

Факультет комп'ютерних  
наук та адміністрування  
Кафедра  
автоматизованих систем  
моніторингу навколишнього  
середовища і інформатики

## **Магістерська кваліфікаційна робота**

на тему: **Розробка датчика тиску на базі процесора ANDROID**

Виконав студент 6 курсу  
Коломеєць Богдан Сергійович

Керівник д. ф.-м.н., проф.  
Ковальчук Володимир Володимирович

Рецензент д.т.н, проф.  
Браїлов Олександр Юрійович

Одеса 2022

№ п/п	ЗМІСТ	стор
	Вступ	3
1.	ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНІ СИСТЕМИ ТА КОЛО ЗАДАЧ, ЩО ВОНИ ВИРІШУЮТЬ	5
1.1.	Задачі, що вирішують інформаційно-вимірювальні системи	5
1.2.	Забезпечення метрологічної сумісності об'єкта вимірювання з процесором	11
2.	ВИМІРЮВАЛЬНА СИСТЕМА ДЛЯ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОГО СЕНСОРА	17
2.1	Чутливий інтелектуальний сенсор на основі метеоропереходу	17
2.2	Вимірювання товщини чутливого елемента інтелектуального сенсора методом кварцевого резонатора	23
3.	АПАРАТНО ОБЧИСЛЮВАЛЬНА ПЛАТФОРМА	29
3.1	Платформи для розробки проекту	29
3.2	Апаратні та програмні характеристики датчика	34
4.	РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ	40
4.1	Обґрунтування вибору плати для проекту	40
4.2	Структура та методи організації вхідних та вихідних даних	42
4.3	Плата призначена для управління пристроями	50
4.4.	Робота з використанням платформи ANDROID	55
	Висновки	60
	Додаток	62
	Література	69

## ВСТУП

**Актуальність.** Сьогодні проведення науково-технічних досліджень у напрямку отримання, обробки та передачі прикладної інформації набуває надзвичайно актуального характеру [1-5]. Реалізація стартапів подібного типу сприяє більш глибокому осмисленню природних явищ, розробці коректних методів та ефективних пристроїв щодо проведення наукових досліджень, взагалі, та метеорологічних вимірювань фізико-хімічних характеристик навколишнього середовища зокрема [6-10]. Нові розробки сприяють розширенню спектру вирішених практичних задач, що пов'язані з реалізацією досить сучасного напрямку обчислювальної техніки – створення компонент інформаційно-вимірювальних систем (ІВС) на основі нових сучасних технологій [11]. Серед важливих завдань слід виокремити необхідність розробки і використання таких апаратно обчислювальних платформ [12-16], що доводять свою високу ефективність у багатьох сферах сучасної вимірювальної інженерії, зокрема у метеорології [17-27].

**Метою** дипломного проекту є розробка пристрою на базі плати ARDUINO (для гаджетів, що мають ANDROID процесор). Це дасть змогу аналізувати данні тиску, температури та вологості повітря в реальному часі.

**Предметом** дослідження є складові датчика (зокрема, датчика тиску) та апаратно обчислювальна платформа на базі плати ARDUINO та сфера її застосування з використанням гаджетів на процесорі ANDROID.

**Об'єкт** дослідження склав процес отримання та обробки метеорологічної інформації (зокрема, за допомогою датчика тиску) та передача її на гаджети з процесором ANDROID.

**Задачі** роботи полягали у наступному.

1. Провести аналітичний огляд та розглянути коло задач, що вирішують сучасні інформаційно-вимірювальні системи.

2. Провести розробку чутливого елемента інтелектуального сенсора для створення датчика тиску та інших метеорологічних характеристик.
3. Проаналізувати можливості апаратно обчислювальної платформи ARDUINO та визначити сфери її застосування для гаджетів з ANDROID-процесором.
4. Виокремити основні компоненти апаратно обчислювальної платформи ARDUINO та створити програмне середовище для гаджетів з ANDROID-процесором.
5. Довести до практичної реалізації пристрій, що дозволяє виміряти метеорологічні характеристики (тиск, температура, вологість повітря). Зробити його зручним у використанні для гаджетів з ANDROID-процесором, завдяки гнучкій системи налаштування (що дозволить змінювати проміжок часу між даними).

**Практичне** значення результатів дослідження – результати аналізу обмежень та погрішностей вимірювання метеорологічних характеристик (тиску, температури, вологості повітря) на основі схеми кодування з використанням апаратно-обчислювальної платформи ARDUINO для гаджетів з ANDROID-процесором.

**Апробація** результатів кваліфікаційної роботи проведена у ВСП «Одеський фаховий коледж комп'ютерних технологій Одеського державного екологічного університету» під час виконання дипломної практики на навчальних заняттях для студентів, що навчаються за спеціальністю 121 «Інженерія програмного забезпечення».

За результати дослідження опубліковано тези на X Міжнародній науково-практичній конференції «Study of world opinion regarding the development of science», 22-25 листопада 2022 р. (Прага, Чехія). Отримано сертифікат учасника.

# **1. ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНІ СИСТЕМИ ТА КОЛО ЗАДАЧ, ЩО ВОНИ ВИРІШУЮТЬ**

## **1.1. Задачі, що вирішують інформаційно-вимірювальні системи**

Сучасна інформаційно-вимірювальна система (ІВС), як правило, орієнтована на вирішення двох нагальних проблем: вимірювання будь-якого параметру, або характеристики (фізико-хімічної, біологічної, поведінкової, психологічної, соціальної тощо) та інформування про її величину виконавцю, споживачу, досліднику (далі – суб'єкту) [1,2].

Головними практичними задачами ІВС є: розширення функціональних можливостей суб'єкта у процесі збору та обробки інформації, її адаптування до конкретно поставленої задачі, її розподілу шляхом використання програмного забезпечення (soft) в процесі створених масивів вимірювальної інформації, керування складними автоматичними комплексами і системами; підвищення точності і якості вимірювань, за рахунок усереднення і статистичної обробки вимірюваних даних з урахуванням впливу середовища.

Обчислювальні засоби, що використовуються для вимірювань повинні виконувати наступні функції [3]:

- фільтрація, знаходження і усунення відхилень сигналів від заданого рівня;
- внесення поправок;
- облік впливу зовнішніх факторів;
- обчислення результатів непрямих, сукупних і сумісних вимірювань;
- визначення статистичних характеристик вимірюваних величин;
- оцінка достовірних результатів вимірювань;
- накопичення і зберігання отриманої інформації;
- керування блоками з метою організації запитів, пріоритетів, ділового режиму, адресації до архіву і пам'яті;

- контроль робочої здатності блоків;
- самоперевірка і визначення метеорологічних характеристик.

Відповідно до назви системи розглянемо з чого складається інформаційна її частина. Якщо до основної моделі розподілу інформаційних процесів додати принцип, завдяки якому кожній інформації відповідає самостійна функціональна властивість і технічні засоби реалізації, то перелік інформаційних процесів можна подати у вигляді схеми, що наведена на рисунку 1.1.

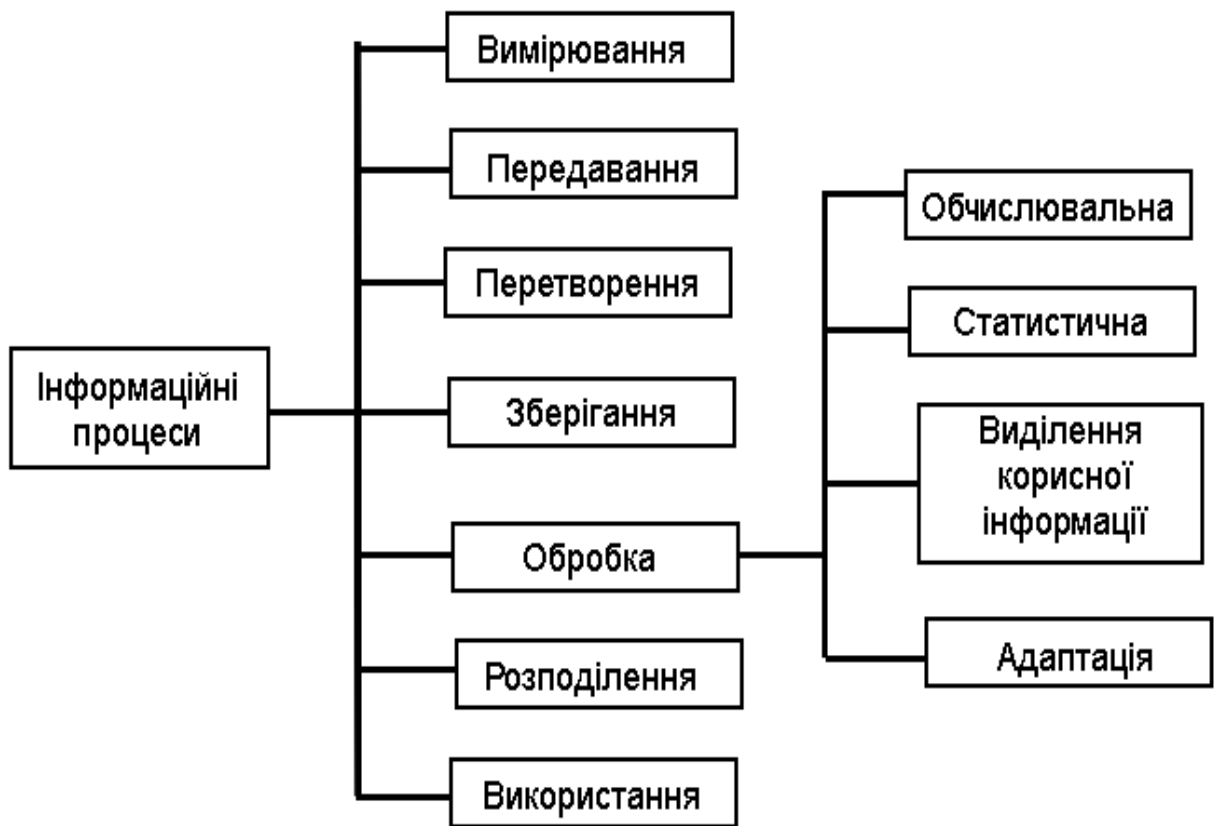


Рисунок 1.1. - Схематичне зображення інформаційних процесів у ІВС

Вимірювальна частина ІВС визначається як технічне, математичне і метрологічне, що ілюструє рисунок 1.2. У ІВС процес вимірювання і вхідних і

проміжних величин закінчується видаванням вимірювальної інформації у форматі, що є зручним до подальшого практичного використання.

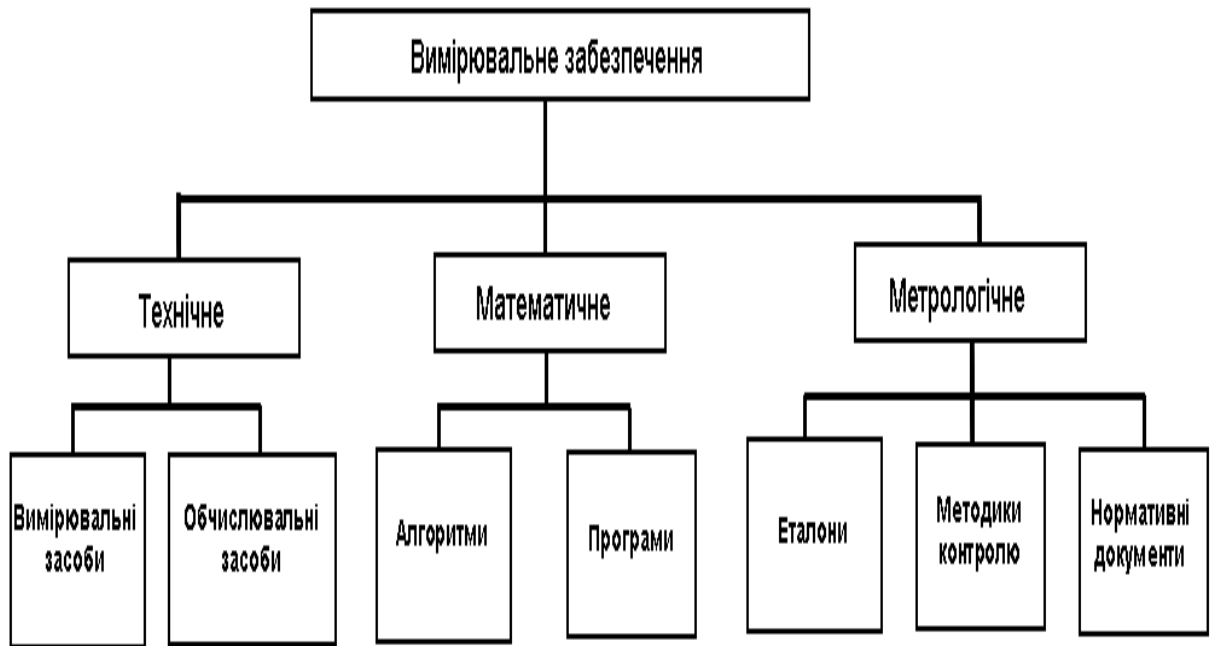


Рисунок 1.2. - Схематичне зображення вимірювального забезпечення у ІВС

Науко-технічною основою створення будь-якої системи, або комплексу є системна сумісність всіх функціональних елементів, що входять до її складу. Для передавання еквівалентних вимірюваних величин до комп'ютера, іншого гаджета необхідно забезпечити умови сумісності властивостей і характеристик всіх елементів системи [6].

