСЛІДЖЕНЬ

### 2.1 Вибір мінікомп'ютера

Одним з найголовніших етапів розробки даної метеорологічної станції є вибір мінікомп'ютера. Вибір падає на основних лідерів на ринку, таких як Orange Pi, PINE A64, ODROID, LattePanda, Raspberry Pi. По черзі пройдемося по задовольняючим наші умови моделям [9].

Для початку розглянемо Orange Pi Zero зображений на рисунку 2.1. Це досить мініатюрний комп'ютер з однією платою на процесорі Allwinner H2 +, в котрий входять чотири ядра Cortex A7 для обчислень даних, які мають тактову частоту до 1,2 ГГц. та повністю відкритий вихідний код. Він може працювати під управлінням Android 4.4, Ubuntu, Debian. Він має 256 МБ оперативної пам'яті DDR3 типу SDRAM. Також цей міні-комп'ютер оснащений прискорювачем графіки на базі ядра Mali-400MP2, що дозволяє розшифровувати 4К-відео формату H.265 / HEVC з можливою частотою відтворення до 30 кадрів в секунду. Крім того даний пристрій підтримує карти пам'яті типу microSD, з ємністю до 64Гб [9].



Рисунок 2.1 – Orange Pi Zero



Міні-комп'ютер Orange Pi Zero ідеально підійде для побудови на його базі вбудованих систем управління і систем контролю. Наявна підтримка наступних операційних систем: Debian, Armbian, Ubuntu MATE. Вартість даного мінікомп'ютера становить близько 25 долларів. Має досить малий вибір операційних систем та графічний прискорювач, який в нашому випадку не є необхідним однак він досить сильно гріється при навантаженні на сервер на самому мінікомп'ютері [10].

Перейдемо до PINE A64 +, який зображено на рисунку 2.2. Як впливає з назви, комп'ютер працює на потужному 64-бітному чотирьохядерному процесорі 1,2 ГГц ARM Cortex A53. Цей комп'ютер обладнаний одним гігабайтом оперативної пам'яті DDR3. Графічна підсистема працює на основі двоядерного графічного процесора Mali 400 MP2, який дозволяє легко переглядати відео з роздільною здатністю до 4К.

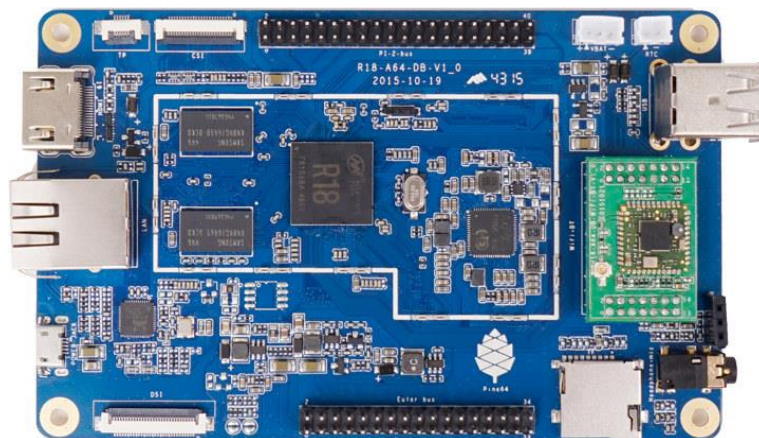


Рисунок 2.2 – PINE A64+

На платі є два роз'єми GPIO - сумісний Raspberry Pi2, другий формат шини Euler. Наявність такої кількості цифрових портів робить Pine64 гідним конкурентом Raspberry за вбудовані системи управління, які потребують великої кількості GPIO I / Os. Є підключення до акумуляторів (є перемички, які відключають живлення комп'ютера як від зовнішніх, так і від 5В до 3,6 В). До карти можуть бути підключені цифровий дисплей шини, відеокамера та тачпад. На жаль це все відображається і на ціні, яка знаходиться в діапазоні 50 долларів [9].

Плата дуже швидка. Необхідно переконатися що в системі виставлена максимальна частота процесора і обрана продуктивна схема управління частотою. При харчуванні від батарей слід навпаки, вибрати збалансований режим або режим економного споживання [9].

ODROID-C2, зображений на рисунку 2.3 - це новітня 64-розрядна платформа на 4-ядерному процесорі Amlogic ARM® Cortex®-A53 (ARMv8), що працює на частоті 2ГГц. Платформа призначена для побудови високопродуктивних систем автоматизації та обробки даних, домашніх кінотеатрів, прототипування вбудованих пристроїв [9].



Рисунок 2.3 – ODROID-C2

Платформа ODROID-C2 підтримує такі операційні системи як Ubuntu, Android, ARCHLinux, Debian та інші портовані 64-розрядні операційні системи з відкритим вихідним кодом. У комп'ютері поєднуються мале споживання, розміри і висока продуктивність з можливістю підключення зовнішніх датчиків, пристроїв і виконавчих механізмів. За багатьма параметрами ODROID-C2 більш ніж в 2 рази перевершує такий популярний міні-комп'ютер як Raspberry Pi 2 B. Але і ціна в 90 доларів змушує замислитись над необхідністю вибору даного мінікомп'ютера [10].

Новинка від розробників Raspberry - Raspberry Pi 3 Model B. Тепер цей міні-комп'ютер отримав потужний просунутий процесор і бездротові пристрої

комунікації. Нова версія Raspberry Pi працює набагато швидше попередників з такими додатками як електронні таблиці, обробка текстів, відтворення відео високої чіткості і ігри. Raspberri PI 3 може працювати під управлінням декількох різновидів Linux і навіть безкоштовної Windows 10 [9].

У даній моделі використовується потужний процесор Broadcom BCM2837 з ядром ARM Cortex-A53 та шестидесятичотирьохбітний процесор Quad-Core SoC. Даний процесор для обробки графіки повністю забезпечує підтримку Open GL ES 2.0, а також апаратне прискорення OpenVG і 1080p30 H.264 досить високого рівня розшифровки і може обробити до 1Gpixel / с, 1.5Gtexel / с або 24 гігафлопса інших обчислювальних ресурсів. Що це означає? Це означає, що якщо ви підключите Raspberry Pi 3 до вашого HDTV телевізору, отримаєте можливість переглядати відео з якістю BluRay, використовуючи міжнародний стандарт конверсії H.264 на 40МБіт / с [10].

Найбільше зміна, яка була зроблена в Raspberry Pi 3, є оновленням основного процесора наступного покоління і поліпшені можливості комунікації з Bluetooth Low Energy (BLE) і BCM43143 WiFi контролером на борту. Крім того, в Raspberry Pi 3 розробники поліпшили управління живленням і підвищили потужність USB до 6Вт (1,2 ампер) для підтримки більш потужних зовнішніх USB пристроїв [11].



Рисунок 2.4 – Raspberry Pi 3 Model B

Чотири вбудованих USB порту Raspberry Pi 3 забезпечують широкі можливості для підключення миші, клавіатури, або будь-який інший периферії, але якщо ви захочете додати ще більше USB пристроїв, то потрібно буде використовувати USB концентратор. Майте на увазі, що рекомендується використовувати хаб, щоб не перевантажувати стабілізатор напруги на платі. Включення Raspberry Pi 3 легке і просте - досить підключити будь-який USB джерело живлення в порт мікро-USB. На платі немає кнопки живлення, так що Pi почне завантажуватися, як тільки буде підключений до джерела живлення. Щоб вимкнути його просто вимкніть живлення. Чотири вбудованих USB-порту можуть виводити до 1.2А, дозволяючи підключати більше ненажерливі пристрої USB (Це вимагає 2,5А мікро-USB джерело живлення) [10].

Вагомим додатком до всього виступлять периферійні пристрої низького рівня, що робить Raspberry Pi досить вдалим інструментом для того щоб збирати і обробляти інформацію, а також для управління пристроями, які під'єднані до системи. Сорока контактний роз'єм з інтерфейсом введення та виведення GPIO на пристрої надає можливість використовувати до 27 підключень через різноманітні інтерфейси, як I2C, SPI, UART, GPIO, а також підключати як 3,3 так і 5V джерела живлення. Кожний з контактів GPIO є ідентичним до свого попередника Model B +. Крім того вартість даної моделі складає всього 40 доларів [10].

Враховуючи всі наведені вище характеристики різноманітних моделей та марок мінікомп'ютерів вибір зупиняється на Raspberry Pi 3 Model B в зв'язку з його потужним процесором, доступністю та порівняно з іншими низькою ціною.

## 2.2 Розробка структурно-функціональної схеми системи метеорологічних досліджень

Структурною схемою чи блок-схемою називають схему, що визначає найосновніші функціональні частини пристрою, показує їх взаємозв'язок та надає пояснення щодо їх призначення. Функціональна частина розуміється як складова частина схеми: елемент, пристрій, функціональна група, функціональна одиниця [11].

Структурна схема призначена для зображення основної структури приладу, а саме його основних деталей, блоків, вузлів взаємодії та показує з'єднання між ними. Структурна схема повинна пояснювати, для чого потрібен даний виріб, суть його роботи в основних режимах, і за допомогою чого всі його частини працюють разом. Позначення структурних елементів може бути довільним, хоча для виконання схем слід дотримуватися загальноприйнятих правил [11].

Для розробки необхідного пристрою знадобиться під'єднати до мінікомп'ютера датчики тиску, температури та вологості для отримання даних і дисплей для виведення інформації в реальному часі. Структурно-функціональна схема пристрою зображена на рисунку 2.5.

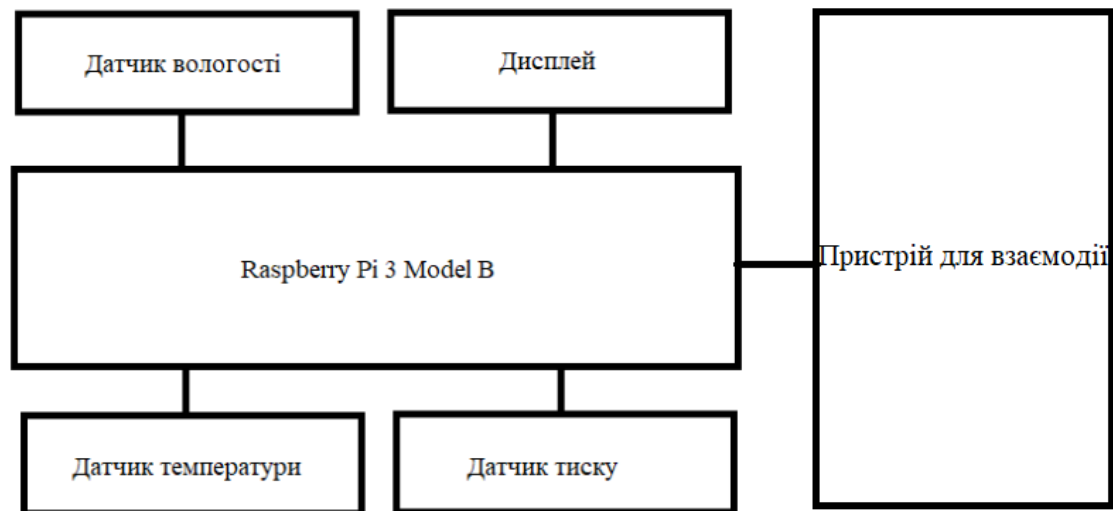


Рисунок 2.5 – Структурно-функціональна схема пристрою

Також можна використати не три різних датчика а один, який буде містити в собі функції всіх трьох.

### 2.3 Вибір електронних компонентів

Електронні датчики (вимірювачі) – важлива складова в автоматизації будь-яких технологічних процесів і в управлінні різними машинами і механізмами. За допомогою електронних пристроїв можна отримати повну інформацію про параметри контрольованого обладнання [11].

Принцип роботи будь-якого електронного датчика побудований на перетворенні контрольованих показників в сигнал, який передається для подальшої обробки керуючим пристроєм. Можливо вимір будь-яких величин – температури, тиску, електричної напруги і сили струму, сили світла та інших показників [11].

Для початку оцінимо ринок датчиків в цілому.

DS18B20 - водонепроникний цифровий датчик температури. Він є простим у використанні, якщо потрібно щось виміряти на великих відстанях або при намоканні. Це дозволяє використовувати велику кількість датчиків на одній шині. Точність:  $\pm 0,5^\circ \text{C}$  від  $-10^\circ \text{C}$  до  $+85^\circ \text{C}$ . Біт: з 9 до 12 біт, передача даних через серійний 1-провідний інтерфейс, 64-бітний унікальний і фіксований серійний номер, багатоточне зчитування, фантомна потужність Робоча напруга: 3,0 В - 5,5 В, діапазон робочих температур: від  $-55$  до  $+125^\circ \text{C}$ , пороги температури ( $^\circ \text{C}$ ): 2 [12].

Датчик близькості ІЧ-термометра пірометр MLX90614 дозволяє вимірювати температуру від  $-40$  по  $+380^\circ \text{C}$ .

Датчик відрізняється високою метрологічною точністю і невеликим зміщенням часу. Датчик є частиною системи моніторингу, вимірювання та контролю температури та вологості [12].

Датчик вологості та температури Aries PVT10 для безперервного перетворення відносної вологості та неагресивної температури газу у два рівномірні вихідні сигнали 4 ... 20 мА та для передачі вимірюваних значень через інтерфейс RS-485 (Modbus RTU) [13].

IS42R1.01 використовується для вимірювання внутрішньої та зовнішньої температури корпусів. Елемент вимірювання датчика знаходиться в корпусі. Щоб уникнути негативного впливу нагріву ланцюга на результат вимірювання, датчик ізолюється від електронної частини ланцюга [13].

Система контролю температури та вологості температури застосовується :

— на автомобілях-рефрижераторах, холодильних установках;

- на зерновозах, комбайнах, зернохранилищах;
- у приміщеннях, що потребують жорсткого контролю параметрів мікроклімату;

Як приклад використання можна вибрати музеї, художні галереї, церковні будівлі, документальні магазини, бібліотеки, винні льохи та багато інших місць.

Завдяки чудовим налаштуванням датчика є можливість отримати сертифікат на вимірювання температури та вологості. Вимірювання температури у сотнях градусів є унікальним параметром серед аналогів, а тривала стабільність вимірювань забезпечує тривалий термін служби модуля.

Під час численних випробувань датчик фіксував зміни температури та вологості в приміщенні при вході та виході з приміщення.

Для порівняння, фактична похибка вимірювання температури з простими вбудованими датчиками DS1820 зазвичай становить лише  $\pm 5$  ° C, оскільки електронні компоненти поблизу нагріваються і довговічна стабільність таких датчиків не нормалізується. IS42R1.01 виграє від своєї високої точності та стабільності.

Барометр, розроблений Bosch, базується на мікросхемі BMP180 із вбудованим АЦП, за допомогою якого можна визначити температуру навколишнього середовища та тиск. Він заснований на п'єзорезистивної технології ЕМС [13].

Використовується в проектах для визначення швидкості підйому / спуску, в системах автопілотування, на метеостанціях, в GPS - навігації тощо.

За допомогою барометра BMP180 можна виміряти тиск від 300 до 1100 гектопаскалів, що еквівалентно +9000 до -500 м відносно рівня моря. Він призначений для прямого підключення до мікроконтролера через шину I2C [13].

BME280 працює як BMP180 в інтерфейсі I2C. Шина SPI підтримується. Новий датчик, показаний на малюнку 2.6, має лише 4 входи порівняно з попередньою версією. В цілому логіка датчика залишилася колишньою, але зазнала деяких довгоочікуваних удосконалень (у прямому розумінні цього слова).

Як і раніше, коефіцієнти повинні спочатку зчитуватися датчиком (вони є унікальними для кожного екземпляра). Потім прочитайте АЦП температури та тиску та обчисліть фактичні значення температури та тиску, використовуючи формули в документації на датчики. BME280 значно покращив стабільність температури та збільшив роздільну здатність АЦП для температури та тиску до 20 біт [13]..



Рисунок 2.6 – Схема підключення BME280

BME280 має три робочих режими [14]:

- SLEEP - режим сну,
- режим FORCED, аналогічний BMP085 та BMP180. Тобто вони дали команду вимірювання, чекали, читали дисплеї, датчик перейшов у сплячий режим.

- NORMAL - Цей режим є найцікавішим і приводить датчик в циклічну роботу (постійний повтор). Тобто він автоматично вимикає сплячку та проводить вимірювання через певний проміжок часу, збирає дані в регістри пам'яті та перезавантажується у визначений час.

Не доведеться чекати вимірювання, щоб прочитати дані датчика (використовується режим NORMAL). Крім того, датчик має фільтр, який також можна адаптувати під різноманітні потреби. Фільтр працює з попереднім значенням. Його робота описується формулою 3.1.



$$data\_filtered = \frac{data\_filtered\_old \cdot (filter\_coefficient - 1) + data\_ADC}{filter\_coefficient}, \quad (3.1)$$

Вологовимірювач в ВМЕ280 вимкнено. Тільки після внесення відповідних налаштувань можна зчитувати дані гігрометра. Вологість обчислюється за формулою, використовуючи поправочні коефіцієнти. Діапазон вимірювання вологості від 0 до 100%. Більшість дешевих гігрометрів здатні вимірювати вологість від 20 до 80%.

Як видно, всі датчики є конкурентоспроможними, але лише до тих пір, поки не постає питання в порівнянні їх ціни. Провівши огляд всіх вибраних варіантів досить гарним вибором може являтися датчик ВМЕ280, враховуючи його можливості та ціну в чотири долара.

Всі дисплеї поділяються на [15]:

- РК-дисплеї були винайдені багато років тому. РК-панелі використовують рідкокристалічне світло, яке також підсвічується окремою системою невеликих ламп. РК-екрани встановлені в комп'ютерних моніторах, телевізорах, камерах та багатьох інших пристроях. РК-екрани IPS споживають мало енергії і добре поведуться під сонцем. Перша і, можливо, найважливіша відмінність від OLED, яка одразу кидається в очі у порівнянні - набагато менший контраст. Як результат, чорний колір на РК-екрані світліший і блідіший, ніж на екрані OLED.-екрані.