Основні функції (операції) ІВС реалізуються загальною схемою, що представлена на рисунку 1.3. Схема складається з сукупності первинних вимірювальних перетворювачів (ПВП) – сенсорів (**sensor**), вмонтованих в об'єкт вимірювання, центральну ЕОМ та засоби, що забезпечують сумісність сенсора з ЕОМ. Процесори аналогових сигналів це пристрої первинної обробки

інформації. У наведеній схемі необхідним елементом є комутація каналів, що здійснюється всередині наведених засобів. Дистанційне ж передавання даних також є надзвичайно важливою складовою. На схемі (рис.1.3) наведений первинний вимірювальний перетворювач, яким є, власне, сенсор.

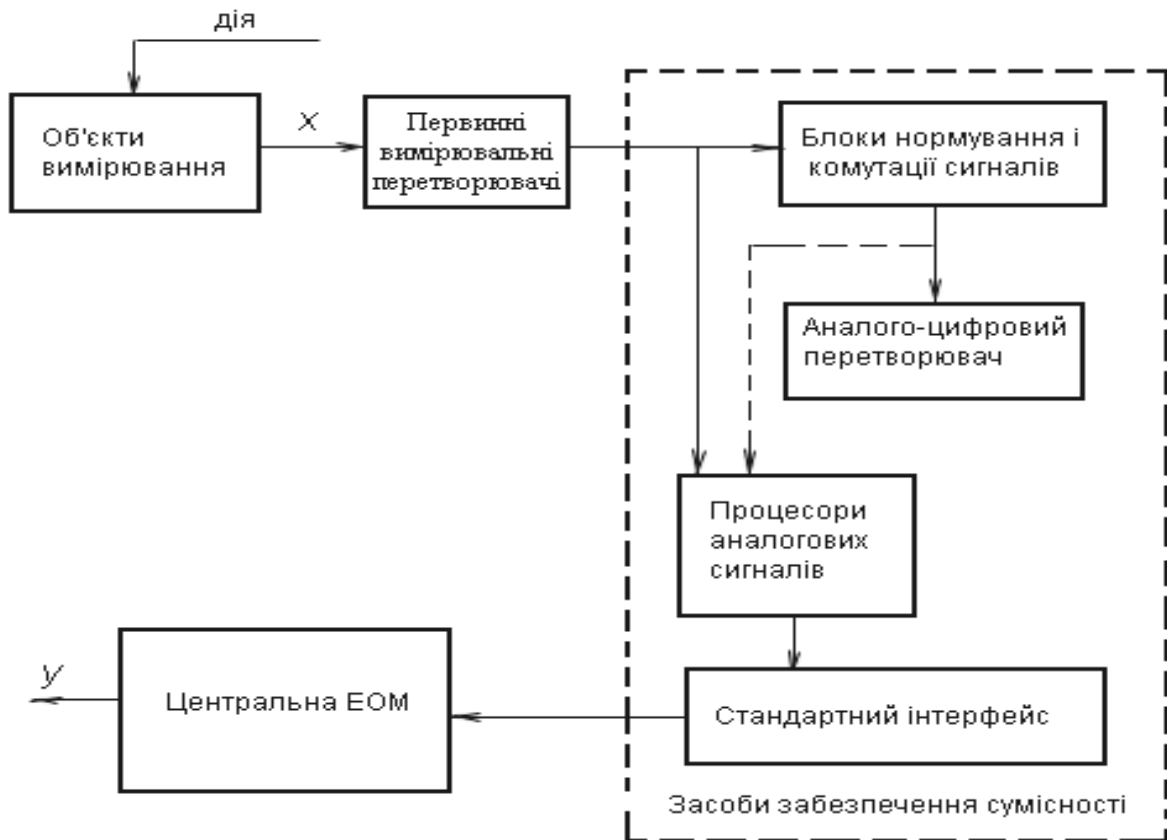


Рисунок 1.3. - Загальне схематичне зображення ІВС

Засобами забезпечення сумісності є: блоки нормування комутації; аналого-цифровий перетворювач; процесори аналогових сигналів; стандартний інтерфейс. Центральним елементом є комп'ютер з повною інформацією. Рівняння, що зв'язує вхідні і вихідні сигнали ІВС можна представити у загальному вигляді:

$$\{Y\} = \Phi[\{X\}],$$

де  $\{Y\}$  - множина вихідних змінних системи;



$\{X\}$  - множина вхідних вимірюваних і впливаючих величин;

$\Phi$  - оператор системи, що встановлює зв'язок між вхідною та вихідною величинами, з урахуванням алгоритму задачі, що вирішується.

Відповідно з прийнятою в математичній фізиці термінологією задачі вимірювання (контролю), що вирішує ІВС можна віднести до двох загальних класів: прямі і зворотні задачі.

Перша задача це визначення відгуку - поведінки об'єкта за відомою інформацією про вхідний сигнал. Для вирішення прямої задачі на об'єкт контролю діють еталонним випробувальним сигналом і за вимірюваною інформацією на виході об'єкта – відгуку визначають характеристики об'єкта, що обумовили його поведінку у часі і просторі.

Об'єктом у такому процесі виступає, власне, ІВС, якщо необхідно провести її випробування – калібрування, або метрологічну атестацію.

Друга задача – це визначення вхідного сигналу ІВС за отриманим відгуком на виході. Будь-яка задача інтерпретації результатів спостережень (вимірювань), вирішення якої дозволяє виключити або зменшити вплив на ці результати похибок засобів, що використовуються, і обробки даних, є зворотною.

Задача вимірювань у визначенні (ідентифікації) процесу  $x(t)$  по зареєстрованому сигналу  $x_p(t)$  та оператору  $\Phi = A \cdot A^{-1}$ , де  $A$  – прямий оператор засобів вимірювання, що використовуються, і перетворення вимірювальної інформації в сигнал, визначається шляхом розв'язку задачі:

$$x_p(t) = A[x(t) + \varepsilon(t)]$$

де  $\varepsilon(t)$  - загальна похибка вказаних засобів;

$A^{-1}$  - оператор, зворотній прямому оператору  $A$ .

Зворотній оператор повинен приводити процес  $x_p(t)$  до процесу  $\tilde{y}(t)$ , який є оцінкою вхідного процесу  $x(t)$ .

Вхідний процес відтворюють відповідно до заданих критеріїв точності. Зворотній оператор може бути реалізований центральним процесором ІВС, в тому числі в сполученні з процесорами аналогових сигналів.

Вирішення прямої і зворотної задач базується на аналізі сигналів і процесів в часі (параметричний аналіз), частотній області (спектральний аналіз) і на аналізі статистичних характеристик сигналів як в часі та і в частотній області (статистичний аналіз).

Такий поділ аналізу сигналу пов'язаний з існуванням двох видів вимірювань – статичних і динамічних.

Універсальні ІВС потребують великої кількості апаратних (hard) і програмних (soft) засобів, тому що вони мають вирішувати різні класи задач.

Обробка даних в таких системах є централізованою, а отже ЕОМ повинна забезпечуватися бібліотекою різноманітних програм.

Вибір структурної схеми ІВС визначається характеристиками функціональних елементів, що використовуються, та їх можливостями сумісності в системі.

Основу ІВС з децентралізованою обробкою даних складають мікропроцесори аналогових сигналів, що вирішують типові задачі вимірювань.

Саме таку ІВС для визначення метеорологічних показників (тиск, температура, вологість повітря) нами буде запропоновано у чинному дипломному магістерському дослідженні.

Обов'язковим функціональним елементом розробленої нами ІВС є аналого-цифровий перетворювач (АЦП). Призначення АЦП у ІВС полягає у тому, щоб забезпечувати сумісність джерел безперервної вимірювальної інформації – сенсорів - з процесором, або реєстратором за формою представлення інформації.

Реалізація умов і таких видів сумісності об'єкта вимірювання з процесором і, власне, самого АЦП з процесором і системою виокремлюються наступними показниками: метрологічними, часовими, просторовими, інформаційно-енергетичними, електро – енергетичними та ін.

## **1.2. Забезпечення метрологічної сумісності об'єкта вимірювання з процесором**

Розглянемо загальносистемні умови оцінки точності ІВС, що пов'язані з питаннями метрологічної сумісності системи і визначенням загальної точності результату вимірювань.

Метрологічна сумісність це погодження метрологічних характеристик всіх функціональних елементів (ФЕ), що створюють вимірювальний шлях перетворення, передавання і обробки інформації.

Метрологічна сумісність ФЕ досягається шляхом встановлення єдиного складу метрологічних характеристик, єдиних форм їх представлення, способів нормування, методів оцінювання і контролю.

Основною умовою метрологічної сумісності функціональних елементів є забезпечення балансу похибок методичних, алгоритмічних, апаратурних і програмних засобів системи, який базується на гранично допустимій похибці результату поставленої задачі.

Аналітичний запис вказаного балансу визначається характером нормованих похибок, які вимагає суб'єкт, і які відповідають умовам достовірності.

Точність результату вимірювання на виході ІВС має відповідати середній квадратичній похибці  $\delta$ , яку можна розрахувати за формулою

$$\delta = \sqrt{\delta_{aцп}^2 + \delta_{\kappa}^2 + \delta_{\mathcal{M}}^2 + \delta_{in}^2 + \delta_{mp}^2}, \quad (1.1)$$

де  $\delta_{aцп}$  - похибка АЦП;

$\delta_k$  - похибка перетворення сигналів в каналі передавання даних;

$\delta_m$  - методична похибка, що обумовлена обраним чисельним методом обчислення;

$\delta_{in}$  - інструментальна похибка, що обумовлена кінцевою розрядністю процесора і похибкою округлення результату обчислень;

$\delta_{tr}$  - трансформована похибка, що залежить від чисельного методу, похибки представлення вимірювальних величин і похибки апроксимації кінцевого результату.

Допустима статична похибка АЦП, яка складається з похибки кодування вимірювальної величини і похибок динамічних вимірювань та сенсорів, впливає на величину  $\delta$  найбільше, ніж інші похибки.

Необхідним і обов'язковим елементом сучасної ІВС є традиційні стандартні сенсори (первинні перетворювачі-датчики), незважаючи на властиві їм переваги, мають низку технічних і економічних недоліків. Для ефективної роботи їх необхідно обов'язково калібрувати. Це означає, що його вихідні дані мають відповідати визначеному стандарту, щоб отримані значення коректно відображали вимірюваний параметр, тобто відповідав певній точності.

Наступна проблема стандартних сенсорів це наявність дрейфу, який характеризує певну властивість, її зміни за часом. На показники стандартних сенсорів впливають зміни навколишнього середовища.

Сигнали стандартних сенсорів частіше за все дуже малі і потребують створення спеціальних схем підсилення. Це означає, що різні типи стандартних сенсорів, або сфери їх застосування вимагають спеціального налаштування схем формування сигналу. Стандартні сенсори необхідно встановлювати поблизу до систем керування, що приймають дані для конкретної ІВС.

Стандартний сенсор з аналоговим вихідним сигналом представлений на рис.1.4. Тут наведений приклад термоелектричного перетворювача, який призначений для вимірювання і регулювання температури в технологічному процесі виготовлення дротових і волоконно-оптичних кабелів.



Рисунок 1.5. - Стандартний сенсор з аналоговим вихідним сигналом

Найбільше складним при виготовленні подібних кабелів є регулювання температури від 70 до 400 °С в процесі створення оболонки-ізоляції на всіх видах цієї продукції. При цьому, від точності відтворення і регулювання температури залежить якість продукції. Найбільше розповсюдженими, точними і добре дослідженими і не дуже дорогими є стандартні термоперетворювачі термоелектричні – термоелектричні аналогові сенсори. Термоелектричні аналогові сенсори мають двоелектродний чутливий елемент – термопару, де виникає термоелектричний ефект при різниці температури робочого і вільного кінців електродів. Термопари коштують дешево, працюють в широкому діапазоні, а їх недоліки (малий сигнал, нелінійна характеристика, вимоги компенсаційних та подовжувальних дротів тощо) можуть бути компенсовані. На виході стандартного термоперетворювача термоелектричного при зміні температури виникає невелике значення напруги (у межах *мілі*Вольт), яка вимірюється в присутності сильного електричного шуму (що може сягати сотні Вольт, наприклад від нагрівачів печей розжарення пластмаси). Досить ускладнює процес вимірювання те, що температурна характеристика стандартного сенсора є нелінійною. Тому перед передаванням величини напруги необхідно виконувати операцію лінеаризації.

Отже, від кожного стандартного сенсора, що вимірює будь-який показник (тиск, температуру, вологість) в одній конкретній зоні (а таких зон моніторингу параметра є багато), до регулятора показника у ІВС виникає багато шумів. Особливо це важливо враховувати там, де існує безліч з'єднань з використанням компенсаційних дротів.

Отже, для гарантування якісної роботи сенсорів в ІВС, необхідно визначити наступні важливі фактори:

- типи сенсорів;
- необхідний діапазон вимірювань;
- необхідну роздільну здатність вимірювань;
- точність вимірювань;
- характеристики сигналу від сенсора;
- джерела шуму в вимірюваному сигналі.