- На дисплеях органічного світлодіода (OLED) використовуються спеціальні світлодіоди, які виробляють набагато більше світла і не потребують окремої системи підсвічування. В результаті темні ділянки екрану стають значно вираженішими та глибшими, а ті, що світяться порівняно з ними, здаються більш насиченими та яскравими. Оскільки немає підсвічування, OLED є тоншими, ніж РК - вони не мають цілого шару, відповідального за освітлення пікселів.

Опираючись на наведені вище приклади вирішуємо вибрати датчик атмосферного тиску, температури і вологості BME280 та LCD дисплей WH 1602. Отримуємо більш явну картину майбутнього пристрою.

#### 2.4 Огляд та вибір інтерфейсів підключення

Потужною особливістю Raspberry Pi є ряд штифтів GPIO (загального призначення введення / виводу) уздовж верхнього краю плати. 40-контактний заголовок GPIO знаходиться на всіх поточних платах Raspberry Pi (незаселених на Pi Zero та Pi Zero W). До Pi 1 Model B + (2014) плати мали всього 26-контактів.

Будь-який із штифтів GPIO може бути позначений (у програмному забезпеченні) як вхідний або вихідний штифт і використовуватися для широкого кола цілей.

На платі присутні два штирі 5В та два штирі 3В3, а також ряд заземлень (0В), які не можна налаштувати. Інші штифти - це загальноприйняті 3В3 штирі, значення виходів встановлено на 3В3, а на входах - 3В3.

Штифт GPIO, позначений як вхідний контакт, може читатися як високий (3В3), так і низький (0В). Це полегшується із застосуванням внутрішніх підтягуючих чи знижувальних резисторів. Штифти GPIO2 та GPIO3 мають фіксовані резистори, що підтягуються, але для інших штифтів це можна налаштувати в програмному забезпеченні.

Окрім простих пристроїв введення та виведення, GPIO-штифти можна використовувати з різними альтернативними функціями, деякі доступні на всіх штирях, а інші на певних штифтах.

ШИМ (широтно-імпульсна модуляція)

Програмне забезпечення ШИМ доступне на всіх штирях

Обладнання ШИМ доступне на GPIO12, GPIO13, GPIO18, GPIO19, SPI.

Зручну довідку можна отримати на Raspberry Pi, відкривши вікно терміналу та запустивши команду `pinout`. Цей інструмент надається бібліотекою GPIO Zero Python, яку він встановлюється за замовчуванням на графічному зображенні Raspbian, але не на Raspbian Lite. Отриманий результат зображено на рисунку 2.7.

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ pinout
-----
 00000000000000000000 J8
 10000000000000000000
-----
Pi Model 3B V1.2
-----
  [D] [SoC]
  [S]
  [I]
-----
 pwr HDMI I A Net
-----

Revision      : a02082
SoC           : BCM2837
RAM           : 1024Mb
Storage       : MicroSD
USB ports     : 4 (excluding power)
Ethernet ports : 1
Wi-fi        : True
Bluetooth    : True
Camera ports (CSI) : 1
Display ports (DSI) : 1

J8:
 3V3 (1) (2) 5V
GPI02 (3) (4) 5V
GPI03 (5) (6) GND
GPI04 (7) (8) GPI014
GND (9) (10) GPI015
GPI017 (11) (12) GPI018
GPI027 (13) (14) GND
GPI022 (15) (16) GPI023
 3V3 (17) (18) GPI024
GPI010 (19) (20) GND
GPI09 (21) (22) GPI025
GPI011 (23) (24) GPI08
GND (25) (26) GPI07
GPI00 (27) (28) GPI01
GPI05 (29) (30) GND
GPI06 (31) (32) GPI012
GPI013 (33) (34) GND
GPI019 (35) (36) GPI016
GPI026 (37) (38) GPI020
GND (39) (40) GPI021

For further information, please refer to https://pinout.xyz/
pi@raspberrypi:~ $

```

Рисунок 2.7 – Вивід обробки команди pinout

Міжінтегральна схема (I2C) (NPX) - двопровідний протокол, який може працювати з різною швидкістю (стандартний режим, швидкий режим та швидкісний режим). Хоча інші протоколи більш прості і не мають обмеження швидкості (наприклад, SPI), масштабованість, запропонована I2C™, зробила його привабливим як інтерфейс для деяких CDC-продуктів. I2C запобігає пошкодженню даних, виконуючи провідну операцію AND, оскільки інтерфейс I2C має вихід з відкритим зливом або відкритим колектором. Коли лінія SDA низька, це перекладається в шині I2C, як зайнято.

Також цей інтерфейс дозволяє майстру перевірити стан шини. Підтягуючи головну лінію SDA високо, вона може перевірити стан шини: зайнята, якщо лінія залишається низькою (оскільки деякий пристрій тягне лінію низькою), або вільна, якщо лінія висока.

Протокол I2C формується за допомогою SDA (двонаправлена лінія: отримує шаблон I2C і передає підтвердження, коли спілкування було успішним) та SCL (одностороння лінія). Оскільки протокол I2C дозволяє декілька ведених пристроїв, підключених до однієї шини I2C, кожен з них ідентифікується з іншою адресою. Для адреси веденого пристрою використовується сім біт.

Передача ініціюється стартовою умовою (передній край ПДД, коли високий рівень SCL), а передачі закінчуються, коли виникає умова зупинки (піднімаючий край на ПДД, поки високий рівень SCL).

Інструкція надсилається через SDA в блоки одного байта, де перший біт є найбільш значущим бітом (MSB). Біти SDA повинні бути стабільними, поки SCL високий, і вони повинні змінюватися, коли лінія SCL низька. Якщо біти SDA не є стабільними, коли SCL високий, це розуміється як стан зупинки або запуску, типова причина для пошкодженого повідомлення або відсутнього підтвердження.

На рисунку 2.8 зображено послідовність запрограмованої інформації, що передається по шині

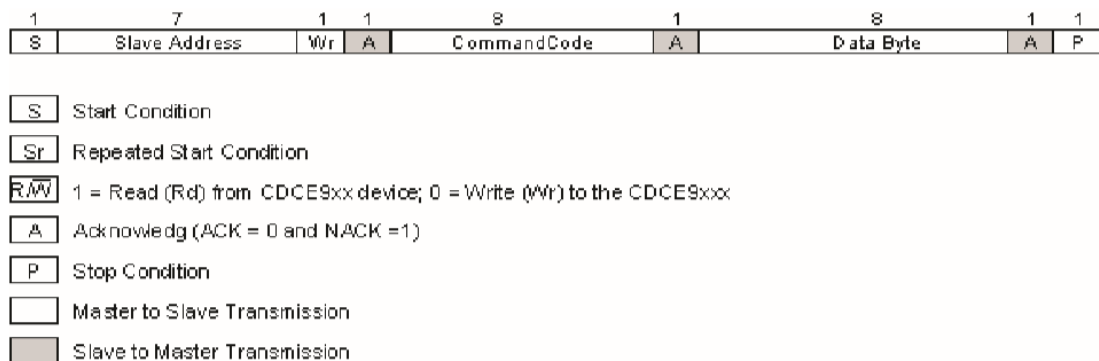


Рисунок 2.8 – Дані, що передаються по I2C

Кількість переданих байтів необмежена; Кожен байт повинен супроводжуватися бітом підтвердження форми підлеглого, що підтверджує успішний прийом байту. За його відсутності рівень залишається високим під час дев'ятого пульсу SCL.

Навіть при використанні процесора, який генерує двосприможну послідовність інтерфейсу, час перевірки все одно повинен бути перевірений. Можливо, що затримка між SCL та SDA не відповідає або занадто обмежена визначеним термінам. У цих випадках в середині надісланого шаблону I2C можуть виникати небажані умови START або STOP, внаслідок чого підлеглий пристрій отримує інше повідомлення або призводить до відсутності підтвердження.

Невдале виконання відбувається, якщо відсутні або несподівані імпульси SCL знаходяться в протоколі. Якщо трапиться щось несподіване, дизайном заповнюється, що стан START відкидає попереднє неповне або помилкове повідомлення, і пристрій готовий приймати нове повідомлення.

Коли відправляється менше восьми біт, підлеглий чекає решти даних. Якщо не буде надісланий восьмий біт, підлеглий не генерує підтвердження. Спосіб перезапуску передачі - це запуск нового повідомлення з умовою запуску для надсилання нового шаблону I2C.

Коли виконується операція з блоком, кількість байтів, що надсилаються або отримуються, може вже фіксуватися в іншому регістрі. Якщо кількість відправлених або отриманих байтів менша за очікувану кількість, машина чекає решти повідомлення. Умова запуску може бути зразком, щоб пропустити стан очікування. BSCOUNT, розташований в адресі регістра 08xh біт 7: 1, встановлює кількість байтів, які читаються під час виконання зчитування блоку. Під час виконання блоку зчитування може виникнути повідомлення про відсутність підтвердження або неправильно зрозуміле повідомлення, якщо BSCOUNT встановлено на більшу величину, ніж байти, доступні зі зміщення, встановленого в коді команди.

Кожен підлеглий пристрій I2C, підключений до шини I2C, має власну підлеглу адресу і відповідає лише на цю адресу, коли протокол I2C розбирається на частини. Коли ведений пристрій ідентифікує себе адресою, що транслюється в

шині I2C, він прив'язує сигнал SDA до землі. З цієї причини SDA є двонаправленим, щоб він міг обробляти підтвердження, отримане від веденого .

Важливо пам'ятати про попередження при програмуванні через I2C CDCE (L) 949, CDCE (L) 937, CDCE (L) 925 і CDCE (L) 913.

Під час інструкції запису (блоку чи байту) кожна зміна байтів є ефективною, коли останній біт байту даних заданий. Після зміни бітів 1: 0 з адреси регістру 01xh змінюється адреса ведених; тому наступне повідомлення I2C починається з нової адреси підлеглого. Використання старої адреси підлеглого викликає відсутнє підтвердження через отриману помилкову адресу підлеглого.

Якщо під час запису блоку значення біт1: 0 з адреси регістру 01xh змінюється, це переводиться на інший ведений адресу, але це не впливає на зв'язок, оскільки ведений адресу перевіряється лише в першому байті повідомлення запису блоку.

Конфлікт виникає, коли до однієї шини I2C підключено більше одного пристрою з однаковою веденою адресою.

CDCE (L) 949, CDCE (L) 937, CDCE (L) 925 та CDCE (L) 913 належать до однієї групи продуктів з однаковою продуктивністю, але мають різний вихід. Їх адреса залежного пристрою має одну частину фіксовану / провідну та одну частину програмовану. CDCE (L) 949 та CDCE (L) 937 мають однакову мережу. Така ж ситуація стосується CDCE (L) 925 та CDCE (L) 913. З цієї причини, якщо програмована частина адреси ведучих змінюється в межах кожної із визначених груп, можна перепрограмувати їх на однакові адреси ведених. Тобто CDCE (L) 949 може бути запрограмований таким же чином, що і CDCE (L) 937. Але в цій ситуації при підключенні до тієї ж шини I2C виникає конфлікт.

CDCE (L) 949, CDCE (L) 937, CDCE (L) 925 і CDCE (L) 913 можуть перепрограмувати останні два біти своєї адреси підлеглого. Тому до шини I2C можна підключити до чотирьох пристроїв кожного.

На кожному з цих пристроїв налаштування за замовчуванням встановлені з різною адресою підлеглого. Тому, за допомогою налаштувань за замовчуванням,

одна з кожної з них повинна мати змогу ділитися тією ж шиною I2C без конфлікту.

SPI часто використовується, коли доступно небагато ліній вводу / виводу, але зв'язок між двома або більше пристроями повинен бути швидким та легким у здійсненні.

SPI розшифровується як Serial Peripheral Interface. SPI - це синхронний протокол, який дозволяє ведучому пристрою ініціювати зв'язок з веденим пристроєм. Дані обмінюються між цими пристроями. SPI реалізований у PICmicro MCU апаратним модулем під назвою Synchronous Serial Port або Master Synchronous Serial Port. Цей модуль вбудований у безліч різних пристроїв PICmicro. Це дозволяє послідовно спілкуватися між двома або більше пристроями з високою швидкістю і досить легко виконувати вплив.

SPI - це синхронний протокол. Тактовий сигнал подається ведучим для забезпечення синхронізації. Тактовий сигнал керує, коли дані можуть змінюватися і коли вони дійсні для зчитування. Оскільки SPI є синхронним, він має тактовий імпульс разом з даними. RS-232 та інші асинхронні протоколи не використовують тактовий імпульс, однак дані мають бути чітко визначені. Оскільки SPI має тактовий сигнал, годинник може змінюватися, не порушуючи даних. Швидкість передачі даних буде просто змінюватися разом зі змінами годинної стрілки. Це робить SPI ідеальним, коли мікроконтролер точно працює так, як RC-генератор.

Тільки головний пристрій може керувати лінією годинника, SCK. Не буде передано дані, якщо годинником не керувати. Усі залежні пристрої контролюються годинником, яким керує головний пристрій. Регістри конфігурації SSP контролюватимуть, як пристрій реагуватиме на вхід годинника.

SPI - протокол обміну даними. Коли дані закриваються, вводяться і нові дані. Коли "передає" дані, вхідні дані повинні бути прочитані перед тим, як спробувати передати знову. Якщо вхідні дані не зчитуються, то дані втрачаються, і в результаті модуль SPI може бути відключений. Завжди читайте дані після

передачі, навіть якщо дані не використовуються у вашій програмі. Дані завжди "обмінюються" між пристроями. Жоден пристрій не може бути просто "передавачем" або просто "приймачем" в SPI. Однак у кожного пристрою є дві лінії даних, одна для введення та одна для виведення. Ці обміни даними керуються тактовою лінією SCK, яка контролюється головним пристроєм.

Часто сигнал вибору підлеглого пристрою вибирається при доступі до пристрою. Цей сигнал слід використовувати, коли в системі існує більше одного підлеглого, але може бути необов'язковим, коли в ланцюзі існує лише один підлеглий. Як правило, його слід використовувати. Цей сигнал відомий як сигнал SS і означає "Slave Select". Він вказує веденому, що ведучий бажає запустити обмін даними SPI між цим веденим пристроєм і самим собою. Сигнал найчастіше активний низький, тому низький на цій лінії вказуватиме на те, що SPI є активним, тоді як високий буде сигналізувати про бездіяльність. Він часто використовується для підвищення шумостійкості системи. Його функція полягає в скиданні підрядного SPI, щоб він був готовий отримати наступний байт.

У SPI дані, як правило, змінюються під час висхідного або спадаючого краю SCK. Ось так дані синхронізуються з тактовим сигналом. За логікою, точка, в якій читаються дані, є протилежною, коли вона змінюється. Дані дійсні в момент читання.

Універсальний асинхронний приймач / передавач (UART) - це стандартний протокол послідовного зв'язку для обміну даними між двома пристроями. У цьому протоколі зв'язку дані передаються послідовно, по одному біту. Ця реалізація використовує кадр, що складається з 8 бітів даних, одного стартового біта, одного необов'язкового біта парності та одного або декількох стоп-бітів. Біт парності - це біт зі значенням 0 або 1, який додається до блоку даних для цілей виявлення помилок. Цей біт необов'язковий; він може або не може бути доданий до корисного навантаження даних. Його також можна встановити як непарне, так і парне. Ці біти часто використовуються при передачі даних для того, щоб дані не були пошкоджені в процесі передачі. Якщо протокол передачі даних



встановлений у непарному паритеті, кожен пакет даних повинен мати непарний паритет. Якщо він встановлений на рівне, кожен пакет повинен мати парний парність. Якщо пакет отриманий з неправильним паритетом, буде створена помилка, і дані потрібно буде повторно передавати. Біт парності для кожного пакету даних обчислюється до передачі даних.

Оскільки протокол UART є асинхронним, йому не потрібен тактовий сигнал. У зв'язку UART швидкість визначається швидкістю передачі. Швидкість передачі даних дорівнює кількості бітів, переданих за секунду, включаючи біти початку та зупинки. Для надійної передачі та прийому даних без втрати бітів і передавач, і приймач повинні мати однакову швидкість передачі даних. Невідповідність швидкості передачі зазвичай призводить до помилки об'ємлення. Загальна швидкість передачі даних становить 4800, 9600, 19200, 38400, 57600 та 115200, але можуть застосовуватися й інші частоти.

Модуль користувача UART - це 8-розрядний універсальний асинхронний передавач, який підтримує двосторонній сумісний RS-232 послідовний зв'язок у форматі даних по двох провідних пристроях. Формат отриманих та переданих даних включає в себе початковий біт, необов'язковий паритет і біт зупинки. Підтримується програмоване синхронізація та вибирається операція стилю переривання або опитування. Підпрограми програмного забезпечення інтерфейсу прикладного програмування (API) надаються для ініціалізації, налаштування та роботи UART. Також надається додатковий API високого рівня для підтримки фоновому прийому команд та друку рядків.

Модуль користувача UART реалізує послідовний передавач та приймач. UART перетворюється на два блоки цифрових комунікацій PSoC, позначені TX та RX, в редакторі пристроїв PSoC Designer™. Блок TX PSoC надає функціональність передавача, а блок RX PSoC надає функціональність приймача. RX і TX працюють незалежно. У кожного є власний реєстр управління та стану, програмовані переривання, введення / виведення, реєстр буфера та реєстр зсуву. Вони мають однаковий формат включення, годинника та даних.

Встановлення біта Enable в регістрах управління RX і TX Control дозволяє UART виконувати роботу. Увімкнення та відключення виконується за допомогою наданих API функцій. Годинник модуля UART користувача поділяється на компоненти RX і TX. Вибрана тактова частота повинна бути у вісім разів більше частоти потрібної швидкості передачі бітів. Кожен отриманий або переданий біт даних вимагає восьми циклів вхідних годин. Годинник конфігурується за допомогою редактора пристроїв PSoC Designer. Отримані та передані дані - це бітовий потік, який складається з початкового біта, восьми бітів даних, необов'язкового біта парності та біта зупинки. Паритет може бути встановлений на жодне, парне або непарне, і встановлюється за допомогою редактора пристроїв PSoC Designer або за допомогою API UART. І RX, і TX встановлені в одній і тій же конфігурації паритету.

Максимальне відхилення, дозволене в тактовому джерелі, становить  $\pm 4\%$  для надійного зв'язку UART. IMO PSoC 1 має максимальну точність 2,5%, а тому може бути використаний. Але 6MHz SLIMO-годинник не можна використовувати, оскільки він має допуск  $\pm 4,2\%$ , що не є прийнятним для надійного зв'язку UART. Деякі пристрої PSoC 1, такі як CY24x94 та CY21x34, мають IMO з допуском  $\pm 4\%$ . При підключенні до шини USB IMO синхронізується з USB-годинником і, отже, стане таким же точним, як тактова шина USB. Якщо він не підключений до шини USB, він працює при  $\pm 4\%$  допуску. На цих пристроях, коли розміщується модуль користувача UART, програма перевірки правил дизайну видає попередження. Наприклад, коли розміщується UART UM в CY8C24x94, сформоване попередження матиме вигляд "UART не слід використовувати на пристроях CY8C24x94 без підключення до шини USB". Системна помилка - сума помилки на обох кінцях каналу зв'язку. Для правильної роботи зв'язку UART повинно бути менше 4%.

API високого рівня додають додаткові вбудовані програми на додаток до основних функцій, щоб надати функціональність команд і рівнів. Редактор пристроїв дозволяє встановити розмір буфера команд прийому, термінатора

команд, роздільника параметрів і нижче того, яке значення одержувач повинен ігнорувати символи. Не всі функції API високого рівня вимагають включення цього буфера. Функції передачі приймають покажчики на рядки, так що цілий рядок може бути надрукований одним викликом функції замість додаткових рядків коду C або ASM.

Функції передачі не вимагають включення переривання передачі UART. Щоб використовувати функції приймача високого рівня, які потребують буфера прийому, перейдіть до вікна "Редактор пристроїв", виберіть UART та виберіть параметр "Увімкнути" для параметра "RxCmdBuffer". Далі виберіть "RxBufferSize", який є достатньо великим, щоб вмістити вашу найбільшу команду плюс одну. Виберіть символ командного термінатора "CommandTerminator." Найчастіше це встановлюється як повернення каретки, так і лінійний канал. Якщо ваші команди містять два або більше параметрів, виберіть роздільник параметрів "Param\_Delimiter". Звичайними роздільниками команд зазвичай є пробіл або кома.

Контрольні символи нижче вибраного значення також можуть ігноруватися. Більшість керуючих символів знаходяться в діапазоні від 0 до 31. Слід встановити "IgnoreCharsBelow" на 32, щоб ігнорувати ці символи. Можна встановити цей параметр на "1", якщо всі символи дійсні. На символ, вибраний для командного термінатора (CommandTerminator), параметр "IgnoreCharsBelow" не впливає. Буфер команд працює в режимі обслуговування переривань UART RX, щоб збирати символи в буфері, доки не буде отриманий символ термінатора команди. У цей час встановлюється прапор, який сигналізує про те, що буфер команд готовий до читання. Буфер прийому може бути доступний безпосередньо, читаючи масив `INSTANCE_NAME_aRxBuffer` або використовуючи функції `szGetParam ()` або `szGetRestOfParams ()`. Якщо більше символів отримано, ніж буфер-розмір -1, наступні символи ігноруються. Буфер команд збирає символи, поки буфер не заповнений або не буде виявлено термінатор команди. Будь-які символи, отримані після будь-якої з цих двох умов, ігноруються, поки не буде

виконана команда CmdReset. Після виконання команди CmdReset прошивка RX ISR знову починає збирати символи.

Зробивши короткий огляд та аналіз інтерфейсів передачі, можемо зупинитись на тому, що для підключення датчиків найкраще підійде шина I2C, оскільки вона має в своєму запасі можливість підключення до 127 пристроїв, а швидкість передачі даних в швидкісному режимі може досягати 3.4 Мбіт/сек з тактом 3.4 МГц. Для підключення ж пристроїв для контролю, можливе використання шини даних 1-Wire, використовуючи адаптер переходу від COM-порта на порт RJ-11.

## 2.5 Розробка електричної принципової схеми

Raspberry Pi (CM1), обчислювальний модуль 3 (CM3) та обчислювальний модуль 3 Lite (CM3L) - це сумісні з DDR2 SODIMM системні модулі (SoMs), які включають процесор, пам'ять та Flash eMMC (для CM1) та CM3) та енергетичної схеми. За допомогою цих модулів розробник може використовувати апаратні та програмні шляхи Raspberry Pi у власних системах. Крім того, цей модуль має додаткові інтерфейси вводу / виводу, доступні у моделях Raspberry Pi A / B. Це відкриває більше можливостей для варіацій [15].

Отже, для підключення датчика температури, тиску та вологості знадобиться використати I2C шину, а саме третій та п'ятий піни GPIO02, GPIO03, подати на датчик напругу 3.3 В та вивести нуль. Як бачимо зі специфікації мінікомп'ютера, відповідна напруга є на першому та сімнадцятому пінах.

Для підключення дисплею знадобиться підключити його до напруги ємністю 5 В та підключити піни DB4, DB5, DB6, DB7 для чотирибітного передавання інформації. Опишемо підключення кожного з пінів почергово використавши список:

1 : GND

2 : 5V

3 : Contrast (0-5V)

- 4 : RS (Register Select) -> GPIO25
- 5 : R/W (Read Write) -> GND
- 6 : Enable or Strobe -> GPIO24
- 7,8,9,10 : Data Bit 0/1/2/3 – NOT USED
- 11: Data Bit 4 -> GPIO23
- 12: Data Bit 5 -> GPIO17
- 13: Data Bit 6 -> GPIO27
- 14: Data Bit 7 -> GPIO22

Використавши всі необхідні елементи отримуємо наступну електричну принципову схему наведену в додатку Б. Перелік використаних елементів наведено в додатку В.

## 2.6 Висновок

Отже в другому розділі було розглянуто основні види мінікомп'ютерів, які можна було б використати в реалізації роботи. Наведено аргументацію вибору Raspberry Pi 3 Model B, а саме порівняно з іншими низьку вартість за не гірші характеристики.

Було створено структурну модель пристрою та вибрано електронні компоненти для його реалізації, а саме датчик атмосферного тиску, температури і вологості BME280 та LCD дисплей WH 1602.

Крім того було розроблено і наведено електричну принципову схему, на якій зображено підключення датчика за допомогою I2C інтерфейсу з напругою 3.3В та дисплею до 5В напруги.

### 3 МАКЕТУВАННЯ ТА НАЛАШТУВАННЯ СИСТЕМИ МЕТЕОРОЛОГІЧНИХ ДОСЛІДЖЕНЬ

#### 3.1 Розробка алгоритму функціонування системи

Суть роботи пристрою полягає в наступному:

- Зчитування даних з датчиків;
- вивід даних на екран;
- підключення до бази даних та запис у неї;
- видалення застарілих даних з бази даних;
- формування графіків на основі показників та запис до HTML файлу;
- закриття бази даних;
- коригування дій обладнання на основі нової отриманої інформації;
- відправка даних на сайт.

Тобто необхідно встановити базу даних та розташувати сервер за допомогою Apache або вручну. Створити скрипт формування графіків. Організувати механізм відправки та прийому даних на сервер.

#### 3.2 Підготовка програмного забезпечення

Для початку роботи з обраним мінікомп'ютером необхідно встановити операційну систему. На вибір є такі системи як: Raspbian, OSMC, OpenELEC, ArchLinux, Pi MusicBox, RetroPie, OpenMediaVault. Відмінністю встановлення операційної системи на даний пристрій порівняно з стандартними комп'ютерами є те, що тут відсутній BIOS, та не має ні жорсткого диску встановленого всередині ні підтримки встановлення через змінні носії (DVD, CD, USB-Flash). Реалізована система зберігання за допомогою мікро – SD карт, на яку і необхідно записати обраний образ дистрибутиву [12].

На сьогодні є величезна кількість різноманітних дистрибутивів та варіацій їх встановлення доступних для Raspberry Pi. Система Raspbian – основна та офіційно підтримувана операційна система, а саме тому найбільш підтримувана

та стабільна система на основі Debian, вона є оптимізованою спеціально під Raspberry Pi.

Встановити систему Raspbian на пристрій Raspberry Pi не складе труднощів. Необхідно завантажити Raspbian і записати зображення диска на карту microSD, потім завантажити Raspberry Pi на картку microSD. Для цього проекту знадобиться карта microSD (мінімум на 8 ГБ), комп'ютер з слотом для неї, і, звичайно, Raspberry Pi та основні периферійні пристрої (миша, клавіатура, екран та джерело живлення). Це не єдиний метод встановлення Raspbian., але це корисна методика для вивчення, оскільки вона також може бути використана для установки і інших операційних систем на Raspberry Pi.

Зображення диска Raspbian стискається, тому потрібно розпакувати його. У файлі використовується формат ZIP64, тому потрібно використати певні програми, щоб розпакувати їх, можна скористатись утилітою 7-Zip.

Після цього необхідно записати образ на SD-карту, для чого буде використано програму Win32 Disk Imager. Необхідно вибрати місце призначення (важливо переконатися вибрана необхідна карта microSD!) та зображення диска (нерозпакований файл Raspbian). Потрібно вибрати, перевірити, а потім натиснути кнопку для запису.

Після того, як зображення диска буде записано на карту microSD, система готова до роботи. Залишилось вставити карту в роз'єм та підключити периферійні пристрої та джерело живлення. Поточне видання Raspbian завантажиться безпосередньо на робочий стіл [13].

На рисунку 3.1 наведено послідовність дій [15].

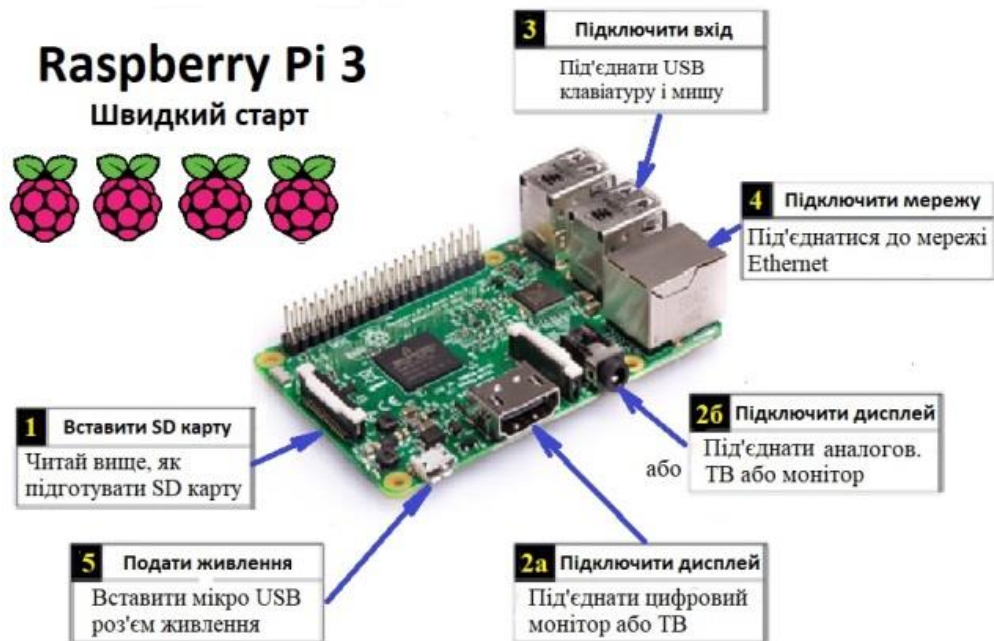


Рисунок 3.1 – Послідовність дій при ввімкненні

При першому запуску здійснюється автентифікація користувача, це відбувається автоматично. Відразу ж відбувається запуск не рядка терміналу, а пустого робочого столу

Перш ніж буде можливо підключитися до мережі Pi, спочатку потрібно виконати конфігурацію Pi. Потрібно помістити своє ім'я мережі WiFi та пароль WiFi у файл конфігурації.

Якщо є клавіатура, миша та монітор HDMI, безпосередньо підключитися до комп'ютера буде легко. Просто необхідно увійти у свій Pi та виконати наведені нижче дії. Коли ви ввійдете в Pi, вам буде запропоновано ім'я користувача та пароль. Облікові дані за замовчуванням - це ім'я користувача pi і пароль raspberry. [14].

На Raspberry Pi, необхідно переміститись до директорії `/wpa_supplicant`. для чого необхідно на терміналі ввести команду `cd /etc/wpa_supplicant`

Слід відредагувати файл, який там знаходиться, під назвою `wpa_supplicant.conf`. Для цього спершу скористаємось командою "ls", щоб переконатися, що вона там була, а потім редактором nano так, як зображено на рисунку 3.2



```

pi@raspberrypi: /etc/wpa_supplicant
pi@raspberrypi:~ $ cd /etc/wpa_supplicant
pi@raspberrypi:/etc/wpa_supplicant $ ls
action_wpa.sh  functions.sh  ifupdown.sh  wpa_supplicant.conf
pi@raspberrypi:/etc/wpa_supplicant $ sudo nano wpa_supplicant.conf

```

using the "ls" command to verify that the "wpa\_supplicant.conf" file is there

edit the "wpa\_supplicant.conf" file

Рисунок 3.2 – Послідовність дій при ввімкненні

У файлі `wpa_supplicant.conf` слід перейти донизу та додати стрічку `network={ssid="Пароль" psk="Назва мережі"}`, відповідно змінивши дані, які необхідні для входу до мережі WiFi.

Збережемо файл натиснувши `Ctrl-O` і вийдемо з редактора натиснувши `Ctrl-X`.

Перезавантажимо Pi. Використовувати необхідно команду вимкнення, щоб дати час редактору та ще будь-якому процесу записати зміни на SD-карту.

Через хвилину або близько того Raspberry Pi повинен підключитися до WiFi. Щоб налаштувати зв'язок з комп'ютером через мережу, потрібно дізнатися IP-адресу пристрою.

Один із способів зробити це - відкрити веб-браузер на комп'ютері та перейти на веб-сторінку модему WiFi, у більшості модемів є такий. Ввести IP-адресу модему, щоб підключитися до нього. Потім вводимо адресу в адресному рядку браузера, розташованому в лівій верхній частині вікна браузера, і натискаємо `Enter`. Зазвичай встановлюється адреса на кшталт `192.168.1.1` або `192.168.0.1`.

У розділі «Мережа» веб-сторінки можна спостерігати схему того, що пов'язано з модемом.

### 3.3 Налаштування програмного забезпечення отримання даних

В зв'язку з багатофункціональністю мінікомп'ютера та наявністю різноманітних інтерфейсів прийому та передачі інформації постає питання, який саме обрати.

Для магістерської кваліфікаційної роботи було обрано інтерфейс I2C в зв'язку з можливістю підключення до нього до 112 різноманітних пристроїв на високій швидкості, чого вистачить для використання у найбільших системах.

Для початку необхідно включити I2C шину для ініціалізації датчиків та можливості їх використання. Щоб це зробити, спочатку редагуємо файл `raspi-blacklist.conf` що знаходиться за адресою `etc/modprobe.d/` використавши відповідну команду переходу до директорії [18].

В даному файлі необхідно закоментувати рядок, що забороняє використання інтерфейсу (починається з `blacklist`).

Після цього необхідно відкоригувати файл `modules` в папці `etc`, для цього також скористаємось командою `nano`. Слід пам'ятати, що всі команди необхідно виконувати під правами суперкористувача (приставка `sudo`).

Тут необхідно додати наступні рядки:

```
i2c-bcm2708
```

```
i2c-dev
```

Потім потрібно змінити файл `config.txt` в папці `boot`, додавши до нього параметр `dtparam=i2c_arm=on` для того, щоб мати можливість використовувати спеціально призначені піни для використання інтерфейсу замість стандартних GPIO.

Після цього необхідно встановити `i2c-tools` утиліти за допомогою команди `python-smbus i2c-tools`

Тепер за допомогою команди `reboot` перезавантажуємо мінікомп'ютер та використавши команду `i2cdetect -y 1` можемо побачити всі підключені пристрої до шини I2C, як на рисунку 3.1 [19].