Для вирішення перерахованих проблем перш за все необхідно аналоговий електричний сигнал перетворити в цифровий (числовий) вигляд. Для цього і створені аналого-цифрові перетворювачі (АЦП). Якщо після цього додати в схему мікропроцесор і обладнати датчик стандартними засобами передавання його інформації іншими елементами, то отримаємо (Smart sensor), тобто, інтелектуальний сенсор, схематичне зображення якого наведено на рис.1.6.

Для стандартного і інтелектуального датчиків спільним є тільки чутливий елемент. Саме чутливий елемент вимірює його фізико-хімічний параметр.

В інтелектуальному сенсорі окрім чутливого елемента присутні:

- схема дискретизації, що перетворює аналоговий сигнал в цифровий;
- обчислювальний елемент, який аналізує данні вимірювань, що отриманні за допомогою чутливого елемента;
- інтерфейс з зовнішнім середовищем, що дозволяє пристрою здійснювати обмін інформацією з іншими компонентами у ІВС.

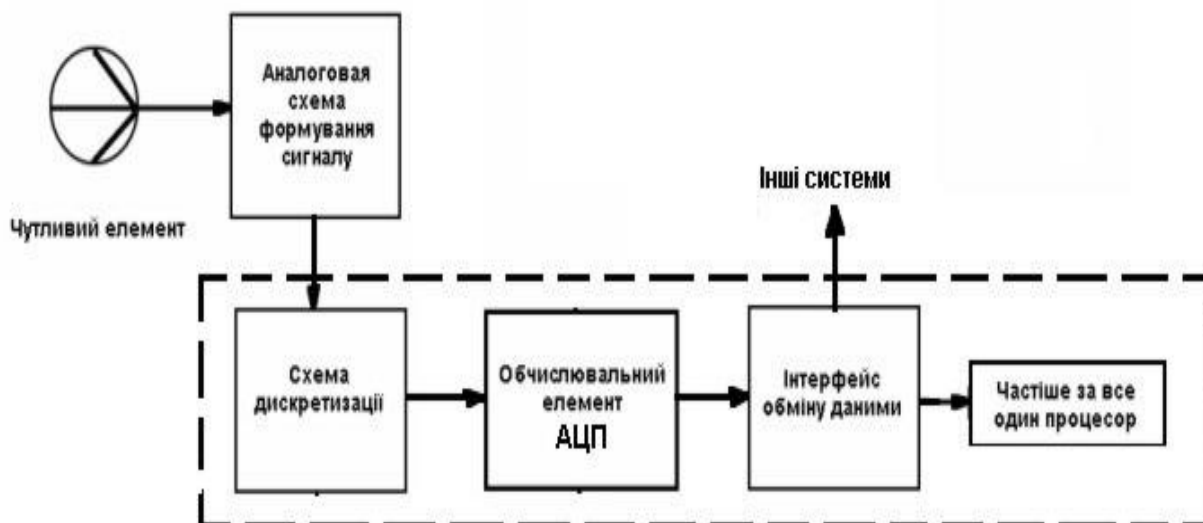


Рисунок 1.6 – Схема інтелектуального сенсора

При вирішенні реальної задачі, що пов'язана з обробкою сигналу інтелектуального сенсора необхідними є наступні складові:

- фізична характеристика, що вимірюється;
- взаємозв'язок між вимірюваною фізичною характеристикою і фіксованим значенням параметру;
- очікуваний частотний спектр сигналу і вірогідних джерел шуму навколишнього середовища;
- фізична характеристика робочого середовища;
- умови виникнення похибок і відповідні методики їх унеможливлення;
- вимоги до калібрування;
- вимоги до системного інтерфейсу;
- вимоги до обслуговування.

Інтелектуальний сенсор має надійно виявляти типові похибки, обробляти і усувати недоліки; передбачати небезпечні стани і недоліки та забезпечувати коректну роботу системи, а також фіксувати все в протоколах.

На рисунку 1.7 наведена схема структурна функціональна інтелектуального сенсора, що може бути використаний при вимірюванні температури.

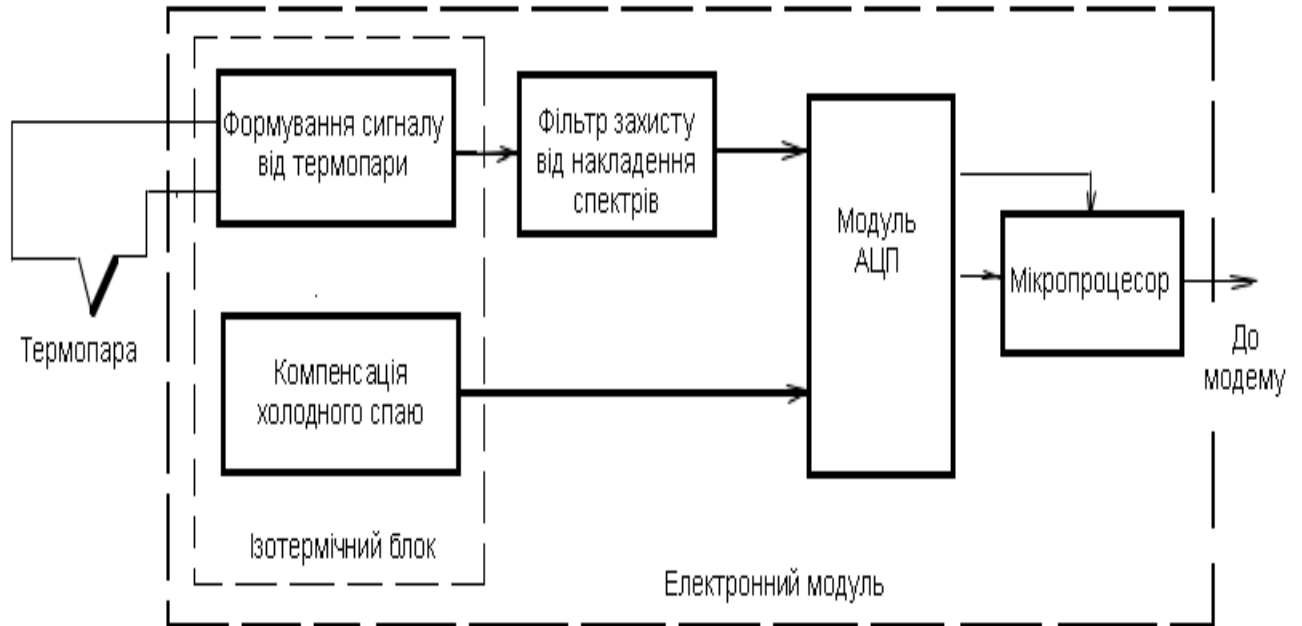


Рисунок 1.7 - Функціональна схема інтелектуального сенсора з чутливим елементом, до прикладу, термопарою

У наведеному прикладі (на схемі, рис.1.7) враховані три основні вимоги для інтелектуального сенсора:

- мінімізація споживчої потужності для зменшення самонагрівання;
- якісна і компактна друкована плата, де завади і електричний шум мінімізовані;
- якісна розв'язка джерела живлення для зменшення електричного шуму.



## **2. ВИМІРЮВАЛЬНА СИСТЕМА ДЛЯ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОГО СЕНСОРА**

### **2.1. Чутливий інтелектуальний сенсор на основі гетеропереходу**

Методам управління, діагностики, контролю параметрів навколишнього середовища присвячена низка статей і монографій [1-12]. Серед яких слід виділити температурні та радіаційні методи. У цих підходах, як правило, не торкаються шляхів використання новітніх технологій – т.зв. нанотехнологій [4-6]. Зокрема, мова йде про такі прилади, які експлуатують квантово-розмірні ефекти. У цьому контексті, на основі єдиного підходу, що класифікує різноманітні прояви квантово-розмірних (зокрема, кластеризованих) фаз речовини, у цьому дипломному проекті наведено результати дослідження механізмів формування плівкової кластеризованої структури (ПКС), яка може стати основним компонентом інтелектуального сенсора для вимірювання фізико-хімічних властивостей навколишнього середовища, зокрема атмосферного тиску.

Запропонована нами ІВС використовується для визначення морфологічних параметрів (маса, товщина шару, геометрія) складових інтелектуального сенсора. Для проведення кількісних оцінок були розроблені програмні засоби, інтерфейс для роботи ІВС.

У вимірювальній техніці знаходять широке застосування гетеро структури до складу яких входять тонкі плівки з високою надчутливістю. Підбір матеріалу і товщину плівок обирають з урахуванням потрібних, для проведення вимірювань, фізико-хімічних властивостей, або особливостей навколишнього середовища. Як правило, перевага віддається плівкам, що мають співрозмірну з підкладкою основного кристалу зонною структурою, відповідною радіаційною та температурною стійкістю.

В такому контексті перспективними є саме ті плівки, що мають нанокластерну морфологію і сформовані на поверхні твердотільної речовини. Технологія виготовлення таких плівок, як складових чутливого елемента

інтелектуального сенсора, або інших приладів електронної техніки – це комплекс складних технологічних процесів, одним з етапів якого є вимірювання її товщини та визначення її морфології – субмікоронної структури.

Нами досліджувалися процеси формування нанокластерної підсистеми на основі кремнію, що були синтезовані з парової фази на підкладинку монокристалічної Si- пластини. В якості підкладинки використано монокристали з площею 0,28 см<sup>2</sup>, та товщиною у декілька мкм.

Підкладинка наносилася на поверхню Ag-електродів кристалічного кварцу, який є основою пьезокварцового резонатора. Кристалічна структура нанокластерної підсистеми відповідала «сильно розмитій текстурі» (що визначається нами, як кластерний растр на Si).

Досліджувані зразки нанокластерної підсистеми синтезувалися методом відкритого випаровування порошку кремнію у вакуумі. Їх товщина контролювалася за допомогою мікроінтерферометра типу МІ-4. Електронно-мікроскопічні дослідження морфології плівки, тобто її субмікронної структури, проводилися за допомогою електронного мікроскопу (GSI, Darmshtadt, Germany).

Процес формування нанокластерної підсистеми проводився з використанням п'єзо – кварцевого датчика зі сталою частотою у 1,67 МГц/мм.

Датчик підключався до стандартного кварцового вимірювача товщини плівки (КВТП-5, розроблений в Інститут фізики напівпровідників НАНУ країни імені проф. В.С.Лашкарьова) з попередньо проградуїрованою шкалою (на монокристал кремнія).

Градувальний графік будувався таким чином, щоб його можна було вжити для вимірювання малих товщин, що наносили на підкладинку кремнію. Оперативний контроль товщини нанокластерної підсистеми проводився методом кварцевого резонатора з елементами методики, яка розроблена групою проф. кафедри АСМНІ В.В. Ковальчука (ОДЕКУ)

Принцип роботи КВТП-5 базується на вимірюванні зсуву резонансної частоти кварцевого датчика в залежності від осадженого на нього шару

матеріала. Система дозволяє провести вимірювання товщини напиленого у вакуумі покриття з точністю до 0,1 нм.

Призначення КВТП: а) кварцевий контроль фізичної товщини матеріалу у процесі нанесення у вакуумі тонкоплівкової структури на підкладку; б) для вимірювання використовують ефект зсуву резонансної частоти кварцового кристала при зміні її маси. Маса кристала збільшуються за рахунок нанесення матеріалу на його поверхню.

КВТП-5 – це програмуючий прилад. Зовнішній вигляд КВТП-5 представлений на рис. 2.1. Прилад дозволяє проводити керування процесом напилення матеріалу на поверхню підкладки за допомогою програмного забезпечення на основі пакету SciLab.

КВТП-5 – оснащений клавіатурою, багатофункціональним дисплеєм для візуалізації режимів роботи і програмування параметрів процесу нанесення тонкої плівки.



Рисунок 2.1 – Кварцевий вимірювач товщини плівки КВТП-5

Процедуру напилу шару матеріалу на підкладинку проводять на основі аналітичного співвідношення, що відповідає математичній моделі текстури плівки. Таким чином можна задавати складні, багатошарові плівки. Кожний шар характеризується конкретною товщиною і типом покриття, густиною напилюючого матеріалу та коефіцієнтом розпилення у камері. Виходячи з цих параметрів та відносної зміни частоти кварцевого датчика, розраховують товщину плівки.

Генератор кварцевого кристалу BF-100 спроектовано для збільшення терміну роботи кварцових кристалів промислового стандарту з робочою частотою у 5 та 6 МГц. Генератор забезпечує роботу кварцевого кристала, формує, або (та) підсилює сигнал для передачі його до КВТП-5. Генератор BF-100 працює з кабелем на відстані у 1 м. від кристала. Кабель, що з'єднує генератор і контролер, можуть мати довжину до 4-х м. Генератор BF-100 має модифікацію корпусу з проушинами для кріплення на плоску поверхню підкладинки.

Охолоджуваний за допомогою води тримач кристалу типу BFAG-140 використовують у складі системи кварцевого контролю товщини покриття. Датчик запроектований таким чином, щоб заміна кристала кварца відбувалася за максимально спрощеною схемою. До утримувача типу BFAG-140 встановлено стандартні кварцові кристали з геометричним розміром 0,55 мм. Наявність у BFAG-140 системи водного охолодження дозволяє здійснювати контроль за нанесенням плівкового покриття при температурі до 300°C.

Основні характеристики КВТП-5:

- а) діаметр патрубків для охолодження водою до 5 мм;
- б) довжина патрубків – до 500 мм;
- в) матеріал корпусу – нержавіюча сталь; д) термостійкий коаксиальний кабель.

Утримувач кристалу встановлюють у вакуумну камеру через герметичний вхід. До BFAG-140 приєднують два входи для трубок діаметром 5 мм і один електричний вхід. Трубки для водоохолодження можна вигинати. Це

дозволяє орієнтувати положення кристала у камері. З зовнішньої сторони камери до електричного виходу підключають генератор для формування та підсилення сигналів, що поступають від кварцевого кристала.

Сигнал від генератора подається на вимірювальний прилад, що розраховує товщину нанесеної плівки шляхом зміни частоти коливань кварцевого кристалу. Утримувач може додатково оснащуватися пневматичною заслонкою для того, щоб запобігти потраплянню пилу на кристал при різних режимах нанесення покриття. Герметичний вхід є необхідним для введення трубок, що забезпечують охолодження кварцевого держателя всередині камери водою. У комплект з приладом КВТП-5 входить програмний пакет на основі SciLab.

Автоматизована система роботи КВТП-5 забезпечена протоколом. Сервісна програма на платформі SciLab для персонального комп'ютера дозволяє забезпечити нанесення більш складних за морфологією плівок. Програма дозволяє створювати необмежену кількість шарів та плівок з використанням будь-яких матеріалів. Забезпечує зберігання усіх конфігураційних даних та їх редагування. Графіки дозволяють наочно проілюструвати та відслідковувати хід нанесення покриття.

У камері, де підтримують вакуум  $\sim 1 \cdot 10^{-6}$  Торр, розташовано пластини кварцевого вимірювача на сіталої основі, лодочки з фольги тантала, де знаходиться кремній, який випаровується. Через контакти на тантал подають напругу, що нагріває Si до температури випаровування. Струм змінюється в межах 10 – 15 А. Задаючи силу струму, можна отримувати різні за товщиною плівки, різної морфологічної структури.

Інша можливість нагрівання танталової лодочки є подача на електроди через контакти імпульсів напруги, тривалість яких задають таймером. Частота послідовності імпульсів змінювалася у межах від 0,1 до 0,01 Гц. Частота імпульсів була такою, щоб забезпечити відновлення вакуумних умов у вакуумній камері у проміжку між розрядами. Енергія, що випромінювалася при розряді змінювалася від імпульсу до імпульсу. Для контролю параметрів розряду

розроблено схему вимірювання енергії, що виділяють на електродах за один імпульс (рис. 2.2).

Мікрофотографія плівки (індекс 7 на рис. 2.2), була отримана за допомогою тунельного мікроскопу, який був використаний на базі Інституту фізики напівпровідників АН України ім. В.Є. Лашкарьова.

Кластерний растр плівки на підкладці кремнія дозволяє провести оцінку геометричних розмірів глибоко субмікронних утворень Si (з розмірами у  $15 \div 20$ ) – нанокластерів, які відтінялися вуглецем.

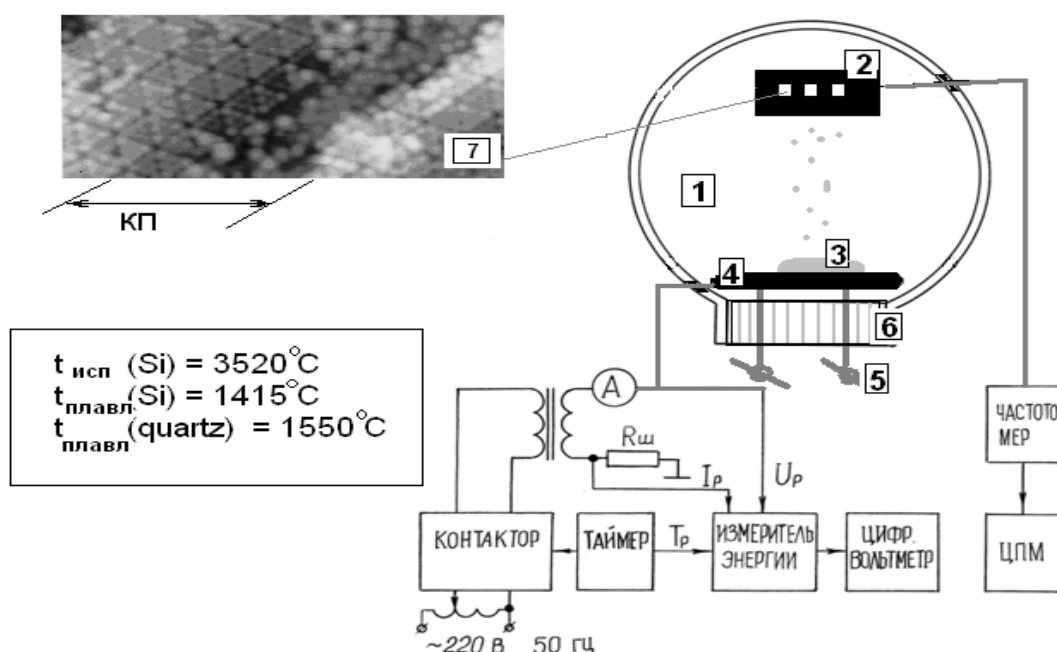


Рисунок 2.2. – Схема експериментальної установки для отримання плівки, як надчутливого елемента датчика

Для отримання особливо тонких плівок товщиною  $\sim 25-50 \text{ \AA}$  передача енергії, що випромінюється при подачі струму на тантал, має бути достатньою для того, щоб густина потоку випарюваних атомів Si була сталою.

## 2.2. Вимірювання товщини чутливого елемента інтелектуального сенсора методом кварцевого резонатора

Як зазначалося вище, основним компонентом КВТП-5 є платівка кварцевого резонатора, що визначає частоту стабільного автогенератора. Оскільки генерована частота залежить від маси кварцевого резонатора, то у випадку відходу генерованої частоти, з урахуванням проведеного калібрування, можна визначити товщину плівки, що осаджена на кварцовій підкладинці, а отже і на зразках що знаходиться у тих самих умовах.

Маса нанесеного на одиницю поверхні кварца шар  $\Delta m_c$  і зміна частоти кварцевого резонатора  $\Delta f$  пов'язані простим співвідношенням:

$$\Delta m = \frac{m_g}{f_g} \Delta f, \quad (2.1)$$

де  $m_g$  – маса кварцевої підкладинки одиничної площі;  
 $f_g$  – резонансна частота.

Чутливість метода кварцевого резонатора є досить високою. Так, наприклад, для кристала з резонансною частотою  $f=5$  МГц, масою  $m_g=100$  мг чутливість дорівнює величині  $\frac{m_g}{f_g}=0,02$  мкг/Гц. Тобто реєструючи відхід частоти

кварцевого резонатора на 1 Гц, можна контролювати збільшення маси плівки у соті долі мікрограма.

Слід зауважити, що кварцовий резонатор досить чутливий до зміни температури.

Щоб врахувати відхід резонансної частоти при зміні температури, у наших вимірюваннях використано кварцові резонатори, що були відібрані за результатами вимірювань температурної залежності резонансної частоти.

Нами було проведено тестування понад 40 резонаторів марки РК170 и РК171 з частотою власного резонанса у 5...10 МГц.

Найбільш типові залежності  $f(T)$  наведені на рис. 2.3.

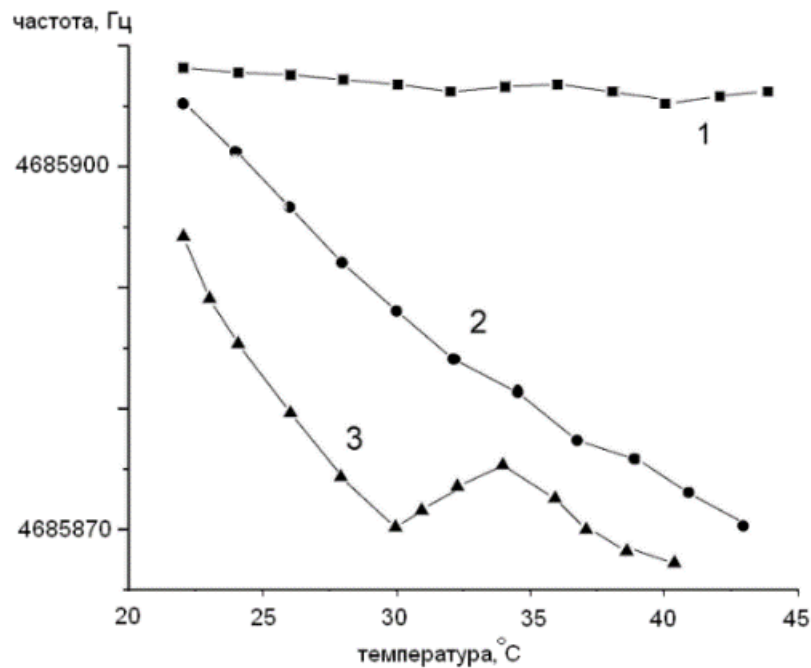


Рисунок 2.3. – Температурні залежності частоти кварцового резонатора від температури

Як видно з графіка температурна крива 1 характеризується слабкою залежністю частоти від температури. Таку графічну залежність мали приблизно п'ять відсотків резонаторів. Приблизно 75% резонаторів характеризувалися спадаючою залежністю (крива 2 на рис. 2.3). Між іншим, нами було виокремлено (відбраковані) приблизно 20% резонаторів, які характеризувалися ломаними залежностями (типу 3) з яскраво визначеним екстремумом.

Крім усього вищезазначеного, враховуючи, що точність методу вимірювання маси нанесеного плівкового шару на підкладку з використанням кварцового резонатора, визначається стабільністю частоти генератора нами запропонована електрична схема стабільного кварцового генератора, яка наведена на рис. 2.4.



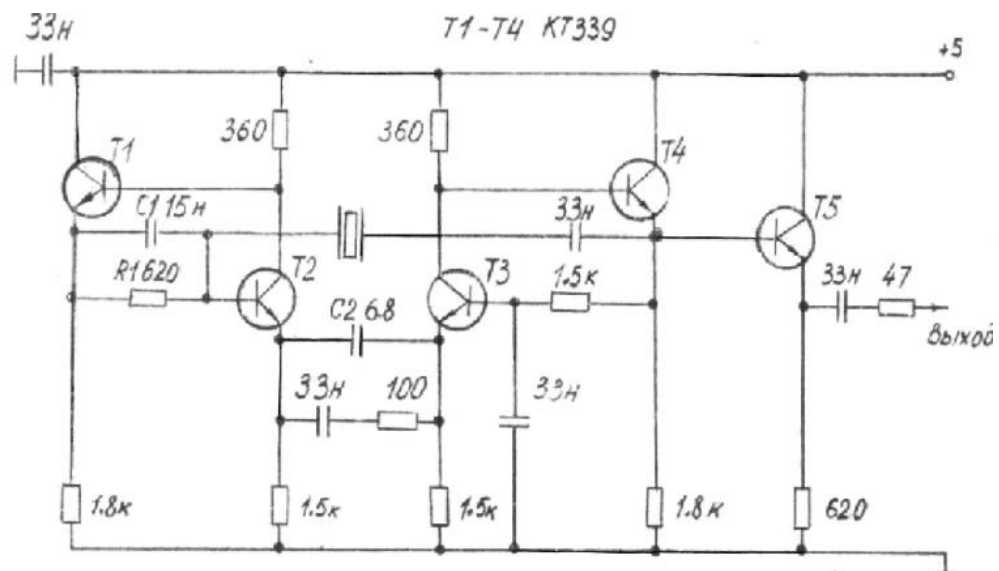


Рисунок 2.4 – Принципова електрична схема кварцевого генератора

Генератор призначений для роботи у діапазоні частот від 1 до 100 МГц і зібраний на високоякісному, надчутливому, високочастотному підсилювачі з кварцовим резонатором (транзистори Т1-Т4). Транзистор Т5 є буферним емітерним повторювачем, що зменшує вплив навантаження на параметри генерованого сигналу. Підсилювач генератора виконаний за схемою: загальний колектор – загальна база – загальний колектор (Т2-Т4). Транзистор Т1 включено в мережу від’ємного оберненого зв’язку, що нейтралізує ємність утримувача кварцу. Між іншим, від’ємний обернений зв’язок стабілізує амплітуду коливань на виході. Корегуюча ємність  $C_2$  збільшує стійкість генерації на високих частотах.

У схемі генератора була досягнута широка полоса пропускання сигналу підсилювача. Саме тому підсилювач має неймовірно малий додатковий набіг фаз майже до частот приблизно у 100 МГц. А це, у свою чергу, обумовлює високу стабільність частоти генерування.

Сума вхідного і вихідного опору підсилювача є меншою за еквівалентний опір розсіювання кварцевого резонатора. Відносна нестабільність частоти генератора складає десяті долі мкГц.