```

0 1 2 3 4 5 6 7 8 9 a b c d e f
00:  -----
10:  -----
20:  -----
30:  -----
40:  -----
50:  -----
60:  -----
70:  ----- 76 -----

```

Рисунок 3.2 – Результат виконання команди `sudo i2cdetect -y 1`

В даному випадку можемо спостерігати на першій шині лише один пристрій, який знаходиться за адресою 0x76.

На жаль при використанні на великих дистанціях можлива втрата сигналу, для попередження цього можливо знизити швидкість передачі даних на шині, або ж можливе використання буферів чи підсилювачів струму. В даному випадку система зможе збільшити свою протяжність в декілька разів.

Ці пристрої досить схожі, у кожному випадку використовується певна форма посилення пристрою та схеми для підвищення рівня струму сигналу. Однак підсилювач струму завжди намагається помножити струм на задану суму. Отже, подібна за дією підсилювач напруги, який завжди намагається помножити струм сигналу на задану величину. Буфер відрізняється від підсилювача струму тим, що він має на меті забезпечити необхідний рівень струму для того, щоб підтримувати напругу сигналу, про яке йому сказано. Отже, він матиме більший коефіцієнт посилення струму при підключенні до більш вимогливого навантаження. Приклад буферів наведено на рисунку 3.3.

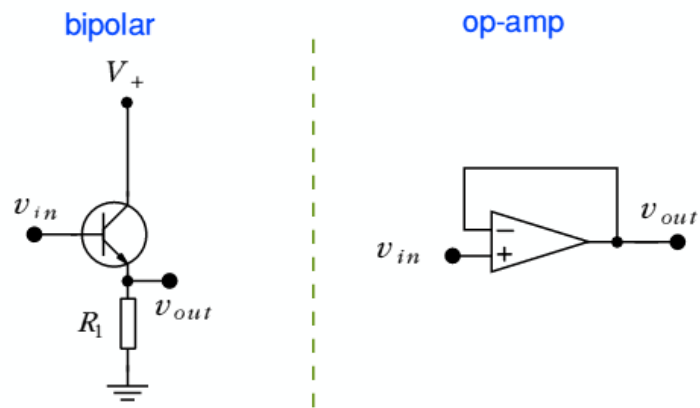


Рисунок 3.3 – Схема найпростіших буферів

На рисунку 3.4 показаний приклад двоступеневого підсилювача струму в типовому застосуванні. Фотодіоди, які часто використовуються для виявлення світла, прагнуть працювати, поглинаючи фотони і вивільняючи носії безкоштовного заряду. Тоді можливо застосувати різницю потенціалів по стику фотодіода і «підмітати» ці заряди, щоб отримати поточний сигнал, пропорційний кількості фотонів в секунду, що вражає фотодіод.

Оскільки знадобиться близько  $10^{16}$  електронів в секунду, щоб отримати амперний струм, а ефективність типового фотодіоду набагато менша, ніж «один електрон на фотон», це означає, що вихідний струм від такого фотоприймача часто досить малий. За допомогою підсилювача струму можливо підвищити цей вихід до більш зручного рівня. У прикладі використовуються два біполярні транзистори, один NPN-тип, інший PNP-тип із посиленням струму та. Для типових малосигнальних транзисторів, таким чином, можна очікувати, що пара подібних етапів підсилює струм між 250 та 250 000 залежно від вибраних транзисторів.

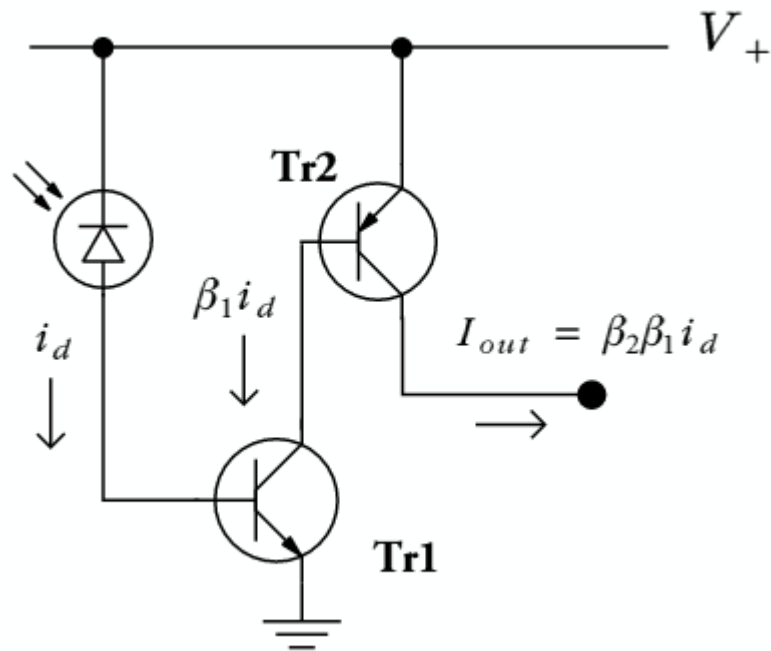


Рисунок 3.4 – Підсилювач струму

Насправді, якщо є необхідність, то можливо перетворити це на напругу, застосувавши отриманий вихідний струм до резистора. Результатом було б змусити ланцюг вести себе як підсилювач високої напруги.

### 3.4 Налаштування серверу та передачі інформації на сайт

Для встановлення необхідного програмного забезпечення, а саме бази даних SQL та Apache [20] можна скористатись командою install. Пакети з даними продуктами мають назви apache2 та sqlite3.

І відразу можна запустити сервер використавши команду service apache2 start. Далі слід створити тестову сторінку для перевірки працездатності. В неї можливо записати команду для виведення інформації phpinfo(). Результат зображено на рисунку 3.5

PHP Version 5.6.20-0+deb8u1		
System	Linux raspberrypi 4.1.19+ #858 Tue Mar 15 15:52:03 GMT 2016 armv6l	
Build Date	Apr 28 2016 00:41:17	
Server API	FPM/FastCGI	
Virtual Directory Support	disabled	
Configuration File (php.ini) Path	/etc/php5/fpm	
Loaded Configuration File	/etc/php5/fpm/php.ini	
Scan this dir for additional .ini files	/etc/php5/fpm/conf.d	
Additional .ini files parsed	/etc/php5/fpm/conf.d/05-opcache.ini, /etc/php5/fpm/conf.d/10-pdo.ini, /etc/php5/fpm/conf.d/20-json.ini, /etc/php5/fpm/conf.d/20-readline.ini	
PHP API	20131106	
PHP Extension	20131226	
Zend Extension	220131226	
Zend Extension Build	API220131226.NTS	
PHP Extension Build	API20131226.NTS	
Debug Build	no	
Thread Safety	disabled	
Zend Signal Handling	disabled	
Zend Memory Manager	enabled	
Zend Multibyte Support	provided by mbstring	
IPv6 Support	enabled	
DTrace Support	enabled	
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip	
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls, tlsv1.0, tlsv1.1, tlsv1.2	
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk	

Рисунок 3.5 – Стартова сторінка

Розглянемо створення скрипта обробки і отримання даних. Спочатку потрібно опитати датчик за допомогою команди `ps.get_data()` та записати отриманий результат до змінної `ps_info`.

Необхідно вивести поточні данні в консоль та на дисплей для чого можна створити окрему обробку наведену в додатку.

Бібліотека роботи з дисплеєм наведена в додатку Г, а в додатку Д наводиться бібліотека роботи з датчиком ВМЕ280.

Для підключення до бази даних та запису інформації можна скористатись наступними командами:

```
con = sqlite3.connect(home_dir + database_name)
cur = con.cursor()
SQL="INSERT INTO weather VALUES({0}, {1}, {2},
{3})".format(time.time(), ps_data['t'], ps_data['p'], ps_data['h'])
cur.execute(SQL)
con.commit()
```

Для видалення застарілих даних можна використати наступну команду:

`SQL = "DELETE FROM weather WHERE (id < {0})".format(time1)`, де `time1` кожен раз є всіма датами більшими за сьогоднішню на 30 днів. Не варто забувати про команду `con.commit()`, за допомогою якої затверджуються зроблені зміни

Зчитується шаблон для заповнення HTML файла та формуються графіки (для температури):

```
f = open(home_dir+'templates/index.tpl', 'r')
txt = f.read()
f.close()
date_time = datetime.datetime.fromtimestamp(time.time()).strftime('%Y-%m-%d
%H:%M')
txt = re.sub('{temperature_unit}', units[temperature_field], txt)
txt = re.sub('{temperature}', str(convert(ps_data['t'], units[temperature_field])),
txt)
txt = re.sub('{temperature24h}', get_chart_data(temperature_field, 1), txt)
txt = re.sub('{temperature7d}', get_chart_data(temperature_field, 7), txt)
```

Після проведених операцій потрібно записати всю інформацію до HTML файлу за допомогою команди `f.write(txt)`, не забувши після цього закрити з'єднання за допомогою `con.close()`.

Для відправлення даних на сайт можна скористатись наступним кодом:

```
s="{0}:{1}:0:{2}:".format(int(ps_data['p']), int(ps_data['t']), int(ps_data['h']))
response = urllib2.urlopen("http://mysite.com.ua/getdata.php?data="+s)
```

Повністю лістинг скрипта наводиться в додатку Е.

Для прийому цих даних на сервері був створений скрипт, який відкриває з'єднання та записує всі необхідні дані за допомогою команди `fwrite`. Як результат маємо робочу сторінку сайту, яка представлена на рисунку 3.6.

Temperature: 23.8 °C  
 Pressure: 752.26 mm Hg  
 Humidity: 26.07 %

### Last 24 hours

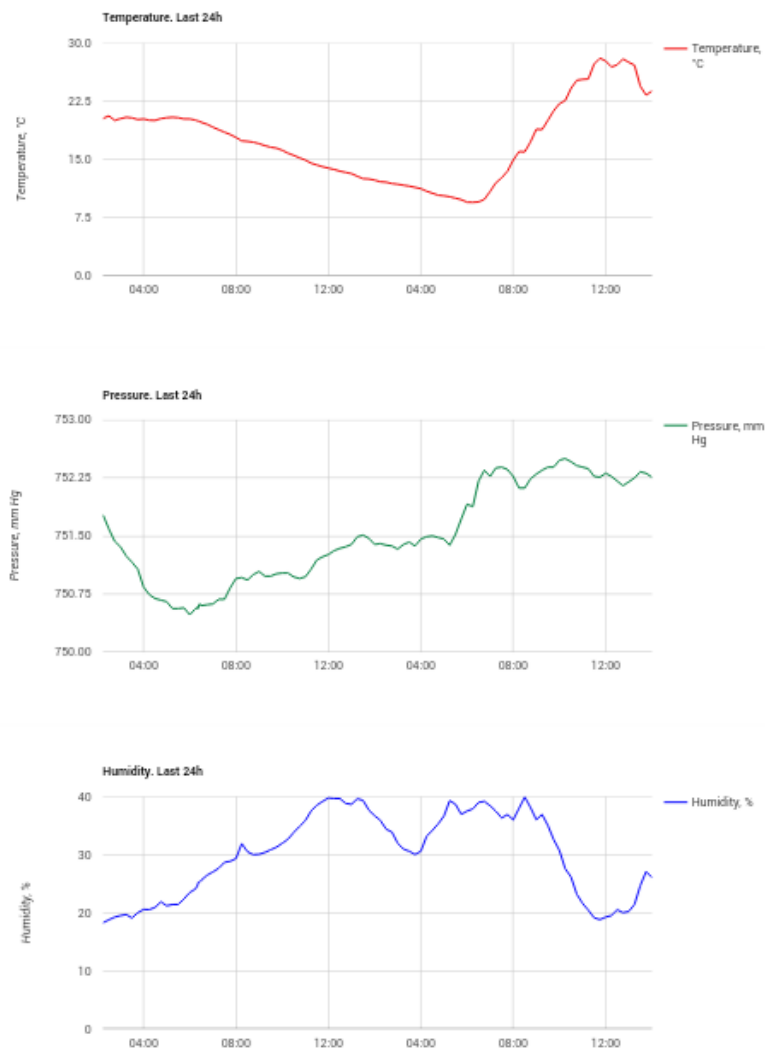


Рисунок 3.6 – Сторінка сайту з отриманою інформацією

Для запуску скрипта збору та обробки даних було встановлено запуск кожних 10 хвилин:

```
sudo echo "*/10 * * * * root /home/pi/weather/weather.py" >> /etc/crontab
```

В додатку Є наведено звичайну та мобільну версію сайту з обробленими даними.

### 3.5 Реалізація взаємодії з іншими елементами системи

Для реалізації роботи з іншими елементами системи було обрано інтерфейс 1-Wire з переходом на інтерфейс 6P-4C за допомогою RJ11, в зв'язку з



можливістю підключення на великі відстані, дешевизною кабелів телефонних передач 4P4C порівняно з кабелями витої пари 8P8C. Крім того для організації завдання немає необхідності в використанні 8 контактів.

В якості ведучого однопровідної системи тут також виступає мінікомп'ютер, укомплектований одним з адаптерів 1-Wire-лінії типу ML97 # - ## або типу ML94 #. В цьому випадку адаптер з'єднується тим же телефонним кабелем відразу з декількома відомими 1-Wire-компонентами, реалізованими на базі функціональних пристроїв ML ##, різних типів, які можуть працювати або на паразитному харчуванні, або з використанням шини зовнішнього живлення EXT\_POWER, якщо їх загальне споживання в будь-якому режимі роботи системи не перевищує 4мА.

Точніше безпосередньо адаптер ML97 # - ## (або ML94 #) з'єднується тільки з одним приймальним гніздом найближчого до нього однопровідного пристрою ML ##. Однак, наявність у складі конструкції будь-якого функціонального пристрою ML ## двох, паралельно включених прийомних роз'ємів-гнізд RJ11 (6p4c), розміщених на торці їх корпусу, дозволяє легко реалізувати структуру однопровідної лінії у вигляді загальної шини. Цього можна досягти при з'єднанні окремих функціональних пристроїв ML ## між собою відрізками плоского телефонного кабелю необхідної довжини, оформлених з обох кінців стандартними телефонними вилками (джек) типу RJ11 (6p4c). Кінцевим варіантом є схема підключення зображена на рисунку 3.7.

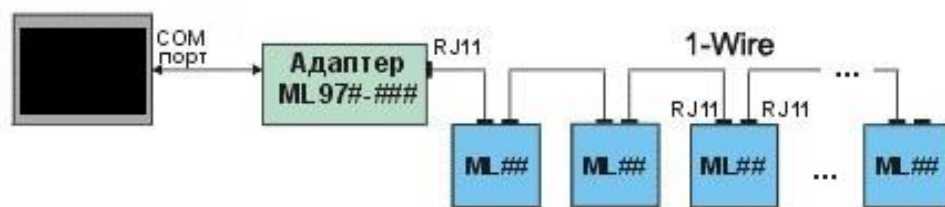


Рисунок 3.7 – Принцип підключення до керованих пристроїв

Загальна протяжність 1-Wire-лінії при цьому може складати  $\sim 100\text{м}$ , при кількості використовуваних пристроїв ML ## до 20шт.

Якщо ж є необхідність збільшити кількість пристроїв до 50, або збільшити протяжність самої системи, можливо додати окремий блок живлення ML00#-XX-### та роздвоювач RJ11 для передачі необхідного заряду. На рисунку 3.8 наведено більш розширену схему підключення.

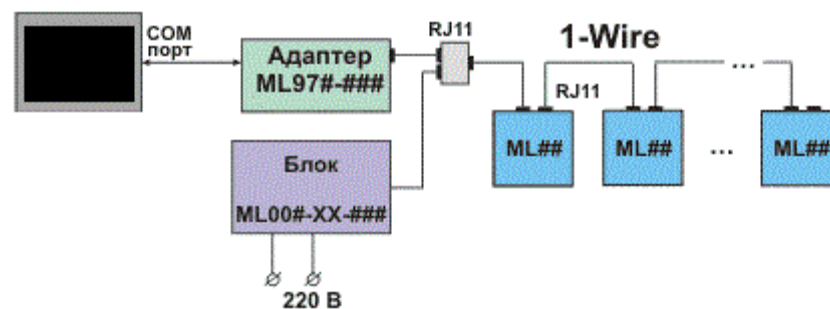


Рисунок 3.8 – Розширена схема підключення до керованих пристроїв

Рівень напруги зовнішнього живлення, що надходить в подібну мережу, вибирається значно більшим рівня, необхідного для живлення будь-яких компонентів, що входять до складу пристроїв ML ##, що допустимо завдяки наявності в схемі цих приладів спеціальних вузлів перетворення зовнішнього живлення. Сполучення 1-Wire-магістралі з блоком живлення здійснюється або завдяки спеціальним розділювачами телефонних розеток системи RJ11, або через вільні прийомні гнізда будь-якого з пристроїв ML ##, що входять до складу формованої однопровідної системи.

Наявність додаткової енергії в 1-Wire-лінії дозволяє значно поліпшити її загальну стійкість до спотворення сигналів.

Для прокладення ліній на більші відстані можливо виконати підключення джерел струму безпосередньо до кожного пристрою ML##. Перш за все, при виникненні проблем з передачею інформації в подібних однопровідних структурах, слід використовувати програмні методи обслуговування елементів,

які можуть бути реалізовані завдяки вибору оптимального в кожній конкретній ситуації робочого режиму мікросхем DS2480B для COM-порту. Ці апаратні драйвера, призначені спеціально для обслуговування протяжних 1-Wire-ліній, і є невід'ємною частиною будь-якого з адаптерів незалежно від інтерфейсу підключення.

З їх допомогою реалізується механізм керованої активної підтяжки лінії даних, а також забезпечується можливість зміни часових співвідношень і форми фронтів сигналів на відомою 1-Wire-магістралі, що дозволяє оптимізувати роботу використовуваного адаптера при обслуговуванні однопровідних ліній з індивідуальними параметрами.

### 3.6 Налаштування роботи з пристроями через інтерфейс 1-wire

Для організації роботи з пристроями було вибрано варіант їх підключення через реле для забезпечення максимальної функціональності Реле - це в основному вимикачі, перемикачі, якими ви керуєте (включаєте чи вимикаєте), застосовуючи або знімаючи певну напругу до релейного пристрою. Реле не потрібні для більшості проектів на базі мікроконтролерів, зможе жити зовнішні пристрої безпосередньо зі своїх портів вводу / виводу. Реле стають в нагоді, коли потрібно відкрити / закрити зовнішній ланцюг або коли схема повинна контролювати більш високі напруги, ніж може забезпечити ваша система.

Було використано реле в пристрої, контрольованому мінікомп'ютером, тому що була необхідність імітації натискання фізичної кнопки, яка вимагала можливості комутації, а не можливості передавати вихідну напругу з плати. Також можна використовувати реле для потреби увімкнення та вимкнення сторонніх пристроїв які живляться безпосередньо від змінного та незалежного електричного струму , тому що Raspberry Pi не може цього зробити безпосередньо.

Існує два поширених типи реле: механічні реле та твердотільні реле. Механічні реле використовують електромагнітну котушку та фізичний вимикач;

при подачі напруги перемикач активується. Твердотільні реле дають однаковий результат, але не мають механічних компонентів; натомість вони використовують електронні компоненти, щоб виконувати ту саму роботу.

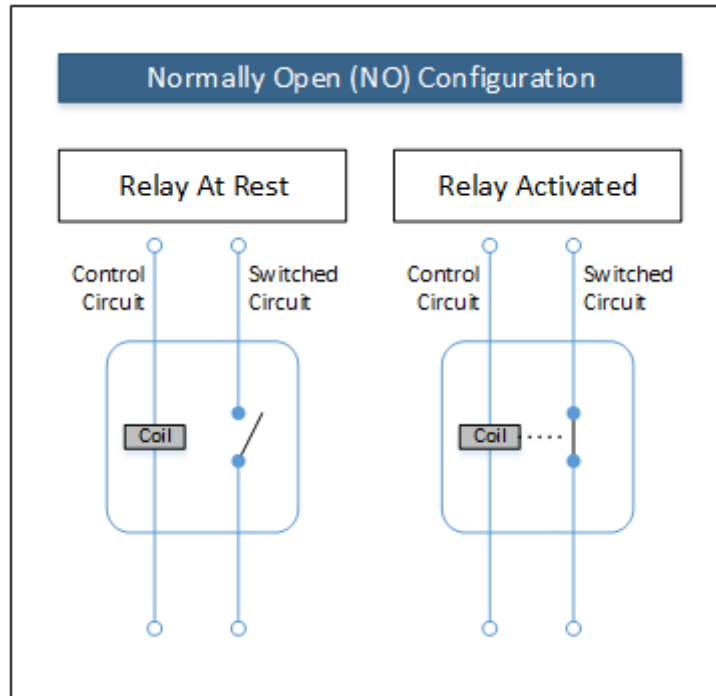


Рисунок 3.9 – Режим роботи реле Normally Open

На рисунку 3.9 можна побачити зображення реле в режимі Normally Open (NO). У цій конфігурації, коли до ланцюга управління не подається напруга (реле в частині спокою), комутований контур відключається і струм не може протікати через з'єднання. Коли відбувається подача відповідної напруги до ланцюга управління, електромагнітна котушка в реле активується і лишає перемикач закритим, даючи можливість струму протікати через комутований контур.

У режимі Normally Closed (NC), що зображений на рисунку 3.10, коли реле знаходиться в спокої та на ланцюг управління не подається напруга, струм продовжує протікати до пристрою. Коли ви активуєте реле, приклавши відповідну напругу до ланцюга управління, електромагнітна котушка в реле активується і відтягує вимикач, зупиняючи будь-який струм, що проходить через комутований контур.

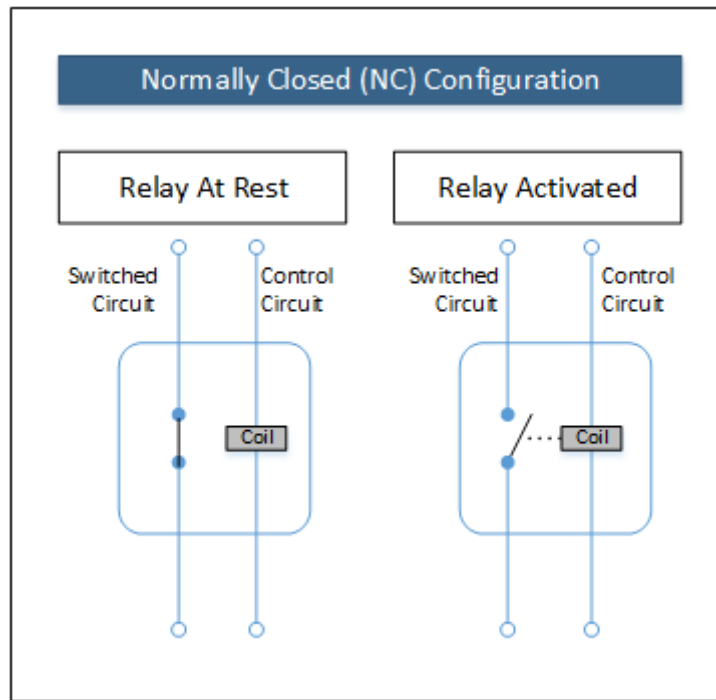


Рисунок 3.10 – Режим роботи реле Normally Closed

Атрибут кидання комутатора описує кількість контурів контуру, що надаються комутатором. Одинарний кидок (ST) має лише один контур. Коли вимикач кинутий в один бік, струм протікає по ланцюгу, при ньому кидається іншим шляхом, ланцюг розривається і струм не тече. Перемикач подвійного кидання (DT) пропонує дві контурні контури. З вимкненим вимикачем в один бік струм протікає по одній із контурних контурів, при ньому кидається інший шлях, струм тече через інший контур. DT-перемикач також може мати центральне положення відключення між двома параметрами контуру контуру.

Для початку виконаємо підключення інтерфейсу 1wire. Спочатку відкриємо термінал та введемо команду `sudo raspi-config`. Це відкриє спеціальну утиліту для налаштувань, зображену на рисунку 3.11

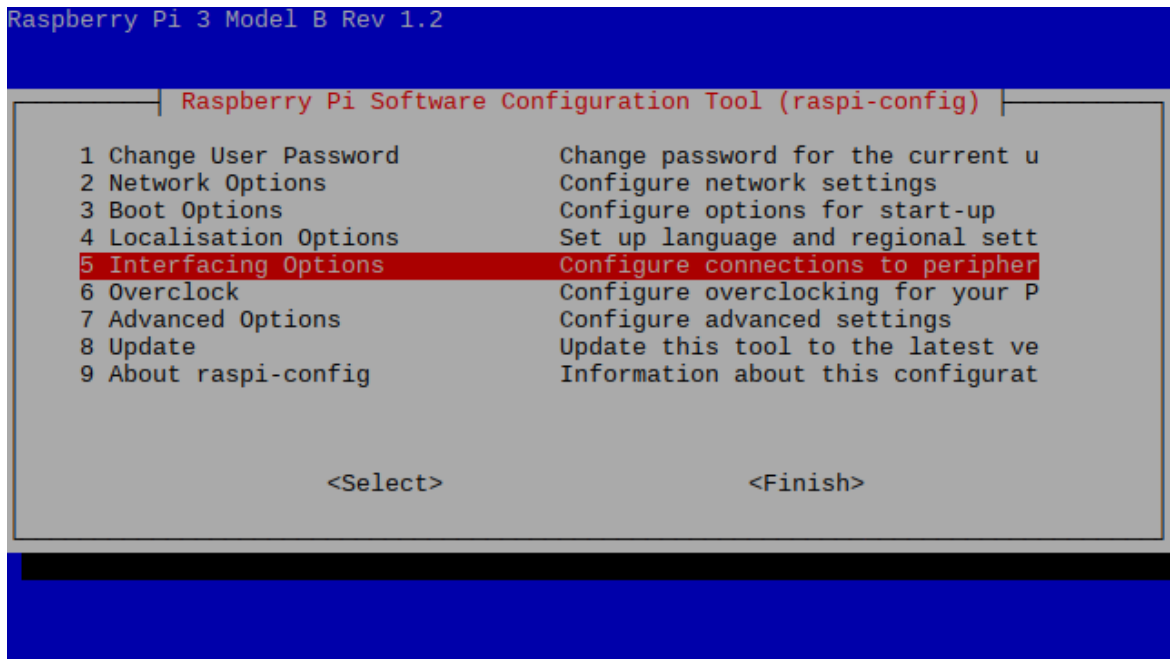


Рисунок 3.11 – Головне вікно утиліти налаштувань

Відкриємо налаштування інтерфейсів та переключимо режим 1-Wire в статус enabled. Результатом роботи буде відображення повідомлення, зображеного на рисунку 3.12 про успішну активацію.

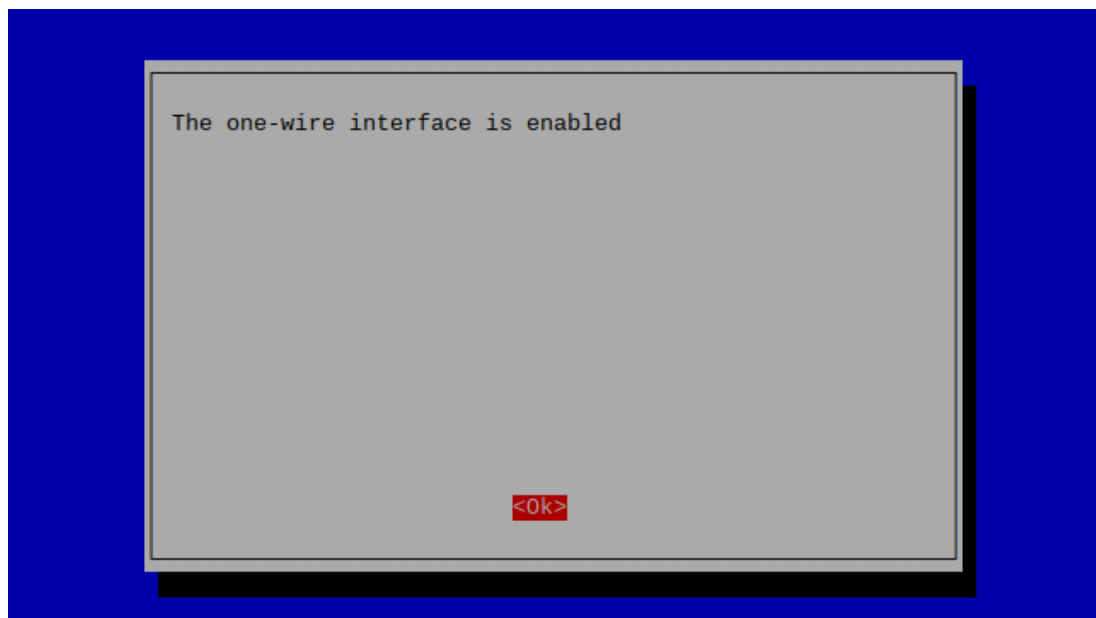


Рисунок 3.12 – Головне вікно утиліти налаштувань

Після цього необхідно перезавантажити пристрій. Після перезавантаження необхідно перейти до перевірки ефективності підключення, для цього слід додати до ядра модулі `wl-gpio` і `wl-therm` за допомогою команди `sudo modprobe`. Перейшовши в каталог з пристроями буде наведено, які з них доступні, повинне відобразитись наступне повідомлення:

```
pi@raspberrypi /sys/bus/w1/devices $ ls
11-0003211c62d2 w1_bus_master1
```

В даному випадку `11-0003211c62d2` – і є адреса шуканого пристрою. Тепер за допомогою команди `MATCH ROM` можливо звертатись до даного пристрою згідно з його адресою. Створимо таблицю, до якої будемо заносити всі підключені адреси.

Для подальшої роботи необхідно організувати додаткову таблицю `PRIV` в базі даних з проведенням необхідних прив'язок датчика з елементом яким він повинен керувати та прописом умов допустимих показів датчика. Для цього просто створимо нову таблицю з двома стовбцями `datchik` та `pristr`, дані яких отримуємо зі списку підключених пристроїв до станції. Перший стовпчик заповнюємо автоматично виходячи з перевірки пристроїв підключених до інтерфейсу `I2C`, другий задаємо вручну

Необхідно змінювати стан реле (вмикати пристрій) лише при певних умовах. Для цього буде актуальною перевірка даних з датчику (для прикладу температура).

Алгоритм обробки наведено на рисунку 3.13

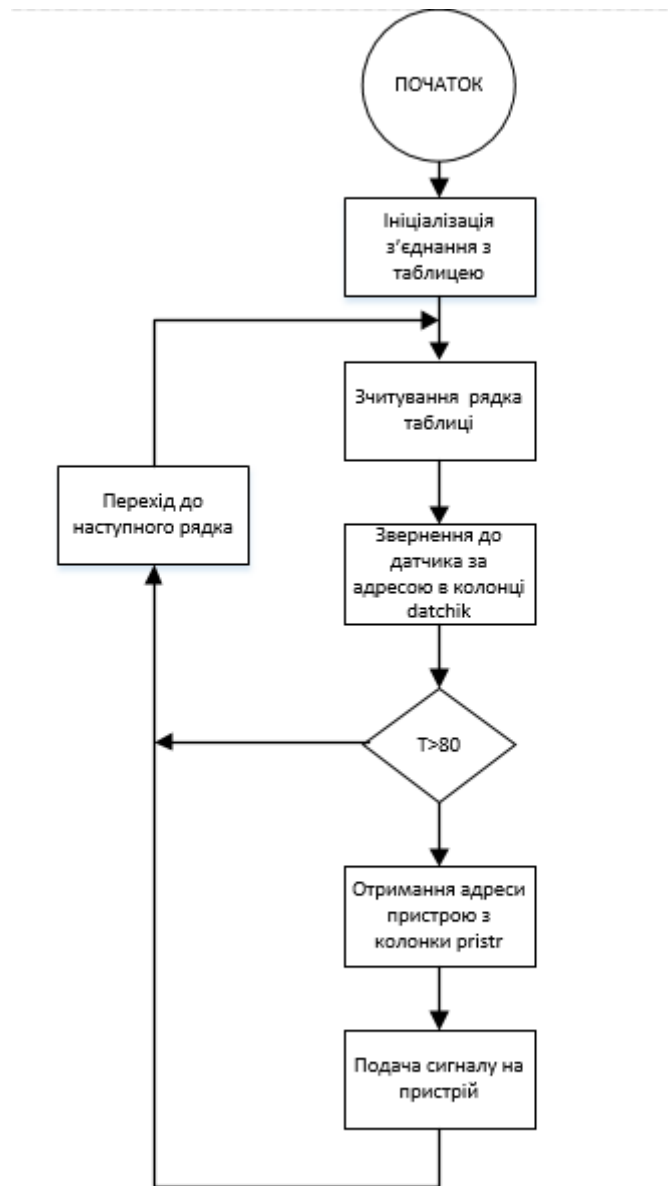


Рисунок 3.13 – Алгоритм взаємодії з керованим пристроєм

Використовуючи команду `foreach()` проходимо повністю по всій таблиці. Робимо звернення до датчика за адресою, яка знаходиться в `$datchik` та записуємо до змінної `temp`. Робимо перевірку на відповідність температури і якщо вона не відповідає необхідній, посилаємо сигнал до пристрою, адресою якого виступає `$pristr`. В результаті реле перемикається і пристрій починає працювати. Алгоритм роботи пристрою повністю наведено в додатку Ж.

Отже було створений пристрій, який може використовуватись як для створення розумного будинку в кожній оселі, так і для інтеграції на різноманітних



підприємствах. Було створено все необхідне програмне забезпечення для тестової версії, за бажанням замовника можлива зміна використовуваних датчиків. Для цього необхідно лише підключити нові скрипти з обробки даних. Другорядні керувані пристрої можливо підключати за допомогою додаткового елемента, як реле, що спрощує необхідність в зміні налаштованої системи.

Також доцільним буде використання пристрою на видобувних підприємствах для постійного контролю стану в віддалених точках входу. Тобто встановивши датчик газу, можливо ввімкнути систему вентиляції на повну потужність для швидкого виводу робітників, за допомогою датчиків тиску можливий контроль протікання з підземних джерел і при збільшенні показників до вищих від звичного стану відразу ж ініціювати закриття шлюзів до даних відділів.

Для керування вже налаштованої системи потрібен лише один оператор, а вибір інтерфейсів I2C та 1wire надає можливість здійснювати гаряче підключення нових датчиків та пристроїв до системи.

## 4 ЕКОНОМІЧНА ЧАСТИНА

### 4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності [21].

Результатом магістерської кваліфікаційної роботи є розробка системи метрологічних досліджень на базі мінікомп'ютера. Для проведення технологічного аудиту залучено трьох незалежних експертів. У нашому випадку такими експертами є: Кадук Олександр Володимирович (к.т.н., доцент каф. обчислювальної техніки ВНТУ), Гарнага Володимир Анатолійович (к.т.н., доцент каф. обчислювальної техніки ВНТУ) та Богомолів Сергій Віталійович (к.т.н., доцент каф. обчислювальної техніки ВНТУ). Оцінювання комерційного потенціалу буде здійснене за критеріями, що наведені в таблиці 4.1.

Таблиця 4.1 - Критерії оцінювання комерційного потенціалу розробки бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-тері-й	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-тер.	0	1	2	3	4
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
<b>Ринкові перспективи</b>					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
<b>Практична здійсненність</b>					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, з військово-промислового комплексу	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються

Продовження таблиці 4.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу експертами розробки зведено в таблицю 4.2.

Таблиця 4.2 - Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	1 – Кадук	2 – Гарнага	3 – Богомолів
	Бали, виставлені експертами:		
1	3	3	3
Ринкові переваги (недоліки):			
2	2	2	2
3	4	3	3
4	4	3	4
5	3	3	3
Ринкові перспективи			
6	2	2	2
7	3	4	3
Практична здійсненність			
8	2	3	3
9	3	2	3
10	3	3	3
11	3	4	3
12	3	3	3
Сума балів	СБ <sub>1</sub> =35	СБ <sub>2</sub> =35	СБ <sub>3</sub> =35
Середньоарифметична сума балів $\overline{СБ}$	35		

За даними таблиці 4.2 можна зробити висновок, щодо рівня комерційного потенціалу розробки. Зважимо на результат й порівняємо його з рівнями комерційного потенціалу розробки, що представлено в таблиці 4.3.

Таблиця 4.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Рівень комерційного потенціалу розробки, становить 35 балів, що відповідає рівню «середній».

Головними конкурентами на ринку є розробники метрологічних систем віддаленого керування на базі мікроконтролерів (Arduino, MediaTek), а також мікрокомп'ютерів Raspberry, Odroid.

Перевагами систем метрологічних досліджень на базі мікроконтролерів є низька вартість та компактні розміри, недоліками – обмежені ресурси процесора та пам'яті. Перевагами мікрокомп'ютерних систем є наявність потужного ядра, великої кількості пам'яті та додаткової периферії, недоліками – більший розмір та висока вартість. Вартість готової системи на базі Arduino коштуватиме 2300 грн. Система створена на мікроконтролерах не буде задовольняти потреб поставленого завдання без багатьох модифікацій.

В якості аналога для розробки було обрано систему на базі Arduino Uno у поєднанні. Недоліком є обмежені ресурси процесора та пам'яті, необхідність поповнення рахунку SIM карти мобільного оператора. Також недоліком є необхідність придбання додаткової периферії при з'єднанні через WI-FI. Хоча аналог і випереджає новий продукт за кількістю можливих підключень датчиків, головною відмінністю являється швидкість роботи при великому навантаженні, де

набагато краще себе проявляє мінікомп'ютер в зв'язку з більшою швидкістю обрахунків.

Нова система набагато простіше підтримуватиме взаємодію датчиків та інших елементів системи за рахунок швидкодії, при запуску на ній встановленого серверу для моніторингу результатів.

У таблиці 4.4 наведені основні технічні показники аналога і нового програмного продукту

Таблиця 4.4 - Основні технічні показники аналога і нового програмного продукту

Показники	Аналог	Нова розробка	Відношення параметрів нової розробки до параметрів аналога
Функціональність	Наперед визначений функціонал	Функціонал системи легко змінити	Більш гнучкою системою являється нова розробка
Надійність	Високий рівень безперебійності при сталих умовах	Високий рівень безперебійності при сталих умовах	Однаковий рівень надійності
Сумісність	Для сумісності з іншими системами необхідний додатковий апаратний елемент	Велика сумісність з іншими системами	Нова розробка має набагато більшу сумісність ніж аналог
Супровід	Можлива модифікація та внесення виправлень	Можлива модифікація та внесення виправлень	Більш проста модифікація в новому пристрої
Економія ресурсів і часу	При великій кількості підключених модифікацій можливі затримки в роботі	Висока швидкодія	Краща швидкодія нового пристрою
Простота використання	Простий у використанні	Простий у використанні після проведення налаштувань	Для роботи «з коробки» краще підходить аналог

Створений пристрій є новим технічним рішенням, яке реалізовано на базі сучасного апаратного обладнання та програмного забезпечення.

Реалізація продукту здійснюється за договором продажу. Продукт приваблює легкістю використання, зрозумілістю інтерфейсу. Також привабливими є мобільність, віддалений доступ, можливість живлення від

автономних джерел енергії, можливість легкої модифікації та відкритість системи.

Основними напрямками впровадження розробки є виробництва з високими умовами небезпеки праці (добувна промисловість) та агрономічна промисловість.

Дана розробка може зменшити ризик втрати життя або травмування на виробництві та попередити ризик забруднення навколишнього середовища.

Розробка програмно - апаратно розроблена, необхідні залучення додаткових спонсорських коштів для реалізації промислового зразка для проведення демонстрації.

На даний момент створюється веб-сайт для організації продажу розробки через нього, триває пошук та збір інформації щодо потенційних інвесторів.

Планується просувати на ринок шляхом гарячих дзвінків за наперед сформованим скриптом з перенаправленням на сайт з більш детальною інформацією. Також, просування програмного продукту можливо здійснювати використовуючи персональний продаж, а також шляхом участі у різних заходах (конференціях та на різних виставках).

Планується проводити після продажну підготовку обладнання та ПЗ, а також можливість придбання підписки для післяпродажного сервісного обслуговування та налаштувань під клієнта. Для виконання розробки необхідні фахівці відповідної кваліфікації, відповідно необхідна наявність фінансових та трудових ресурсів.

Вартість розробки буде встановлюватись в залежності від кількості модифікацій.

4.2 Прогнозування витрат на виконання наукової роботи та впровадження результатів

Проведемо прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи для розробки програмного забезпечення, яке складається з таких етапів:

1-й етап: розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи;

2-й етап: розрахунок загальних витрат на виконання даної роботи;

3-й етап: прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

Виконаємо розрахунок витрат приймаючи до уваги те, що для розробки було залучено одного розробника.

1. Основна заробітна розробника-дослідника  $Z_o$ :

$$Z_o = \frac{M}{T_p} \cdot t \text{ [Грн]}, \quad (4.1)$$

де  $M$  – місячний посадовий оклад – 7000 грн;

$T_p$  – число робочих днів в місяці; приблизно  $T_p = (22)$  дні;

$t$  – число робочих днів роботи розробника-дослідника - 66.

Таким чином:

$$Z_o = \frac{7000}{22} \cdot 66 = 21000,00 \text{ (грн)}.$$

2. Додаткова заробітна плата  $Z_d$  розробника розраховується як 12% від основної заробітної плати:

$$Z_d = 0,12 \cdot 21000,00 = 2520,00 \text{ (грн)}.$$

3. Нарахування на заробітну плату  $H_{зп}$  розробника становить:



$$H_{zn} = (Z_o + Z_d) \cdot \frac{\beta}{100} \text{ [грн]}, \quad (4.2)$$

де  $Z_o$  – основна заробітна плата розробника;

$Z_d$  – додаткова заробітна плата розробника;

$\beta$  – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування – 22%.

$$H_{zn} = (21000,00 + 2520,00) \cdot 0,22 = 5174,40 \text{ (грн)}.$$

У спрощеному вигляді амортизаційні відрахування  $A$  в цілому розраховуємо за формулою 4.1:

$$A = \frac{Ц \cdot Т}{12 \cdot Т_B} \text{ [грн]}, \quad (4.3)$$

де  $Ц$  – загальна балансова вартість обладнання, приміщення тощо, грн;

$Т$  – фактична тривалість використання, міс;

$Т_B$  – термін використання обладнання, приміщень тощо, роки.

Розробка програмного забезпечення ведеться 3 місяці.

Зроблені розрахунки зведено до таблиці 4.5.

Таблиця 4.5 – Амортизаційні відрахування

Найменування	Балансова вартість, грн	Термін використання, р	Фактична трив. використання, міс.	Величина амортизаційних відрахувань, грн
Офісне приміщення	100000	25	3	100,00
Комп'ютер	15000	5	3	750,00
Монітор	5000	6	3	833,33
Всього				1683,33

Інформацію про матеріали, що використовуються при виготовленні даного інноваційного продукту внесено до таблиці 4.6.

Таблиця 4.6 – Матеріали, що використовуються при виготовленні даного продукту

Найменування матеріалу	Ціна за одиницю, грн.	Витрачено, шт.	Вартість витраченого матеріалу, грн
Папір (пачка)	82,00	1	82,00
Ручка	5,00	2	10,00
Всього			92,00

Під час розробки програмного продукту використовувались лише безкоштовні програмні засоби.

Витрати на енергію визначаються на основі витрат на одиницю продукції та тарифів на енергію за допомогою формули 4.2:

$$B_e = B \cdot П \cdot \Phi \cdot K_n \text{ [грн]}, \quad (4.4)$$

де  $B$  – вартість 1кВт електроенергії;

$П$  – установлена потужність обладнання, кВт;

$\Phi$  – фактична кількість годин роботи комп'ютера при створенні програмного продукту, годин;

$K_n$  – коефіцієнт використання потужності ( $K_n = 0,4$ ).

$$B_e = 2 \cdot 0,5 \cdot 528 \cdot 0,4 = 211,20 \text{ (грн)}.$$

Витрати за доступ до Інтернет можна розрахувати за формулою:

$$B_{di} = C_{di} \cdot T \text{ [грн]}, \quad (4.5)$$

де  $C_{di}$  – це ціна доступу за місяць;

$T$  – кількість місяців використання доступу до мережі.

$$V_{\text{ді}} = 150 \cdot 3 = 450,00 \text{ (грн)}.$$

Інші витрати  $I_v$  можна прийняти як 50% від суми основної заробітної плати розробника тобто:

$$V_{\text{ін}} = 50\% \cdot Z_o \text{ [грн]}, \quad (4.6)$$

$$V_{\text{ін}} = 0,5 \cdot 21000 = 10500,00 \text{ (грн)}.$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини (розділу, етапу) роботи –  $V$ :

$$V = Z_o + Z_d + H_{\text{зп}} + A + V_{\text{мат}} + V_e + V_{\text{ді}} \text{ [грн]}.$$

$$V = 21000 + 2520,00 + 5174,40 + 1683,33 + 92,00 + 211,20 + 450 + 10500 = 41630,93 \text{ (грн)}.$$

2-й етап. Розрахунок загальних витрат на виконання даної роботи. Загальна вартість всієї наукової роботи визначається за  $V_{\text{заг}}$  формулою:

$$V_{\text{заг}} = \frac{V}{\alpha} \text{ [грн]}, \quad (4.7)$$

де  $\alpha$  – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях.

Так, як над роботою задіяна одна людина, якою виконується уся робота, то  $\alpha$  становить 1. Підставивши дані у формулу, отримуємо:

$$V_{\text{заг}} = 41630,93/1 = 41630,93 \text{ (грн)}.$$

3-й етап. Прогнозування загальних витрат на виконання та впровадження результатів виконаної роботи. Прогнозування загальних витрат  $ZB$  на виконання та впровадження результатів виконаної роботи здійснюється за формулою:

$$ZB = \frac{B_{\text{заг}}}{\beta} [\text{грн}], \quad (4.8)$$

де  $\beta$  – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Так, якщо розробка знаходиться:

- на стадії науково-дослідних робіт, то  $\beta \approx 0,1$ ;
- на стадії технічного проектування, то  $\beta \approx 0,2$ ;
- на стадії розробки конструкторської документації, то  $\beta \approx 0,3$ ;
- на стадії розробки технологій, то  $\beta \approx 0,4$ ;
- на стадії розробки дослідного зразка, то  $\beta \approx 0,5$ ;
- на стадії розробки промислового зразка,  $\beta \approx 0,7$ ;
- на стадії впровадження, то  $\beta \approx 0,9$ .

Отже, підставимо дані в формулу й отримаємо результат:

$$ZB = \frac{41630,93}{0,7} = 59472,75 \text{ (грн)}.$$

#### 4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Зростання чистого прибутку забезпечить підприємству надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

Збільшення чистого прибутку підприємства  $\Delta\Pi_i$  для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta\Pi_i = \sum_i^n (\Delta\Pi_o \cdot N + C_o \cdot \Delta N) \cdot \lambda \cdot \rho \cdot \left(1 - \frac{v}{100}\right) [грн], \quad (4.9)$$

де  $\Delta\Pi_o$  – покращення основного оціночного показника від впровадження результатів розробки у даному році. Зазвичай таким показником може бути ціна одиниці нової розробки;

$N$  – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

$\Delta N$  – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$C_o$  – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

$n$  – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки;

$\lambda$  – коефіцієнт, який враховує сплату податку на додану вартість.

$\rho$  – коефіцієнт, який враховує рентабельність продукту. Приймаємо  $\rho = 0,25$ ;

$v$  – ставка податку на прибуток (18%).

В результаті впровадження результатів наукової розробки покращується якість продукту, що дозволяє підвищити ціну його реалізації на 700 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 150 шт., протягом другого року – на 350 шт., протягом третього року – на 600 шт.

Орієнтовно: реалізація продукції до впровадження результатів наукової розробки складала 1 шт., а її ціна – 2000 грн.

Спрогнозуємо збільшення чистого прибутку підприємства від впровадження результатів наукової розробки у кожному році відносно базового.

Збільшення чистого прибутку підприємства  $\Delta\Pi_1$  протягом першого року складе:

$$\Delta\Pi_1 = [1 \cdot 2000 + 150 \cdot 2700] \cdot 0,88 \cdot 0,25 \cdot \left(1 - \frac{18}{100}\right) = 73260,00 \text{ (грн)}.$$

Збільшення чистого прибутку підприємства  $\Delta\Pi_2$  протягом другого року (відносно базового року, тобто року до впровадження результатів наукової розробки) складе:

$$\Delta\Pi_2 = [1 \cdot 2000 + 350 \cdot 2700] \cdot 0,88 \cdot 0,25 \cdot \left(1 - \frac{18}{100}\right) = 170136,00 \text{ (грн)}.$$

Збільшення чистого прибутку підприємства  $\Delta\Pi_3$  протягом третього року (відносно базового року, тобто року до впровадження результатів наукової розробки) складе:

$$\Delta\Pi_3 = [1 \cdot 2000 + 600 \cdot 2700] \cdot 0,88 \cdot 0,25 \cdot \left(1 - \frac{18}{100}\right) = 291960,00 \text{ (грн)}.$$

#### 4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Основними показниками, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності. Розрахунок ефективності вкладених інвестицій передбачає проведення таких робіт:

1-й крок. Розраховується теперішня вартість інвестицій PV, що вкладаються в наукову розробку. Такою вартістю можемо вважати прогнозовану величину

загальних витрат ЗВ на виконання та впровадження результатів НДДКР, розраховану за формулою, тобто будемо вважати, що  $ZB = PV = 59472,75$ .

2-й крок. Розраховується очікуване збільшення прибутку  $\Delta\Pi_i$ , що його отримає підприємство (організація) від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження. Таке збільшення прибутку також було розраховане нами раніше.

3-й крок. Будуємо вісь часу, на яку наносимо всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів.

Платежі показуємо у ті терміни, коли вони здійснюються.

Припустимо, що загальні витрати ЗВ на виконання та впровадження результатів НДДКР (або теперішня вартість інвестицій PV) дорівнює 59472,75грн. Результати вкладених у наукову розробку інвестицій почнуть з'являтися протягом трьох років. Ці результати виявляться у тому, що у першому році підприємство отримає збільшення чистого прибутку на 73260,00 грн відносно базового року, у другому році – збільшення чистого прибутку на 170136,00 грн (відносно базового року), у третьому році – збільшення чистого прибутку на 291960 грн (відносно базового року).

Тоді рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, наведений на рис. 4.1.

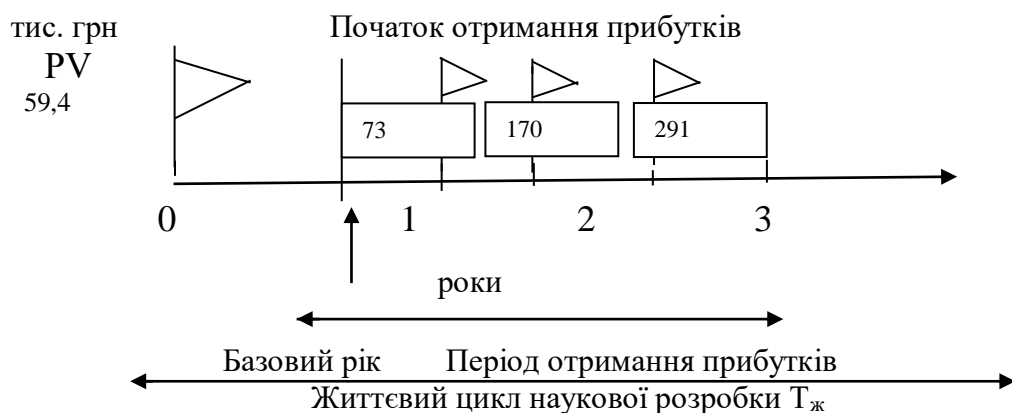


Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

4-й крок. Розраховується абсолютна ефективність вкладених інвестицій  $E_{абс}$ .

Для цього використовується формула:

$$E_{абс} = (ПП - PV), \quad (4.10)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV – теперішня вартість інвестицій  $PV = ЗВ$ , грн.

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_1^t \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.11)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні;

t – період часу (в роках) від моменту отримання чистого прибутку до точки „0”.

Якщо  $E_{абс} > 0$ , то результат від проведення наукових досліджень та їх впровадження принесе прибуток, але це також ще не свідчить про те, що інвестор буде зацікавлений у фінансуванні даного проекту (роботи).



Розрахуємо абсолютну ефективність інвестицій, вкладених у реалізацію проекту. Ставка дисконтування  $\tau$  дорівнює 0,1. Отримаємо:

$$ПП = \frac{73260}{(1 + 0,1)^1} + \frac{170136}{(1 + 0,1)^2} + \frac{291960}{(1 + 0,1)^3} = 430078,49 \text{ (грн)}.$$

Тоді,

$$E_{\text{абс}} = 430078,49 - 59472,75 = 370605,74 \text{ (грн)}.$$

Оскільки  $E_{\text{абс}} > 0$ , то вкладання коштів на виконання та впровадження результатів НДДКР є доцільним.

5-й крок. Розраховуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій  $E_{\text{в}}$ . Для цього використовуємо формулу:

$$E_{\text{в}} = T_{\text{ж}} \sqrt[1 + \frac{E_{\text{абс}}}{PV}]{1} - 1, \quad (4.12)$$

де  $E_{\text{абс}}$  – абсолютна ефективність вкладених інвестицій, грн;

$PV$  – теперішня вартість інвестицій  $PV = 3B$ , грн;

$T_{\text{ж}}$  – життєвий цикл наукової розробки, роки.

Далі, розрахована величина  $E_{\text{в}}$  порівнюється з мінімальною (бар'єрною) ставкою дисконтування  $\tau$  мін, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть.

У загальному вигляді мінімальна (бар'єрна) ставка дисконтування  $\tau$  мін визначається за формулою:

$$\tau = d + f, \quad (4.13)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках;  $d = 0,2$ ;

$f$  – показник, що характеризує ризикованість вкладень; зазвичай, величина  $f = 0,05$ .

Якщо величина  $E_b > \tau \text{ мін}$ , то інвестор може бути зацікавлений у фінансуванні даної наукової розробки. В іншому випадку фінансування наукової розробки здійснюватися не буде. Спочатку спрогнозуємо величину  $\tau \text{ мін}$ . Припустимо, що за даних умов  $\tau \text{ мін} = 0,2 + 0,05 = 0,25$ . Тоді відносна (щорічна) ефективність вкладних інвестицій в проведення наукових досліджень та впровадження їх результатів складе:

$$E_e = \sqrt[3]{1 + \frac{370605,74}{59472,75}} - 1 = \sqrt[3]{7,23} - 1 = 0,93 \text{ або } 93\%$$

Оскільки  $E_b = 93\% > \tau \text{ мін} = 0,25 = 25\%$ , то у інвестора буде зацікавленість вкладати гроші в дану наукову розробку, оскільки значно більші прибутки він отримає від того, що інвестує кошти розробку, а не розмістить гроші на депозит у комерційному банку.

6-й крок. Розраховуємо термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій  $T_{ок}$  можна розрахувати за формулою:

$$T_{ок} = \frac{1}{E_e}. \quad (4.14)$$

Термін окупності вкладених у реалізацію проекту інвестицій  $T_{ок}$  складе:

$$T_{ок} = \frac{1}{0,93} = 1,07 \text{ (року)},$$

Термін окупності менше 3-х років. Отже, фінансування даної наукової розробки є доцільним.

#### 4.5 Висновки.

В даному розділі було виконано оцінювання комерційного потенціалу розробки системи метрологічних досліджень на базі мінікомп'ютера.

Проведено технологічний аудит з залученням трьох незалежних експертів. Визначено, що рівень комерційного потенціалу розробки вище середнього.

Аналіз комерційного потенціалу розробки показав, що програмний продукт за своїми характеристиками випереджає аналогічні програмні продукти і є перспективною розробкою.

Згідно із розрахунками всіх статей витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи загальні витрати на розробку складають 59472,75грн.

Розрахована абсолютна ефективність вкладених інвестицій в сумі 370605,74 свідчить про отримання прибутку інвестором від комерціалізації програмного продукту.

Щорічна ефективність вкладених в наукову розробку інвестицій складає 93%, що вище за мінімальну бар'єрну ставку дисконтування, яка складає 25%. Це означає потенційну зацікавленість інвесторів у фінансуванні розробки.

Термін окупності вкладених у реалізацію проекту інвестицій становить 1,07 року, що також свідчить про доцільність фінансування нової розробки.

## ВИСНОВКИ

В магістерській кваліфікаційній роботі було розглянуто існуючі підходи до побудови систем метеорологічних досліджень, а саме огляд існуючих видів метеостанцій, приладів для отримання метеорологічних досліджень. Було класифіковано та оглянуто види мінікомп'ютерів. Це все дало можливість виділити основні вимоги до розробки.

Був проведений вибір мінікомп'ютера та електронних компонентів, що відповідають поставленій задачі. На основі цього створено структурно – функціональну схему системи метеорологічних досліджень та електричну принципову.

Розроблено алгоритм функціонування системи та виконано налаштування програмного забезпечення. Створено необхідні скрипти для отримання, обробки та конвертації даних з сенсорів в HTML файл. Створено алгоритми роботи з іншими механізмами на основі даних отриманих з датчиків за допомогою інтерфейсу 1-Wire. Запущено сервер для перегляду необхідної інформації через мережу інтернет.

Було проведено аналіз ринку та оцінювання економічного потенціалу розробки. Було розраховано термін окупності інвестицій вкладених в розробку даного продукту та обґрунтовано доцільність її фінансування.

Отже всі завдання необхідні для мети роботи виконано.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Гушля А. В., Мезенцев В. С. Водно - балансовые исследования. – Киев : Высшая школа. Головное изд - во, 1982. – 229 с.
2. Колесник П. И. Метеорология. Практикум. – Киев : Высшая школа. Головное изд - во, 1986. – 175 с.
3. Метрологическое обеспечение измерительных информационных систем (теория, методология, организация)/Е.Т. Удовиченко, А.А.Брагин, А.Л.Семенюк, В.И.Бородатый, Э.С. Браилов, Ю.И. Койфман, А.Д. Пинчевский. – М.: Изд-во стандартов, 1991.-192 с.
4. Вострокнутов Н.Н. Цифровые измерительные устройства. Теория погрешностей, испытаний, поверка. –М.: Энергоатомиздат, 1990.-208 с.
5. Методические указания по экспериментальному определению основных элементов водного и теплового баланса / Под ред. Сорокина В. Г. – Ровно : 1977, - 74 с.
6. Моргунов В. К. Основы метеорологии, климатологии. Метеорологические приборы и методы наблюдений : Учебник. – Ростов / Д.: Феникс. – Новосибирск : Сибирское соглашение, 2005. – 331 с.
7. Наставление гидрометеорологическим станциям и постам. Вып. 3, часть I. Метеорологические наблюдения на станциях. – Л.: Гидрометеоздат, 1985. – 300 с.
8. Збыточные системы счисления, моделирование, обработка данных и системное проектирование в технике преобразования информации: Учеб. пособие / В.А. Поджаренко, А.Д. Азаров, В.А. Власенко, И.И. Коваленко.: - Выща шк., 1990. – 208 с.
9. Проектирование микропроцесорных измерительных приборов и систем/В.Д. Циделко, Н.В. Нагаец, Ю.В. Хохлов и др.- К.: Техніка, 1984.-215 с.
10. Digital Signal Processing: World Class Designs / [David Katz, Kenton Williston, Walt Kester and other]. – Newnes, 2009. – 460 с.

11. Психрометрические таблицы. – Л.: Гидрометеиздат, 1974. – 232 с.
12. Чеботарев А. И., Клибашев К.П. Гидрологические расчеты. – Л.: Гидрометеиздат, 1956. – 296 с.
13. Метрологическое обеспечение измерительных информационных систем./Е.Т. Удовиченко, А.Л.Семенюк, В.И.Бородатый, Э.С. Браилов, Ю.И. Койфман, А.Д. Пинчевский. – М.: Изд-во стандартов, 1993.-176 с.
14. Raspberry Pi. Getting Started Guide [ Электронный ресурс ]. Режим доступа : [http://d4c027c89b30561298bd-484902fe60e1615dc83faa972a248000.r12.cf3.rackcdn.com/supporting\\_materials/Raspberry%20Pi%20Start%20Guide.pdf](http://d4c027c89b30561298bd-484902fe60e1615dc83faa972a248000.r12.cf3.rackcdn.com/supporting_materials/Raspberry%20Pi%20Start%20Guide.pdf)
15. DS18B20 Programmable Resolution 1-Wire Digital Thermometer [Электронный ресурс]. Режим доступа: <http://datasheets.maxim-ic.com/en/ds/DS18B20.pdf>
16. BME280 data sheet [Электронный ресурс] / Bosch Sensortec, 2014. – Режим доступа: <https://www.embeddedadventures.com/datasheets/BME280.pdf>
17. Datasheet SHT1x (SHT10, SHT11, SHT15) Humidity and Temperature Sensor IC [ Электронный ресурс ]. Режим доступа : [http://www.sensirion.com/fileadmin/user\\_upload/customers/sensirion/Dokumente/Humidity/Sensirion\\_Humidity\\_SHT1x\\_Datasheet\\_V5.pdf](http://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/Humidity/Sensirion_Humidity_SHT1x_Datasheet_V5.pdf)
18. MPX4115 Series datasheet [Электронный ресурс]. Режим доступа: <http://www.datasheetcatalog.org/datasheet/Motorola/MPX4115.pdf>.
19. TC3400 [Электронный ресурс]. Режим доступа: [www.gaw.ru/pdf/Microchip/adc/tc3400.pdf](http://www.gaw.ru/pdf/Microchip/adc/tc3400.pdf).
20. C library for Broadcom BCM 2835 as used in Raspberry Pi [ Электронный ресурс ]. Режим доступа: [www.open.com.au/mikem/bcm2835/index.html](http://www.open.com.au/mikem/bcm2835/index.html).
21. Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт / Уклад. В. О. Козловський – Вінниця: ВНТУ, 2012. – 22 с.

## ДОДАТОК А

Міністерство освіти та науки України  
Вінницький національний технічний університет  
Інститут інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТд.т.н., професор Мартинюк Т. Б.

(наук. ст., вч. зв., ініц. та прізви.)

(підпис)

“ \_\_\_\_ ” \_\_\_\_\_ 20\_\_р.

**ТЕХНІЧНЕ ЗАВДАННЯ**

на виконання магістерської кваліфікаційної роботи

Система метеорологічних досліджень на базі мінікомп'ютера08-23.МКР.012.00.000 ПЗНауковий керівник: к.т.н., доц.\_\_\_\_\_ Богомолів С.В.

(підпис)

Магістрант групи \_1КІ-18м\_\_\_\_\_ Осипчук С.В.

(підпис)

Вінниця 2019р.

## 1. Підстава для виконання магістерської кваліфікаційної роботи (МКР)

- а) актуальність досліджень;
- б) наказ про затвердження теми дипломної роботи.

## 2. Мета і призначення МКР

а) мета – забезпечення опрацювання сигналів давачів з різнотипними інтерфейсами і застосуванням сучасних мінікомп'ютерів ;

б) призначення розробки – можливість обробки даних різнотипних датчиків, їх систематизація, вплив отриманих даних на інші механізми системи та створення локального серверу для відображення роботи пристрою.

## 3. Вихідні дані для виконання МКР

- напруга живлення 5В;
- потужність живлення 5Вт;
- запуск власного сервера;
- виведення результатів за допомогою HTML сторінки та на дисплей;
- похибка вимірюваних значень не більше 0.5%.

## 4. Вимоги до виконання МКР

- провести огляд існуючих систем метеорологічних досліджень та сформулювати основні ідеї створення власної;
- здійснити дослідження та сформулювати переваги створення метеостанції на базі мінікомп'ютера;
- розглянути основні мінікомп'ютери на ринку та обґрунтувати вибір для даної системи;
- здійснити вибір додаткових елементів та побудувати схему електричну принципову ;
- розробити алгоритми та програмне забезпечення (ПЗ).



## 5. Етапи МКР та очікувані результати

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Пошук та огляд інформаційних джерел	05.09.19р.	19.09.19р.	Розділ 1
2	Дослідження способів побудови мікропроцесорних систем навігації	20.09.19р.	09.10.19р.	Розділ 2
3	Побудова мікропроцесорного GPS маячка з автономним живленням	10.10.19р.	24.10.19р.	Розділ 3
4	Економічна частина	25.10.19р.	12.11.19р.	Розділ 4, пояснювальна записка

## 6. Матеріали, що подаються до захисту МКР

Пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відзив наукового керівника, відзив опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, нормоконтроль про відповідність оформлення МКР діючим вимогам.

## 7. Порядок контролю виконання та захисту МКР

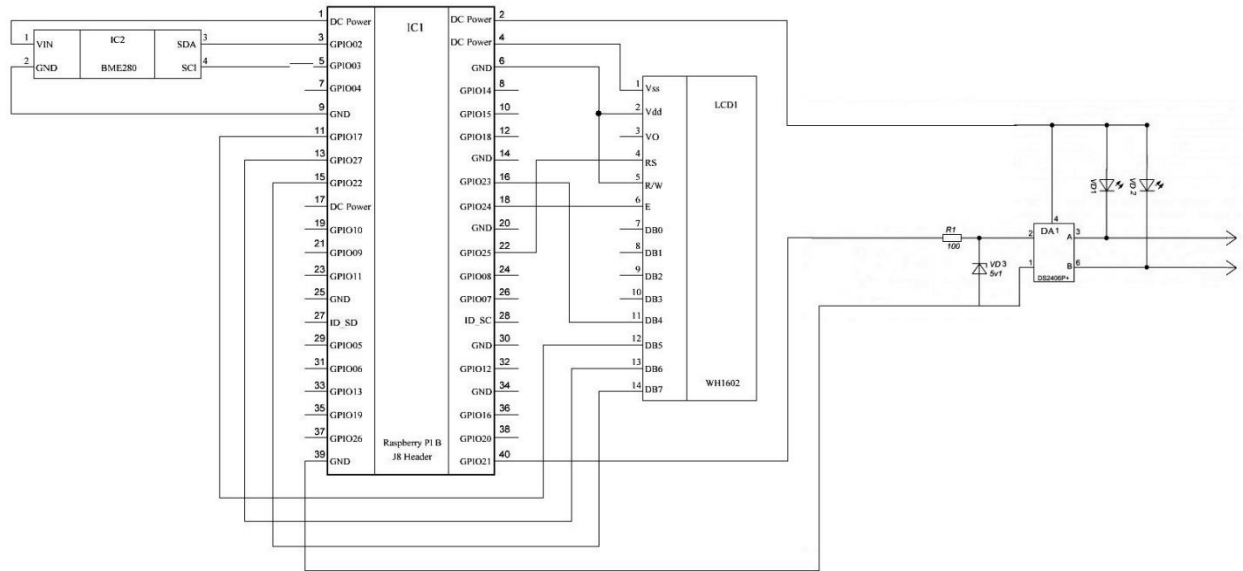
Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

## 8. Вимоги до оформлення МКР

Вимоги викладені в МЕТОДИЧНИХ ВКАЗІВКАХ до дипломного проектування, ДСТУ\_ 3008-95, ДСТУ 3974-2000 «Правила виконання дослідно-конструкторських робіт. Загальні положення» та діючого ГОСТ 2.114-95 ЕСКД.

9. Вимоги щодо технічного захисту інформації в МКР з обмеженим доступом

Відсутні.



					<b>08-23.MKP.012.000E3</b>			
<b>Змн.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дат</b>				
Розроб.	Осипчук С.В.				<b>Система метеорологічних досліджень</b> Схема електрична принципова	<b>Літ.</b>	<b>Арк.</b>	<b>Аркушів</b>
Перевір.	Богомолов С.В.							
Реценз.	Поплавський А.В.							
Н. Контр.	Шець С.І.							
Затверд.	Мартинюк Т.Б.							
						<b>ВНТУ, 1КІ-18М</b>		



## ДОДАТОК Г - Лістинг бібліотеки роботи з LCD дисплеєм