Генератор змонтовано на невеликій платі. Він розміщений у вакуумній камері. Живлення та зняття результуючого сигналу відбувається через вакуумний роз'єм. Таке розміщення дозволяє суттєво скоротити довжину провідників, що з'єднують кварцевий резонатор і зменшують наводки на схему. У момент випаровування кремнію виділяється енергія, частина якої виникає у вигляді випромінювання. Останнє здатне розігріти платівку кварцевого резонатора. Проведені нами вимірювання довели, що температура підкладки дещо збільшується на протязі сеансу. Відновлення відбувається через 2-3 хв. Щоб уникнути паразитний вплив температурного відходу частоти на результати вимірювання, були використані саме ті резонатори, що мали слабку температурну залежність (крива 1, рис. 2.5).

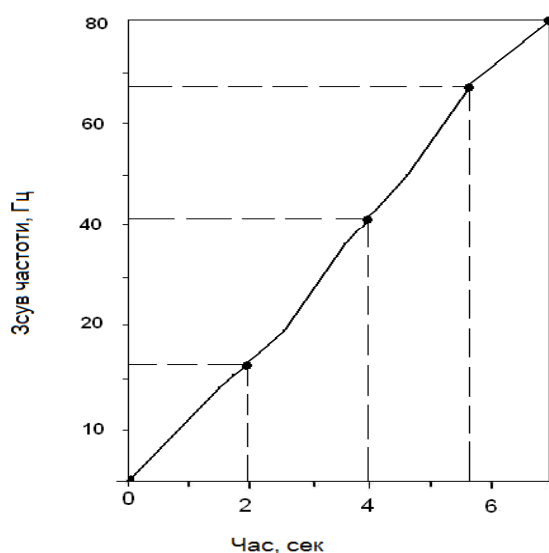


Рисунок 2.5 – Кінетичні криві синтезу плівок на підкладку кремнію, що отримані за допомогою п'єзо резонансного датчика

Рис. 2.5 представляє залежність, що ілюструє кінетику частоти резонанса кварцевого кристалла після проведення напилення плівки. Дрейф частоти після нанесення покриття обумовлений адсорбцією остаточного газу свіженанесеним шаром кремнію.

Використовуючи кристали зі слабкою температурною залежністю (крива 1 рис. 2.5) точність у зміні відходу частоти від резонансного значення  $\Delta f$  нам вдалося довести до 1 Гц.

Похибка метода обумовлена точністю, яка відповідає відходу від резонансної частоти  $\Delta f$  і точністю у визначенні маси кристалу  $m_g$ . У нашому випадку похибка вимірювання товщини покриття відповідає 5%.

Метод кварцевого резонатора дозволив нам розробити методику виготовлення плівок з контрольованою і наперед заданою товщиною.

У табл.2.1 наведено результати вимірювання товщини плівки методом кварцевого резонатора.

Таблиця 2.1 – Порівняння результатів вимірювання товщини плівки кремнію

Номер зразка	$\Delta f$ кварцу, Гц	Товщина плівки за даними кварцу, мкм
1	80	1,034
2	105	1,368
3	120	1,519
4	136	1,748
5	156	2,006
6	200	2,584
7	320	4,134

Отже, описаний у цьому розділі метод дозволив нам вимірювати товщину шару речовини (кремнію) в процесі напилення у чистому вигляді, тобто без урахування маси адсорбованої газу. Це є дуже важливим для визначення товщини мішені в експериментах з іонними потоками.

Наведена вище методика керування процесом формування певної товщини плівки, яка має глибоко субмікронну морфологію, як основа складової

чутливого елемента інтелектуального сенсора відкриває нові можливості для створення високоефективних приладів нового покоління.

За описаною вище методикою був створений чутливий елемент для інтелектуального сенсора, який використано для проведення вимірювань метеорологічних характеристик, математична і комп'ютерна модель якого описана далі.

Практична реалізація цієї технології дозволяє виокремити успіх технології отримання надчутливих плівок і приладі на їх основі.

У процесі реалізації описаного у цьому розділі метода створення чутливого елемента інтелектуального сенсора необхідно враховувати наступне.

Перше. Пружні властивості напиленої речовини відрізняються від пружних характеристик підкладинки.

Друге. Густина напиленої речовини є меншою за густину цієї ж речовини у масивному зразку.

Третє. Зсув частоти резонанса обумовлено не лише масою нанесеного шару, але й напругою, що виникає на границі розділу підкладинка-плівковий матеріал.

Четверте. КВТП-5 використаний для реалізації сучасної технології формування твердотільної матриці з наперед заданим нанокластерним растром колоїдної дисперсності.

П'яте. Запропонований комплексний підхід дозволяє використовувати цей метод, як метод неруйнівного контролю поверхні інтелектуального сенсора з нанокластерною морфологією.

### 3. АПАРАТНО ОБЧИСЛЮВАЛЬНА ПЛАТФОРМА ІВС

#### 3.1. Платформи для розробки проекту

Для розробки проекту були проаналізовані найбільш популярні платформи, а саме: Arduino, BeagleBone, RaspberryPi. Ці моделі були обрані для порівняння оскільки всі вони реально доступні, виконані в компактному вигляді і можуть використовуватися для створення різних цифрових пристроїв.

Платформу Arduino можна вважати найбільш поширеною у використанні серед професійних дослідників так і аматорів. На разі, плати Arduino доступні в різних форм-факторах, з різним набором периферії. Велика їх частина виконана на 8-розрядному мікроконтролері компанії Atmel.

Головна перевага плат Arduino (рис.3.1) – це порівняно просте проектування та збірка макетів. Програмування мікроконтролера здійснюється через спеціальну консоль, яка одночасно є середовищем розробки і терміналом послідовного порту.

Завантаживши програмний код в пам'ять мікроконтролера, спеціальною кнопкою на платі (або командою «логічна одиниця») на обраний цифровий порт вводу / виводу, запускається програмний код на циклічне виконання.

Плата Raspberry Pi (рис. 3.2) є також досить перспективним пристроєм. Це комп'ютер з однією платою, що позиціонується на ринку як дешеве рішення для розробників вбудованих ІВС.

Незважаючи на низьку вартість, ця платформа є окремим комп'ютером, що використовують для розробки великої кількості стартапів. Платформа використовує SD - карту для зберігання бібліотечного та системного програмного забезпечення. Існує можливість підключення широкого асортименту інтерфейсів, зокрема, мереж Ethernet, 3G, різного роду дисплеїв. Використовує мікропроцесор ARM1176JZF-S з тактовою частотою 700 MHz.

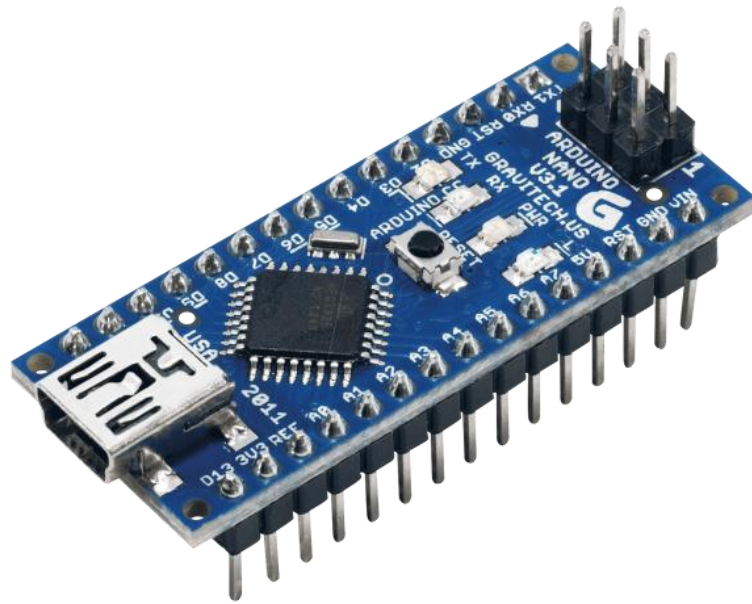


Рисунок 3.1 – Платформа Arduino

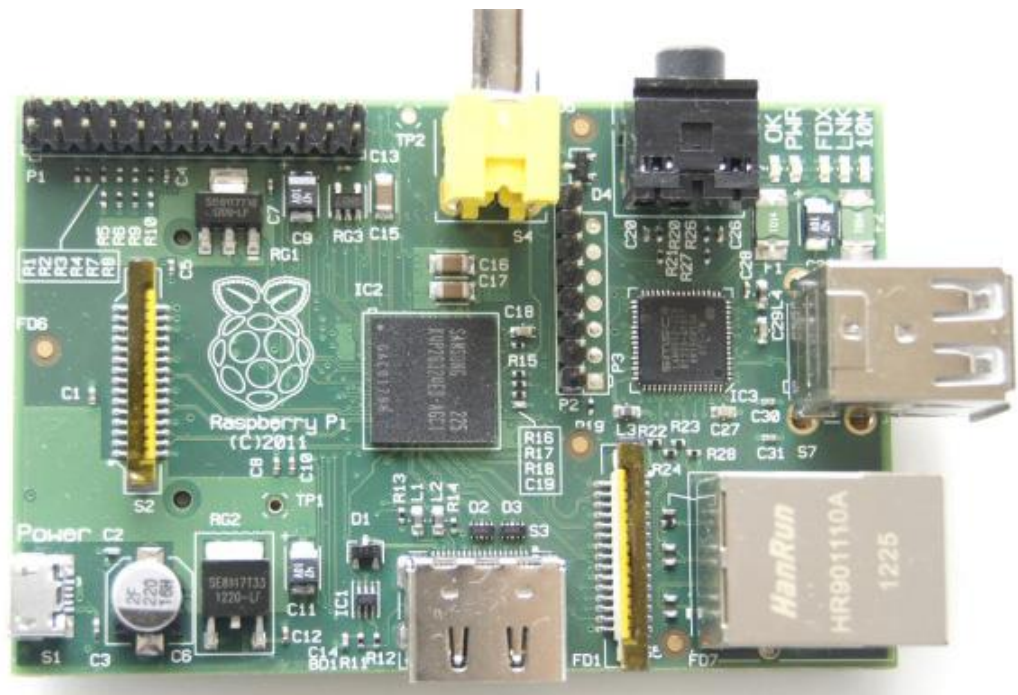


Рисунок 3.2 – Комп'ютер з однією платою Raspberry Pi.

BeagleBone (рис. 3.3) це потужний Linux-комп'ютер компактних розмірів, який підтримує роботу з ОС Android і Ubuntu . Загалом, ця плата подібна до Raspberry Pi, однак відрізняється більшою обчислювальною потужністю і

можливостями щодо розширення. Система побудована на обчислювальному процесорі TI Sitara AM335x з ядром A8 Cortex, якого вистачає навіть для роботи достатньо потужних планшетів.

Raspberry Pi буде оптимальним вибором для додатків, що вимагають наявності дисплея, графічного призначеного для користувача інтерфейсу і підключення до Інтернет, взаємодії з різного роду датчиками.

На платформі BeagleBone ідеально поєднана гнучкість Arduino, продуктивність процесора плати Raspberry Pi і ОС Linux. Маючи достатню кількість входів / виходів, BeagleBone забезпечує просте підключення до мережі і дозволяє реалізувати web-сервер.

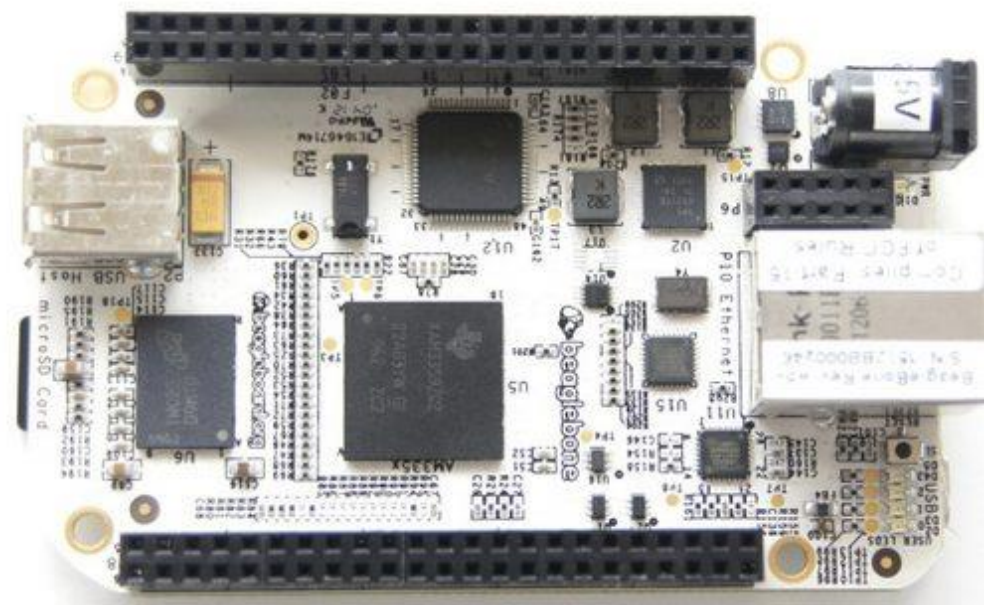


Рисунок 3.3 – Комп'ютер з однією платою BeagleBone

Нами була обрана Arduino, оскільки - це досить функціональна і гнучка платформа розробки, що містить вбудовані додатки з величезними можливостями для взаємодії із зовнішнім світом. Вона прекрасно, підійде для вивчення мікроконтролерів і може служити основою для практичних стартапів,

інженерних проєктів. Для виконання дипломного проєкту була використана плата Arduino на платформах Nano (рис. 3.4) і Uno.