```
#!/usr/bin/python

#
# based on code from lrvick and LiquidCrystal
# lrvic - https://github.com/lrvick/raspi-hd44780/blob/master/hd44780.py
# LiquidCrystal -
https://github.com/arduino/Arduino/blob/master/libraries/LiquidCrystal/LiquidCrystal.cpp
#

from time import sleep

class Adafruit_CharLCD(object):

    # commands
    LCD_CLEARDISPLAY      = 0x01
    LCD_RETURNHOME       = 0x02
    LCD_ENTRYMODESET     = 0x04
    LCD_DISPLAYCONTROL   = 0x08
    LCD_CURSORSHIFT      = 0x10
    LCD_FUNCTIONSET      = 0x20
    LCD_SETCGRAMADDR     = 0x40
    LCD_SETDDRAMADDR     = 0x80

    # flags for display entry mode
    LCD_ENTRYRIGHT       = 0x00
    LCD_ENTRYLEFT        = 0x02
    LCD_ENTRYSHIFTINCREMENT = 0x01
    LCD_ENTRYSHIFTDECREMENT = 0x00

    # flags for display on/off control
    LCD_DISPLAYON        = 0x04
    LCD_DISPLAYOFF       = 0x00
    LCD_CURSORON         = 0x02
    LCD_CURSOROFF        = 0x00
    LCD_BLINKON          = 0x01
    LCD_BLINKOFF         = 0x00

    # flags for display/cursor shift
```

```
LCD_DISPLAYMOVE      = 0x08
LCD_CURSORMOVE       = 0x00
```

```
# flags for display/cursor shift
```

```
LCD_DISPLAYMOVE      = 0x08
LCD_CURSORMOVE       = 0x00
LCD_MOVERIGHT        = 0x04
LCD_MOVELEFT         = 0x00
```

```
# flags for function set
```

```
LCD_8BITMODE         = 0x10
LCD_4BITMODE         = 0x00
LCD_2LINE            = 0x08
LCD_1LINE            = 0x00
LCD_5x10DOTS         = 0x04
LCD_5x8DOTS          = 0x00
```

```
def __init__(self, pin_rs=25, pin_e=24, pins_db=[23, 17, 27, 22], GPIO=None):
    # Emulate the old behavior of using RPi.GPIO if we haven't been given
    # an explicit GPIO interface to use
    if not GPIO:
        import RPi.GPIO as GPIO
        GPIO.setwarnings(False)
    self.GPIO = GPIO
    self.pin_rs = pin_rs
    self.pin_e = pin_e
    self.pins_db = pins_db

    self.GPIO.setmode(GPIO.BCM)
    self.GPIO.setup(self.pin_e, GPIO.OUT)
    self.GPIO.setup(self.pin_rs, GPIO.OUT)

    for pin in self.pins_db:
        self.GPIO.setup(pin, GPIO.OUT)

    self.write4bits(0x33) # initialization
    self.write4bits(0x32) # initialization
    self.write4bits(0x28) # 2 line 5x7 matrix
    self.write4bits(0x0C) # turn cursor off 0x0E to enable cursor
    self.write4bits(0x06) # shift cursor right

    self.displaycontrol = self.LCD_DISPLAYON | self.LCD_CURSOROFF |
self.LCD_BLINKOFF

    self.displayfunction = self.LCD_4BITMODE | self.LCD_1LINE |
self.LCD_5x8DOTS
    self.displayfunction |= self.LCD_2LINE
```

```

    # Initialize to default text direction (for romance languages)
    self.displaymode = self.LCD_ENTRYLEFT |
self.LCD_ENTRYSHIFTDECREMENT
    self.write4bits(self.LCD_ENTRYMODESET | self.displaymode) # set the entry
mode

    self.clear()

def begin(self, cols, lines):
    if (lines > 1):
        self.numlines = lines
        self.displayfunction |= self.LCD_2LINE

def home(self):
    self.write4bits(self.LCD_RETURNHOME) # set cursor position to zero
    self.delayMicroseconds(3000) # this command takes a long time!