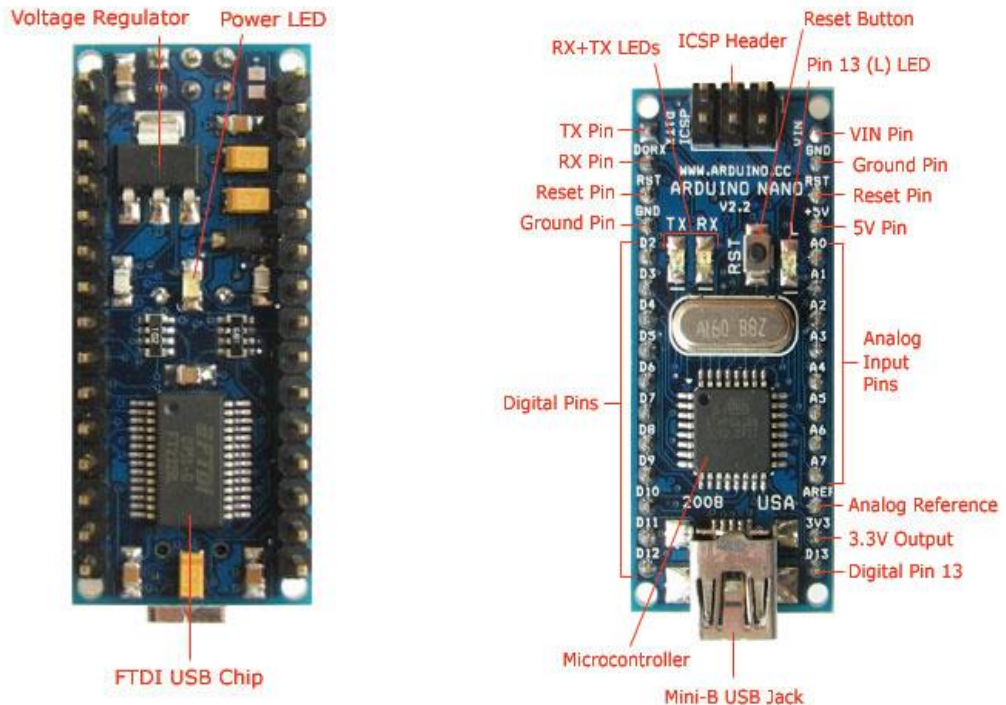


Рисунок 3.4 – Входи і виходи на платі Arduino Nano

Плата Arduino складається з мікроконтролера Atmel AVR (ATmega328P), а також елементів «прошивки» для програмування та інтеграції з іншими схемами. На багатьох платах присутнім є лінійний стабілізатор напруги + 5В або +3,3В. Тактування здійснюється на частоті 16 або 8 МГц кварцовим резонатором (в деяких версіях керамічним резонатором). У мікроконтролер попередньо прошивається завантажувач BootLoader, тому зовнішній програматор не потрібен.

На концептуальному рівні всі плати програмуються через RS-232 (послідовне з'єднання), але реалізація цього способу відрізняється від версії до версії.

У версії платформи Arduino Uno використовується мікроконтролер Atmega8 в SMD-корпусі. Таке рішення дозволяє програмувати конвертер



таким чином, щоб платформа відразу визначалася як миша, джойстик або інший пристрій на розсуд розробника з усіма необхідними додатковими сигналами управління.

Arduino Nano може отримувати живлення через підключення Mini-B USB, або від нерегульованої напруги у 6-20 В, або регульованої напруги у 5В від зовнішнього джерела живлення. Автоматично вибирається джерело з найвищою напругою.

Мікроконтролер ATmega168 має 16 кБ флеш-пам'яті для зберігання коду програми, а мікроконтролер ATmega328, в свою чергу, має 32 кБ (в обох випадках 2 кБ використовується для зберігання завантажувача).

ATmega168 має 1 кБ ОЗУ і 512 байт EEPROM (яка читається і записується за допомогою бібліотеки EEPROM), а ATmega328 - 2 кБ ОЗУ і 1 кБ EEPROM.

Кожен з 14 цифрових висновків Nano, використовуючи функції pinMode digitalWrite, і digitalWrite, може налаштовуватися як вхід або вихід. Виводи працюють при напрузі 5 В. Кожен вивід має навантажувальний резистор (стандартно відключений) 20-50 кОм і може пропускати до 40 мА. Деякі виводи мають особливі функції. А саме:

Послідовна шина: 0 (RX) і 1 (TX). Виводи використовуються для отримання (RX) і передачі (TX) даних TTL. Дані виводи підключені до відповідних виводів мікросхеми послідовної шини FTDI USB-to-TTL.

Будь-який з виводів забезпечує ШІМ з роздільною здатністю 8 біт за допомогою функції analogWrite.

ШІМ: 3, 5, 6, 9, 10, і 11. Будь-який з виходів забезпечує ШІМ з роздільною здатністю 8 біт за допомогою функції analogWrite.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). За допомогою даних виводів здійснюється зв'язок SPI, яка, хоча і підтримується апаратною частиною, не включена в мову Arduino.

LED: 13. Вбудований світлодіод, підключений до цифрового виводу 13. Якщо значення на виведення має високий потенціал, то світлодіод горить.

На платформі Nano встановлені 8 аналогових входів, кожен дозволом в 10 біт (тобто може приймати 1024 різних значення). Стандартно висновки мають діапазон вимірювання до 5 В щодо землі, проте є можливість змінити верхню межу за допомогою функції `analogReference`. Деякі виводи мають додаткові функції: I2C: A4 (SDA) і A5 (SCL). За допомогою виводів здійснюється зв'язок I2C (TWI).

В текст програми (скетчу) програміст не зобов'язаний вставляти заголовки використовуваних стандартних бібліотек. Ці заголовки додасть перед-процесор Arduino відповідно до конфігурації проекту. Однак призначені для користувача бібліотеки потрібно вказувати.

Слід зазначити, що менеджер проекту Arduino IDE має нестандартний механізм додавання бібліотек. Бібліотеки у вигляді вихідних текстів на стандартному C++ додаються в спеціальну папку в робочому каталозі IDE. При цьому назва бібліотеки буде додано до списку бібліотек в меню IDE. Програміст визначає потрібні бібліотеки і вони вносяться до списку компіляції.

### **3.2. Апаратні та програмні характеристики датчика тиску**

Використання інтелектуального сенсору, що описаний у розділі 2, впроваджувалося разом з барометричним датчиком тиску BMP180. Останній призначений для вимірювання барометричного, абсолютного, диференціального, надлишкового тиску, а також значення температури навколишнього середовища.

На початковому етапі, був скомпанований макет на основі датчика тиску, з попереднім підключенням його до управління і живлення. Потім датчик тиску калібрувався за допомогою спеціальних програм. Для нормальної роботи датчика з Arduino контролерами завантажувалася спеціальна бібліотека.

Після виконання таких дій проводилася подальша робота. Практичне застосування інтегрувалося в GPS навігатори, смартфони, прилади і апарати, що експлуатують ANDROID-додаток, які переміщуються в атмосфері, вимірювання тиску всередині і назовні для визначення прогнозу погоди, а також барометричних вимірювань висоти. На корпусі датчика є отвір для закріплення його на будь-якій поверхні і в будь-якому положенні.

Управління датчиком здійснювалося від Arduino контролера, або від іншого мікропроцесорного керуючого пристрою.

Чутливим елементом датчика є два типи мембран:

- А) в корпусі, що працює за п'єзорезистивним принципом (рис.3.5);
- Б) інтелектуальний сенсор, що описаний у розділі 2.

П'єзорезистивного ефект – це залежність опору матеріалу від величини його деформації.

Барометричний датчик тиску BMP180 має одну колодку з 5 клемами:

- VCC - напруга живлення;
- SDA - ІІС інтерфейс;
- SCL - ІІС інтерфейс;
- GND - загальний висновок;

Вимірювання величини тиску і температури передавалися за допомогою шини: послідовним моніторингом даних цифрового барометра-термометра.

Висота над рівнем моря обчислювалася за методикою, викладеною в документації мікросхеми BMP180. Кількість вимірювань в секунду може сягати 128.

Як правило, використовувався датчик в режимі один вимір за секунду. Це збільшує точність і економить струм споживання.

Датчик може вимірювати атмосферний тиск з різною точністю.

Мікроконтролер задавав датчику необхідну точність вимірювання.

Важливо правильно задати затримку в програмі перед зчитуванням регістрів даних.



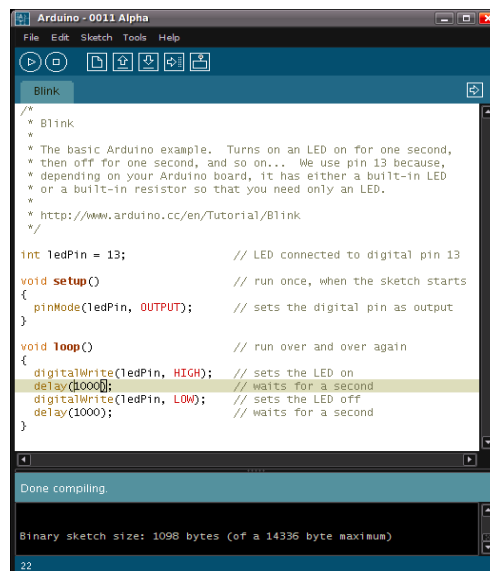
Рисунок 3.5 – Барометричний датчик тиску BME180

Характеристики:

- Напруга живлення: 3.3 або 5 Вольт
- Струм: 5 мкА при швидкості опитування 1 Герц
- Струм: 0.1мкА в режимі очікування
- Діапазон вимірювання тиску: від 300 до 1100 гПа (від -500 від +9000 метрів над рівнем моря)
- Шум: 0.06 гПа (0.5м) в грубому режимі (ultra low power mode) і 0.02 гПа (0.17м) в режимі максимального дозволу (advanced resolution mode)
- Діапазон вимірювання температури: від 0 до 65 ° C
- Точність вимірювання температури:  $\pm 2$  ° C
- Роздільна здатність: 16 біт
- Максимальна швидкість інтерфейсу: 3,4 Мбод
- Робоча температура: від -40 ° C ~ +85 ° C
- Діаметр монтажного отвору: 3 мм
- Розміри: 12 x 10 мм

- Вага: 1.18 гр.

Інтегроване середовище розробки Arduino - це багатоплатформовий додаток на мові програмування Java, що включає в себе редактор коду, компілятор і модуль передачі прошивки в плату. Середовище розробки ґрунтується на мові програмування Processing. Ця мова програмування аналогічна до мови Wiring. Строго кажучи, це C++, доповнений деякими бібліотеками. Програми обробляються за допомогою препроцесора, а потім компілюється за допомогою AVR-GCC. Програми Arduino написані на мові програмування C або C++. Середовище розробки Arduino йде разом із бібліотекою програм, яка називається «Wiring», який дозволяє робити багато стандартних операцій вводу/виводу. Це спрощує використання ANDROID-додатку (рис.3.6).



```
Arduino - 0011 Alpha
File Edit Sketch Tools Help
Blink
/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;          // LED connected to digital pin 13

void setup()              // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()               // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);              // waits for a second
  digitalWrite(ledPin, LOW); // sets the LED off
  delay(1000);             // waits for a second
}

Done compiling
Binary sketch size: 1098 bytes (of a 14336 byte maximum)
22
```

Рисунок 3.6 – Arduino IDE з прикладом простої програми

Середовище розробки Arduino складається з вбудованого текстового редактора програмного коду, області повідомлень, вікна виведення тексту (консолі), панелі інструментів з кнопками часто використовуваних команд і декількох меню. Для завантаження програм і зв'язку середовище розробки підключається до апаратної частини Arduino.

Програма, написана в середовищі Arduino, називається скетч. Скетч пишеться в текстовому редакторі, що має інструменти вирізки / вставки, пошуку / заміни тексту. Під час збереження і експорту проекту в області повідомлень з'являються пояснення, також можуть відобразитися виникли помилки. Вікно виведення тексту (консоль) показує повідомлення Arduino, що включають повні звіти про помилки та іншу інформацію.

Кнопки панелі інструментів дозволяють перевірити і записати програму, створити, відкрити і зберегти скетч, відкрити моніторинг послідовної шини.

**Мікроконтролер ATMEGA328P.** ATMEGA328P є восьми бітовим мікроконтролером на базі ядра AVR з архітектурою RISC, розроблений компанією Atmel.

Мікроконтролер працює максимальною тактовою частотою 20MHz.

Пристрій містить статичну оперативну пам'ять SRAM розміром 2KB, незалежну пам'ять EEPROM розміром 1024B, Flash пам'ять на 32KB. ATMEGA328P має шести канальний 10-bit A / D конвертер, програмований сторожовий таймер з внутрішнім генератором, двох провідний послідовний інтерфейс SPI з послідовним портом, внутрішні і зовнішні переривання, 32 регістра загального призначення, 3 гнучких таймера / лічильника з функціями порівняння.

До набору можливостей входять: 23 лінії введення / виводу загального призначення, а також програмне забезпечення з п'ятьма програмованими режимами енергозбереження.

Основні характеристики:

- Ядро: AVR;
- Ширина шини даних: 8 bit;
- Максимальна тактова частота: 20MHz;
- Розмір програмної пам'яті: 32KB;
- Розмір ОЗУ даних: 2KB;

- Розмір ПЗУ даних: 1КВ;
- Тип інтерфейсу: 2-Wire, SPI, USART;
- Максимальна робоча температура: + 85 °С;
- Мінімальна робоча температура : -40 °С;
- Кількість програмованих входів / виходів: 23;
- Тип пам'яті програм: Flash;
- Максимальна напруга живлення: 5.5В;
- Мінімальна напруга живлення: 1.8В;

## 4. РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЕКТУ

### 4.1. Обґрунтування вибору плати для проекту

Вибір плати для розробки проекту зупинився на платформі Arduino Nano. Головним конкурентом у виборі для Arduino була платформа Raspberry Pi. За тактовою частотою Raspberry Pi в 40 разів швидше, ніж Arduino. Ще більше розбіжність у оперативної пам'яті: Raspberry Pi має в 128000 разів більше оперативної пам'яті ніж Arduino.

Raspberry Pi є комп'ютером, на якому може бути запущена операційна система Linux, що підтримує багатозадачність. До USB-портів можна підключати різні пристрої, наприклад для бездротового підключення до мережі інтернет. Загалом ця крихітна плата є досить потужною і може функціонувати як повноцінний комп'ютер (хоча, звичайно з сучасними персональними комп'ютерами і ноутбуками конкурувати за потужністю Raspberry Pi не може).

Може скластися враження, що Raspberry Pi перевершує Arduino. Але ця перевага скоріше в програмному забезпеченні. Для чисто апаратних проектів набагато краще підходить Arduino. Таку перевагу Arduino має за рахунок своєї простоти.

Arduino здатна краще ніж Raspberry Pi і дійсно в реальному часі зчитувати аналогові сигнали. Ця гнучкість дозволяє Arduino працювати практично з будь-яким видом датчиків або чіпів. Raspberry Pi не така гнучка, наприклад, для читання аналогових датчиків потрібні додаткові апаратні засоби.

Arduino є менш вимогливою до живлення. Так, рекомендоване живлення для Arduino 7-12 Вольт, напруга стабілізується до 5 Вольт. А плата Raspberry Pi вимагає лише 5 Вольт на вході, тому для роботи з нею не обійтися без фільтра живлення зі струмом 1А.

Arduino може працювати з будь-яким комп'ютером і може працювати від батареї.



Arduino IDE значно простіше у використанні, ніж Linux. Наприклад, якщо ви хочете написати програму для миготіння світлодіодом з Raspberry Pi, ви повинні встановити операційну систему і деякі бібліотеки коду - і це тільки початок. Оскільки Arduino не призначений для запуску операційної системи або безлічі додатків, ви можете просто підключити плату і почати роботу.

Raspberry Pi дозволяє використовувати переваги багатозадачності, кілька програм у фоновому режимі можуть працювати одночасно. Наприклад, Raspberry Pi, може працювати і як сервер друку, і як сервер VPN одночасно.

З іншого боку, ви можете залишити Arduino, що виконує будь-який процес підключеною протягом тривалого часу, і просто відключити плату потім, коли ви вирішили. У цьому плані Arduino простіше, її складніше пошкодити.

Arduino працює з будь-яким комп'ютером, гаджетом, що використовує ANDROID – додаток. Він може працювати автономно від батареї електроживлення. Arduino можна включити і вимикати у будь-який час.

Операційна система на Raspberry Pi може бути пошкоджена, якщо відключити плату без належного завершення роботи. З Raspberry Pi ви повинні знати дещо про Linux, а також про програмному середовищі, наприклад, Python. З іншого боку, Raspberry Pi має вигоду від десятиліть програмного забезпечення Linux.

Таким чином, для даного проекту буде доцільно використовувати саме плату Arduino, використовуючи необхідні переваги. Великою перевагою є саме те, що плата Arduino може працювати з більшістю датчиків без встановлення додаткового елемента, що зменшує її розміри, підвищує практичність. Іншою досить великою перевагою, є те, що Arduino споживає набагато менше живлення для роботи і може працювати від батареї, та може зчитувати данні з датчиків в режимі реального часу, що є безперечним плюсом для проекту термометр і гігрометр. Через те, що Arduino легше програмувати, майже будь-яка людина зможе змінити код програми під свої потреби, можна додати нові

датчики, удосконалити старі. Простіше кажучи Arduino є більш гнучкою у налаштуваннях і модернізації на гаджетах, де є ANDROID – додаток.

#### **4.2. Структура та методи організації вхідних та вихідних даних**

Для отримання даних про тиск, температуру та вологість повітря обрано датчики, що описані у розд.2 та розд.4.1 (BMP 180 та датчик з чутливим елементом на гетероструктурній основі). У першого датчика є перевага - ціна. Працює по I2C інтерфейсу, завдяки чому можна використовувати із будь-якими іншими платами / контролерами, просто підключивши до необхідних пінів. Модуль вимірює значення тиску, температури та вологості повітря у реальному часі. Значення, які зафіксував модуль тиску та температури BMP 180 по безпроводній мережі передається на Android. Значення відновлюються кожні 1-5 секунди.

Дані тиску температури та вологості повітря, які представлені в дипломному проєкті відновлюються кожні чотири секунди, про що свідчить екран Android на другій сторінці додатка в четвертому рядку. Час можна змінювати, якщо потрібно - отримується значення тиску температури та вологості кожну секунду. Що дозволить більш детально вивчити зміну показників.

Якщо цей пристрій використати на квадрокоптері, який може змінювати своє положення кожну секунду то обов'язково побачимо більш різку зміну тиску, температури та вологості повітря.

Середовище розробки Arduino представляє собою текстовий редактор програмного коду, область повідомлень, вікно виведення тексту (консоль), панель інструментів і кілька меню. Для завантаження програм і зв'язку середовища розробки підключається до апаратної частини Arduino.

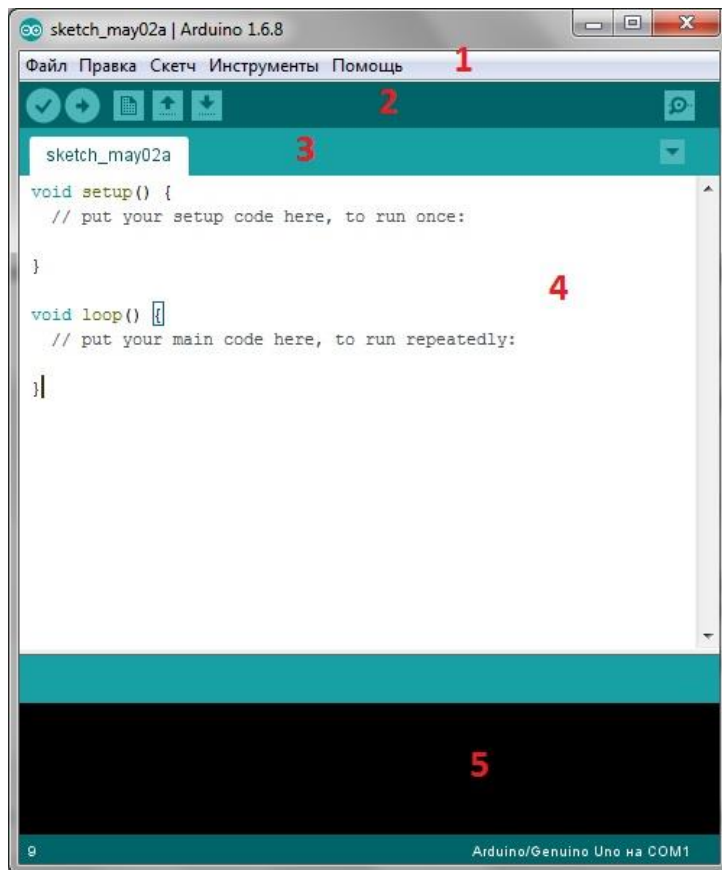


Рисунок 4.1 – Інтерфейс Arduino IDE

Меню редактора включає в себе наступні основні елементи: файл, правка, скетч, інструменти і довідка. Розглянемо докладніше кожен з них.

У меню «Файл» можна знайти команди, що відповідають за створення нової програми, читання старої, збереження її змін, а також команди для завантаження програми на мікроконтролер.

- Створити - створити нову програму (скетч);
- Відкрити - відкрити існуючу програму;
- Папка зі скетчами - відкрити програму із заданої папки;
- Приклади - відкрити приклад програми;
- Закрити – закрити поточне вікно.
- Зберегти - зберегти зміни в попередньо збереженій програмі;
- Зберегти як - зберегти нову програму, із зазначенням імені;

- Завантажити - завантажити програму в Arduino;
- Завантажити за допомогою програматора - завантажити програму

за допомогою програматора;

- Налаштування друку - налаштування принтера;
- Друк - виведення на друк коду програми;
- Налаштування – налаштування редактора;
- Вихід - вихід з Arduino IDE.

В меню «Правка» розташовані команди для роботи з кодом вашої програми. Часто використовувані команди зручні наявністю комбінацій для швидкого доступу за допомогою клавіатури. Зручними функціями є можливість копіювання для форумів і в html форматі, що дозволяють ділитися вашими скетчами, зберігаючи наочність розмітки у вигляді кодів, або html розмітки відповідно.

У меню «Скетч» розміщуються команди для контролю за процесом компіляції програми.

- Перевірити / Компілювати - компілювати програму;
- Показати папку скетчів - відкрити системну папку з програмами;
- Додати файл - додати до проекту файл з даними або програмою;
- Імпортувати бібліотеку - підключити до програми бібліотеку зі

списку встановлених.

Окремо хотілося б зупинитися на імпорті бібліотек.

Arduino IDE містить безліч попередньо бібліотек. Бібліотеки додають додаткову функціональність скетчам, наприклад, при роботі з апаратною частиною або при обробці даних. Одна або кілька директив `#include` будуть розміщені на початку коду скетчу з подальшою компіляцією бібліотек і разом зі скетчем (рис.4.2).

Завантаження бібліотек вимагає додаткового місця в пам'яті Arduino.

Для встановлення сторонніх бібліотек можна скористатися командою «Імпортувати бібліотеку».

Меню «Довідка» містить докладний опис всіх функцій самого редактора Arduino IDE, а також команди і прийоми роботи з платформою Arduino.

Пункт меню «Інструменти» включає в себе допоміжні функції для роботи з самим мікро контролером.

- Автоформатування - автоматична розстановка відступів, переносів рядків і т.п .;

- Архівувати скетч - архівація папки з програмою, і збереження архіву у вказане місце;

- виправити кодування і перезавантажити

- Монітор порту - відкрити вікно для обміну даними з мікро контролером;


- Плата - вибір поточної плати;


- Послідовний порт - вибір порту, до якого підключений пристрій;


- Програматор - вибір програм;


- Записати завантажувач - запис программи загрузчика в мікроконтролер.


## 2. Меню іконок


-  - перевірити / Компілювати програму;

-  - завантажити програму в Arduino;

-  - створити нову програму;

-  - відкрити існуючу програму;

-  - зберегти програму;

-  - монітор послідовного порту.

## 3. Вкладки

Кожна програма для Arduino може складатися з декількох файлів. Для перемикання між цими файлами служить система вкладок у редакторі. Також можна створити нову вкладку, і асоціювати з нею файл в папці з проектом.

## 4. Вікно програми

Безпосередньо, текст програми створюється і редагується в головному

вікні редактора. По суті, вікно редактора являє собою типовий текстовий редактор, з підвічуванням конструкцій коду.

## 5. Вікно повідомлень

У самому низу редактора Arduino IDE є невелике вікно, що служить для виведення повідомлень про проблеми, які виникають в процесі. Нами використувалася також Microsoft Visual Studio — серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.

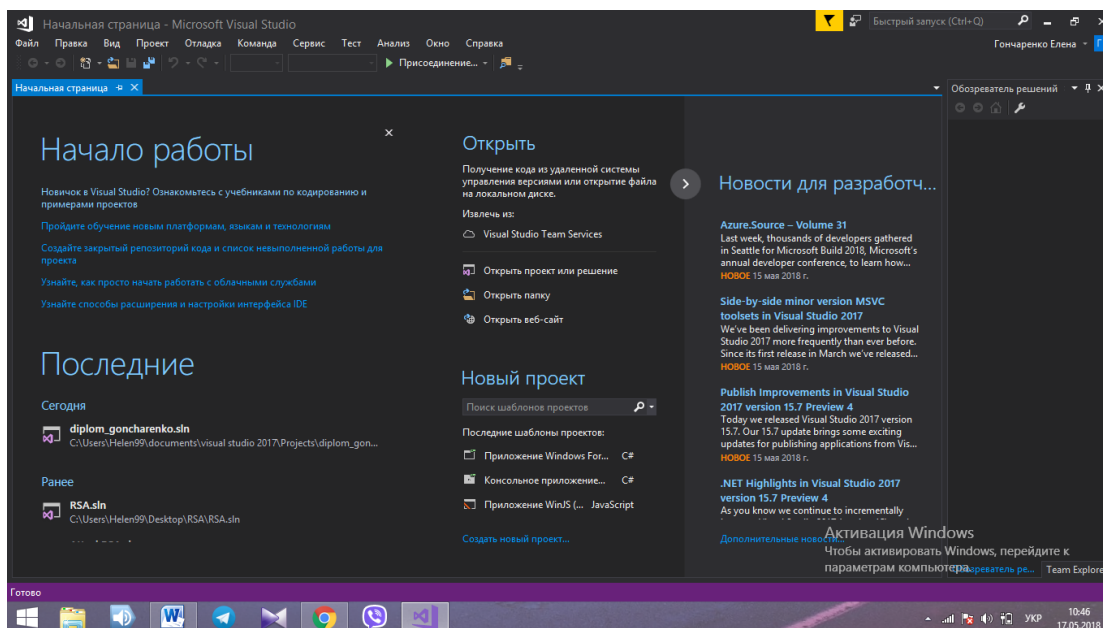


Рисунок 4.2 – Интерфейс Visual Studio

Windows Forms - це компонент інтелектуальних клієнтів .NET Framework, набір керованих бібліотек для виконання поширених завдань, таких як

читання і запис в файлової системі. За допомогою середовища розробки, як Visual Studio, можна створювати додатки Windows Forms, які відображають відомості, запитують введення від користувачів і обмінюються даними з віддаленими комп'ютерами по мережі.