def clear(self):
    self.write4bits(self.LCD_CLEARDISPLAY) # command to clear display
    self.delayMicroseconds(3000) # 3000 microsecond sleep, clearing the display
takes a long time

def setCursor(self, col, row):
    self.row_offsets = [0x00, 0x40, 0x14, 0x54]
    if row > self.numlines:
        row = self.numlines - 1 # we count rows starting w/0
    self.write4bits(self.LCD_SETDRAMADDR | (col + self.row_offsets[row]))

def noDisplay(self):
    """ Turn the display off (quickly) """
    self.displaycontrol &= ~self.LCD_DISPLAYON
    self.write4bits(self.LCD_DISPLAYCONTROL | self.displaycontrol)

def display(self):
    """ Turn the display on (quickly) """
    self.displaycontrol |= self.LCD_DISPLAYON
    self.write4bits(self.LCD_DISPLAYCONTROL | self.displaycontrol)

def noCursor(self):
    """ Turns the underline cursor off """
    self.displaycontrol &= ~self.LCD_CURSORON
    self.write4bits(self.LCD_DISPLAYCONTROL | self.displaycontrol)

def cursor(self):
    """ Turns the underline cursor on """
    self.displaycontrol |= self.LCD_CURSORON

```

```

self.write4bits(self.LCD_DISPLAYCONTROL | self.displaycontrol)

def noBlink(self):
    """ Turn the blinking cursor off """
    self.displaycontrol &= ~self.LCD_BLINKON
    self.write4bits(self.LCD_DISPLAYCONTROL | self.displaycontrol)

def blink(self):
    """ Turn the blinking cursor on """
    self.displaycontrol |= self.LCD_BLINKON
    self.write4bits(self.LCD_DISPLAYCONTROL | self.displaycontrol)

def DisplayLeft(self):
    """ These commands scroll the display without changing the RAM """
    self.write4bits(self.LCD_CURSORSHIFT | self.LCD_DISPLAYMOVE |
self.LCD_MOVELEFT)

def scrollDisplayRight(self):
    """ These commands scroll the display without changing the RAM """
    self.write4bits(self.LCD_CURSORSHIFT | self.LCD_DISPLAYMOVE |
self.LCD_MOVERIGHT)

def leftToRight(self):
    """ This is for text that flows Left to Right """
    self.displaymode |= self.LCD_ENTRYLEFT
    self.write4bits(self.LCD_ENTRYMODESET | self.displaymode)

def rightToLeft(self):
    """ This is for text that flows Right to Left """
    self.displaymode &= ~self.LCD_ENTRYLEFT
    self.write4bits(self.LCD_ENTRYMODESET | self.displaymode)

def autoscroll(self):
    """ This will 'right justify' text from the cursor """
    self.displaymode |= self.LCD_ENTRYSHIFTINCREMENT
    self.write4bits(self.LCD_ENTRYMODESET | self.displaymode)

def noAutoscroll(self):
    """ This will 'left justify' text from the cursor """
    self.displaymode &= ~self.LCD_ENTRYSHIFTINCREMENT
    self.write4bits(self.LCD_ENTRYMODESET | self.displaymode)

def write4bits(self, bits, char_mode=False):
    """ Send command to LCD """
    self.delayMicroseconds(1000) # 1000 microsecond sleep
    bits = bin(bits)[2:].zfill(8)
    self.GPIO.output(self.pin_rs, char_mode)

```

```

for pin in self.pins_db:
    self.GPIO.output(pin, False)
for i in range(4):
    if bits[i] == "1":
        self.GPIO.output(self.pins_db[::-1][i], True)
self.pulseEnable()
for pin in self.pins_db:
    self.GPIO.output(pin, False)
for i in range(4, 8):
    if bits[i] == "1":
        self.GPIO.output(self.pins_db[::-1][i-4], True)
self.pulseEnable()

def delayMicroseconds(self, microseconds):
    seconds = microseconds / float(1000000) # divide microseconds by 1 million
for seconds
    sleep(seconds)

def pulseEnable(self):
    self.GPIO.output(self.pin_e, False)
    self.delayMicroseconds(1)    # 1 microsecond pause - enable pulse must be >
450ns
    self.GPIO.output(self.pin_e, True)
    self.delayMicroseconds(1)    # 1 microsecond pause - enable pulse must be >
450ns
    self.GPIO.output(self.pin_e, False)
    self.delayMicroseconds(1)    # commands need > 37us to settle

def message(self, text):
    """ Send string to LCD. Newline wraps to second line"""
    for char in text:
        if char == '\n':
            self.write4bits(0xC0) # next line
        else:
            self.write4bits(ord(char), True)

if __name__ == '__main__':
    lcd = Adafruit_CharLCD()
    lcd.clear()
    lcd.message(" Adafruit 16x2\n Standard LCD")

```



## ДОДАТОК Д - Лістинг бібліотеки роботи з датчиком BME280

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import smbus
import time

# BME280 default address.
BME280_I2CADDR = 0x76

# BME280 Registers
BME280_CONTROL_MEAS = 0xF4
BME280_CONTROL_HUM = 0xF2

BME280_CONFIG = 0xF5
BME280_PRESSURE = 0xF7
BME280_TEMP = 0xFA
BME280_HUMIDITY = 0xFD

BME280_DIG_T1 = 0x88
BME280_DIG_T2 = 0x8A
BME280_DIG_T3 = 0x8C
BME280_DIG_P1 = 0x8E
BME280_DIG_P2 = 0x90
BME280_DIG_P3 = 0x92
BME280_DIG_P4 = 0x94
BME280_DIG_P5 = 0x96
BME280_DIG_P6 = 0x98
BME280_DIG_P7 = 0x9A
BME280_DIG_P8 = 0x9C
BME280_DIG_P9 = 0x9E
BME280_DIG_H1 = 0xA1
BME280_DIG_H2 = 0xE1
BME280_DIG_H3 = 0xE3
BME280_DIG_H4 = 0xE4
BME280_DIG_H5 = 0xE5
BME280_DIG_H6 = 0xE7

# Oversampling Setting
BME280_OVERS_T1 = 0x20
BME280_OVERS_T2 = 0x40
```

BME280\_OVERS\_T4 = 0x60  
BME280\_OVERS\_T8 = 0x80  
BME280\_OVERS\_T16 = 0xA0

BME280\_OVERS\_P1 = 0x04  
BME280\_OVERS\_P2 = 0x08  
BME280\_OVERS\_P4 = 0x0C  
BME280\_OVERS\_P8 = 0x10  
BME280\_OVERS\_P16 = 0x14

BME280\_OVERS\_H1 = 0x01  
BME280\_OVERS\_H2 = 0x02  
BME280\_OVERS\_H4 = 0x03  
BME280\_OVERS\_H8 = 0x04  
BME280\_OVERS\_H16 = 0x05

# Power Modes.

# This lib uses NORMAL mode only!

#BME280\_SLEEP\_MODE = 0x00  
#BME280\_FORCED\_MODE = 0x01  
BME280\_NORMAL\_MODE = 0x03

BME280\_TSB\_0\_5 = 0x00  
BME280\_TSB\_62\_5 = 0x20  
BME280\_TSB\_125 = 0x40  
BME280\_TSB\_250 = 0x60  
BME280\_TSB\_500 = 0x80  
BME280\_TSB\_1000 = 0xA0  
BME280\_TSB\_2000 = 0xC0  
BME280\_TSB\_4000 = 0xE0

BME280\_FILTER\_OFF = 0x00  
BME280\_FILTER\_COEFFICIENT2 = 0x04  
BME280\_FILTER\_COEFFICIENT4 = 0x08  
BME280\_FILTER\_COEFFICIENT8 = 0x0C  
BME280\_FILTER\_COEFFICIENT16 = 0x10

BME280\_SPI\_OFF = 0x00  
BME280\_SPI\_ON = 0x01

BME280\_CONTROL\_MEAS\_SET = (BME280\_OVERS\_T16 |  
BME280\_OVERS\_P16 | BME280\_NORMAL\_MODE)  
BME280\_CONTROL\_HUM\_SET = BME280\_OVERS\_H2  
BME280\_CONFIG\_SET = (BME280\_TSB\_0\_5 |  
BME280\_FILTER\_COEFFICIENT16 | BME280\_SPI\_OFF)

class BME280(object):

```

def __init__(self, port=1, address=BME280_I2CADDR):
    self.bus = smbus.SMBus(port)
    self.address = address

    # Read calibration values
    self.dig_t1 = self.read_word(BME280_DIG_T1) # Unsigned
    self.dig_t2 = self.read_word_sign(BME280_DIG_T2)
    self.dig_t3 = self.read_word_sign(BME280_DIG_T3)
    self.dig_p1 = self.read_word(BME280_DIG_P1) # Unsigned
    self.dig_p2 = self.read_word_sign(BME280_DIG_P2)
    self.dig_p3 = self.read_word_sign(BME280_DIG_P3)
    self.dig_p4 = self.read_word_sign(BME280_DIG_P4)
    self.dig_p5 = self.read_word_sign(BME280_DIG_P5)
    self.dig_p6 = self.read_word_sign(BME280_DIG_P6)
    self.dig_p7 = self.read_word_sign(BME280_DIG_P7)
    self.dig_p8 = self.read_word_sign(BME280_DIG_P8)
    self.dig_p9 = self.read_word_sign(BME280_DIG_P9)

    self.dig_h1 = self.read_byte(BME280_DIG_H1) # unsigned char
    self.dig_h2 = self.read_word_sign(BME280_DIG_H2)
    self.dig_h3 = self.read_byte(BME280_DIG_H3) # unsigned char

    self.dig_h4 = (self.read_byte(BME280_DIG_H4) << 24) >> 20
    self.dig_h4 = self.dig_h4 | self.read_byte(BME280_DIG_H4+1) & 0x0F

    self.dig_h5 = (self.read_byte(BME280_DIG_H5+1) << 24) >> 20
    self.dig_h5 = self.dig_h5 | (self.read_byte(BME280_DIG_H5) >> 4) &
0x0F

    self.dig_h6 = self.read_byte(BME280_DIG_H6) # signed char
    if self.dig_h6 > 127:
        self.dig_h6 = 127-self.dig_h6

    # Set Configuration
    self.write_byte(BME280_CONFIG, BME280_CONFIG_SET)
    self.write_byte(BME280_CONTROL_HUM,
BME280_CONTROL_HUM_SET)
    self.write_byte(BME280_CONTROL_MEAS,
BME280_CONTROL_MEAS_SET)

def get_data(self):
    adc_t = self.read_adc_long(BME280_TEMP)
    adc_p = self.read_adc_long(BME280_PRESSURE)
    adc_h = self.read_adc_word(BME280_HUMIDITY)

    var1 = (adc_t/16384.0 - self.dig_t1/1024.0) * self.dig_t2;

```

```

        var2 = ((adc_t/131072.0 - self.dig_t1/8192.0) * (adc_t/131072.0 -
self.dig_t1/8192.0)) * self.dig_t3;
        t_fine = (var1 + var2);
        temperature = round((t_fine / 5120.0), 2);

        var1 = (t_fine/2.0) - 64000.0;
        var2 = var1 * var1 * self.dig_p6 / 32768.0;
        var2 = var2 + var1 * self.dig_p5 * 2.0;
        var2 = (var2/4.0)+(self.dig_p4 * 65536.0);
        var1 = (self.dig_p3 * var1 * var1 / 524288.0 + self.dig_p2 * var1) /
524288.0;
        var1 = (1.0 + var1 / 32768.0)*self.dig_p1;

        # Avoid exception caused by division by zero
        if (var1 == 0.0):
            return -1

        p = 1048576.0 - adc_p;
        p = (p - (var2 / 4096.0)) * 6250.0 / var1;
        var1 = self.dig_p9 * p * p / 2147483648.0;
        var2 = p * self.dig_p8 / 32768.0;
        pressure = round((p + (var1 + var2 + self.dig_p7) / 16.0), 2);

        var_H = t_fine - 76800.0
        var_H = (adc_h-(self.dig_h4*64.0+self.dig_h5 / 16384.0 *
var_H))*(self.dig_h2/65536.0 * (1.0 + self.dig_h6/67108864.0 * var_H *
(1.0+self.dig_h3/67108864.0 * var_H)))
        humidity = round(var_H * (1.0 - self.dig_h1*var_H/524288.0), 2)
        if (humidity > 100.0):
            humidity = 100.0
        else:
            if(humidity < 0.0):
                humidity = 0.0

        return {'t':temperature, 'p':pressure, 'h':humidity}

def get_altitude(self, pressure):
    temp = pressure/101325;
    temp = 1-pow(temp, 0.19029);
    altitude = round(44330*temp, 3);
    return altitude;

def read_byte(self, adr):
    return self.bus.read_byte_data(self.address, adr)

def read_word(self, adr):

```

```
# ATANTION! Joke from Bosch! LBS before HBS. For calibration  
registers only!
```

```
lbs = self.bus.read_byte_data(self.address, adr)  
hbs = self.bus.read_byte_data(self.address, adr+1)  
return (hbs << 8) + lbs
```

```
def read_word_sign(self, adr):  
    val = self.read_word(adr)  
    if (val >= 0x8000):  
        return -((65535 - val) + 1)  
    else:  
        return val
```

```
def read_adc_long(self, adr):  
    mbs = self.bus.read_byte_data(self.address, adr)  
    lbs = self.bus.read_byte_data(self.address, adr+1)  
    xbs = self.bus.read_byte_data(self.address, adr+2)  
    val = (mbs << 16) + (lbs << 8) + xbs  
    val = (val >> 4)  
    return val
```

```
def read_adc_word(self, adr):  
    mbs = self.bus.read_byte_data(self.address, adr)  
    lbs = self.bus.read_byte_data(self.address, adr+1)  
    val = (mbs << 8) + lbs  
    return val
```

```
def write_byte(self, adr, byte):  
    self.bus.write_byte_data(self.address, adr, byte)
```

## ДОДАТОК Е - Робота сайту з обробленими даними



Рисунок Е.1 – Повна версія сайту



Temperature: 23.8 °C  
Pressure: 752.26 mm Hg  
Humidity: 26.07 %

## Last 24 hours

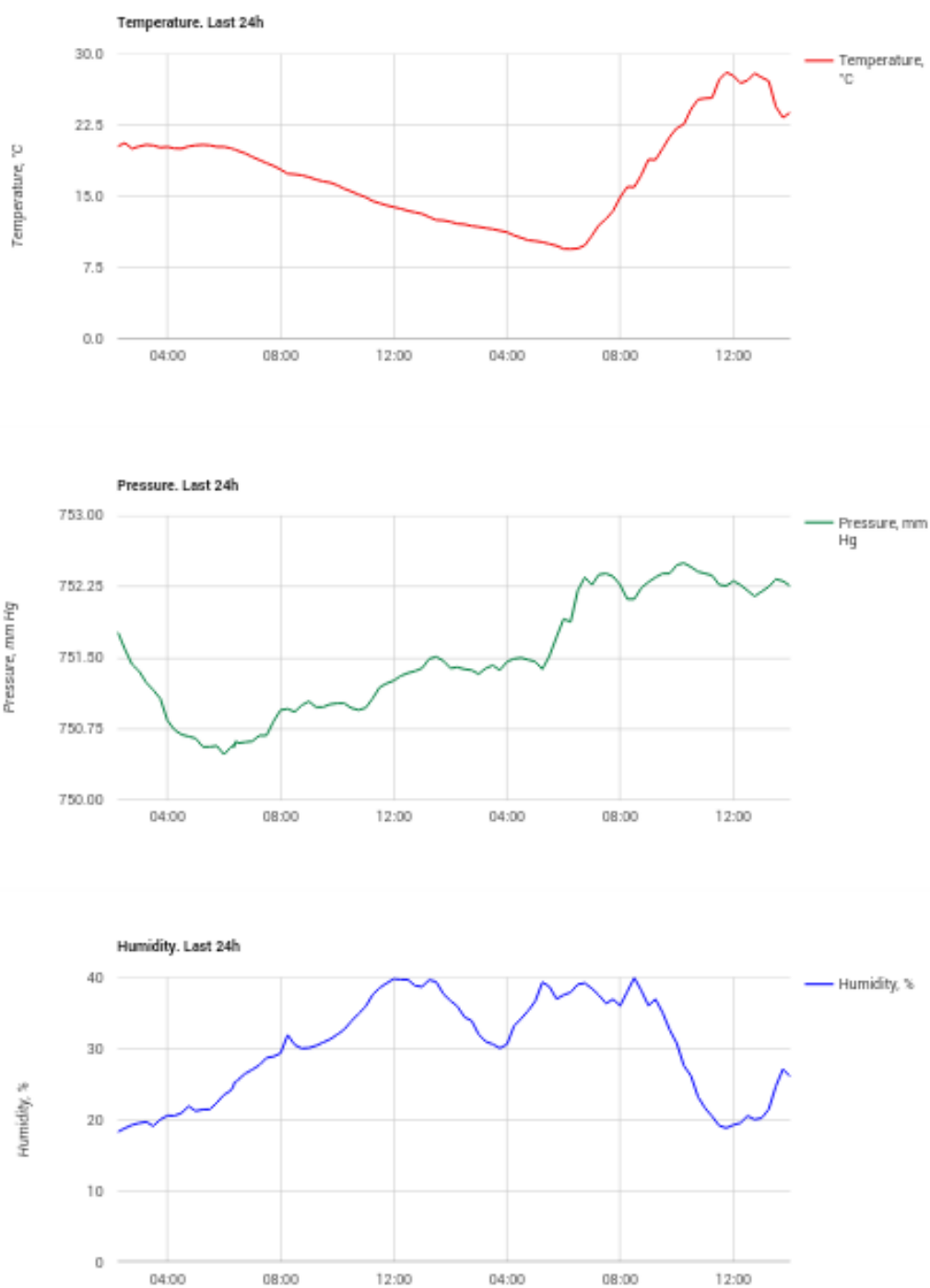


Рисунок Е.2 – Мобільна версія сайту

## ДОДАТОК Ж - Алгоритм роботи програмного забезпечення