У Windows Forms форми - це видима поверхня, на якій відображається інформація для користувача. Ви часто Windows Forms будується шляхом приміщення елементів управління у формах і написання коду для реагування на дії користувача, такі як клацання миші або натискання клавіш. Елемент управління - це окремий елемент призначеного для користувача інтерфейсу, призначений для відображення або введення даних.

Windows Forms включає широкий набір елементів управління, які можна розміщувати на формах: елементи управління, що відображають текстові поля, кнопки, списки, що розкриваються, перемикачі та навіть веб-сторінки. Список всіх елементів управління, які можна використовувати в формі, представлені в розділі Елементи керування для використання в формах Windows Forms. Якщо існуючий елемент управління не задовольняє потребам, в Windows Forms можна створювати власні елементи управління за допомогою класу UserControl.

До складу Windows Forms входять багатофункціональні елементи призначеного для користувача інтерфейсу, що дозволяють відтворювати можливості таких складних додатків, як Microsoft Office. За допомогою ToolStrip і MenuStrip елемента управління, можна створити панелі інструментів і меню, що містять текст і малюнки, підміню і інші елементи управління, такі як текстові поля і поля зі списком.

За допомогою конструктора і перетягнись форм Visual Studio можна легко створювати додатки Windows Forms: досить виділити елемент керування курсором і помістити їх в потрібне місце на формі. Конструктор надає кошти, такі як лінії сітки і «лінії прив'язки» для подолання труднощів вирівнювання елементів управління. За допомогою Visual Studio або компіляції з командного

рядка, можна використовувати `FlowLayoutPanel`, `TableLayoutPanel` і `SplitContainer` елементи управління для створення складних макетів з мінімальними витратами часу і зусиль форм.

У багатьох додатках потрібно відобразити дані з бази даних, XML-файла, веб-служби XML або іншого джерела даних. Windows Forms надає гнучкий елемент управління називається `DataGridView` елемента управління для відображення таких табличних даних в традиційному форматі рядків і стовпців, щоб кожен фрагмент даних займає свою власну клітинку. За допомогою `DataGridView` можна налаштувати зовнішній вигляд окремих осередків, зафіксувати рядки і стовпці на своєму місці і відображення складних елементів управління всередині осередків, крім іншого.

При використанні інтелектуальних клієнтів Windows Forms можна легко підключатися до джерел даних по мережі. Компонент Windows Forms `BindingSource`, що з'явився в Visual Studio 2005 і .NET Framework 2.0, являє підключення до джерела даних і містить методи для прив'язки даних до елементів управління, переходу до попередньої або наступної записи, редагування записів і збереження змін у вихідному джерелі.

Елемент управління `BindingNavigator` надає простий інтерфейс на основі компонента `BindingSource` для переходу між записами.

Мовою програмування було обрано C# — об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft Research (при фірмі Microsoft). C++ - це мова програмування високого рівня з підтримкою декількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Розроблена Б'ярном Страуструпом (англ. Bjarne Stroustrup) в AT&T Bell Laboratories (Мюррей-Хілл, Нью-Джерсі) у 1979. Базується на мові C.

У 1990-х роках C++ стала однією з найуживаніших мов програмування загального призначення. Мову використовують для системного



програмування, розробки програмного забезпечення, написання драйверів, потужних серверних та клієнтських програм, а також для розробки розважальних програм таких як відеоігри. С++ суттєво вплинула на інші, популярні сьогодні, мови програмування: С# та Java.

С # є об'єктно-орієнтованою мовою, але підтримує також і компонентно-орієнтоване програмування. Розробка сучасних додатків все більше тяжіє до створення програмних компонентів у формі автономних і самоопісательних пакетів, що реалізують окремі функціональні можливості. Важлива особливість таких компонентів - це модель програмування на основі властивостей, методів і подій. Кожен компонент має атрибути, які надають декларативні відомості про компоненті, а також вбудовані елементи документації. С # надає мовні конструкції, безпосередньо підтримують таку концепцію роботи. Завдяки цьому С # відмінно підходить для створення і застосування програмних компонентів.

С# безпечніший в порівнянні з С++. Єдиними неявними перетвореннями за замовчуванням є ті, які вважаються безпечними, наприклад, розширення цілих чисел. Це застосовується під час компіляції, під час JIT і, в деяких випадках, під час виконання. Не відбувається неявних перетворень між булевими і цілими числами, а також між членами перерахування і цілими числами (крім літерала 0, який може бути неявно перетворений в будь-який нумерований тип).

Будь-яке призначене для користувача перетворення повинно бути явно позначене як явне або неявне, на відміну від конструкторів копіювання С++ і операторів перетворення, які за умовчанням є неявними.

С# має явну підтримку коварианції та контраваріантності в родових типах, на відміну від С++, яка має певний рівень підтримки контраваріантності просто через семантику типів, що повертаються, на віртуальні методи.

Незважаючи на технологічний прогрес, більшість сучасних комп'ютерів побудовано за тими самими принципами, що й обчислювальні машини

40-х рр. XX ст. В їх основі лежить так звана архітектура фон Неймана (за ім'ям видатного американського вченого, який першим сформулював головні засади архітектури електронних обчислювальних машин).

Система команд, які може виконувати процесор, називається машинною мовою. Для людини машинні мови дуже незручні, вони вимагають глибоких знань про устрій вузлів комп'ютера й подробиці виконання програми. Цими мовами користуються розробники комп'ютерів і деякі інші спеціалісти.

### **4.3. Плата призначена для управління пристроями**

Плата Arduino nano v 3.0 використана на мікроконтролері Mega328p з метою використання ANDROID – додатку. Плата призначена для електронного управління пристроями, розробки електронних схем, що вимагають участі мікроконтролерів і вивчення принципів роботи мікроконтролерів в цілому.

Процес підготовки до налагодження роботи плати може бути значно спрощеним завдяки тому, що до всіх контактів на платі (де це передбачено) припаяні штирові лінійки. Шляхом використання оригінального скетчу, після монтування плати у макет отримані необхідні результати.

Параметри Arduino nano v 3.0:

- мікроконтролер Mega328p
- Робоча напруга (логічний рівень) 5В
- Напруга живлення (рекомендований) 7-12В
- Напруга живлення (граничне) 6-20В
- Підключення до ПК: mini USB
- Цифрові входи / виходи 14
- Аналогові входи 8
- Тактова частота: 16 МГц
- Розміри плати: 1.85 см x 4.3 см
- Максимальний струм одного виведення 40 мА

- Flash-пам'ять 32 КБ (з них 2 КБ під завантажувач)
- SRAM: 2 КБ
- EEPROM: 1 КБ

На рисунку 4.3 наведена основна складова проекту, де детально зображено плату Arduino nano 3.0 що слугує основним елементом програмування датчика, що обробляє данні навколишнього середовища у цифровій формі.



Рисунок 4.3 - Плата Arduino nano 3.0 з контактами

Плата має наступний набір основних показників котрі слугують для виконання певної кількості завдання та процедур.

Характеристики:

- Напруга живлення: 3.3 або 5 Вольт
- Струм: 5 мкА при швидкості опитування 1 Герц
- Струм: 0.1мкА в режимі очікування
- Діапазон вимірювання тиску: від 300 до 1100 гПа (від -500 від +9000 метрів над рівнем моря)
- Шум: 0.06 гПа (0.5м) в грубому режимі (ultra low power mode) і 0.02 гПа (0.17м) в режимі максимального дозволу (advanced resolution mode)
- Діапазон вимірювання температури: від 0 до 65 ° C

- Точність вимірювання температури:  $\pm 2^{\circ}\text{C}$
- Роздільна здатність: 16 біт
- Максимальна швидкість інтерфейсу: 3,4 Мбод
- Робоча температура: від  $-40^{\circ}\text{C}$  ~  $+85^{\circ}\text{C}$
- Діаметр монтажного отвору: 3 мм
- Розміри: 12 x 10 мм
- Вага: 1.18 гр.

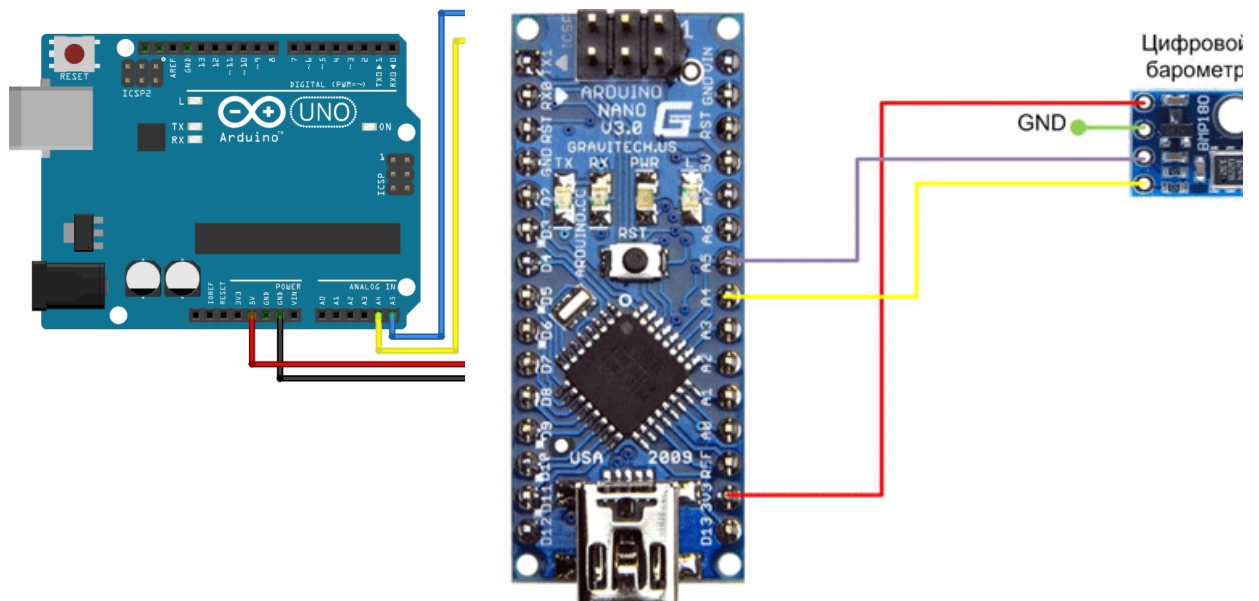


Рисунок 4.4 – Схема підключення барометричного датчика до плати

А) Arduino-Uno

Б) Arduino nano 3.0

Оскільки Arduino - це серце конструктора, де немає кінцевого, певного набору деталей, і немає обмежень в розмаїтті того, що можна зібрати. Все обмежено лише вашою фантазією. Arduino - це не просто плата з мікро контролером, а електронний конструктор, який виявиться знахідкою і для інженера-

електронщика і для дилетанта-радіоаматора. Arduino - це серія мікроконтролерів, з якими впорається навіть будь-хто.

Відмінний варіант для навчання. Всі мікроконтролери Arduino розташовані на отладоночній платі, яка, в свою чергу, може мати на борту сам мікроконтролер, перетворювач USB-UART, USB-порт, роз'єми аналогових і цифрових входів / виходів і багато іншого.

Прошиваються мікроконтролери Arduino, найчастіше, через USB, хоча, можлива заливка скетчу і внутрісхемний через програматор (рис.4.4 б).

Основою цього модуля є новітня розробка фірми BOSCH - інтегральний п'єзорезистивного датчик тиску і температури BMP180, виготовлений по мікро-технології. Кожен датчик індивідуально калібрується на заводі.

Вбудований термометр і калібрувальний коефіцієнти, що зберігаються в датчику, дозволяють отримати характеристики для такого класу пристроїв.

Датчик має високу абсолютну точність і впевнено реєструє зміна тиску атмосфери при переміщенні по висоті на 0,2..0,3м.

Модуль можна застосувати в погодніх станціях, в висотомір і датчиках вертикальної швидкості літальних апаратів, для контролю тиску в медицині та системах вентиляції.

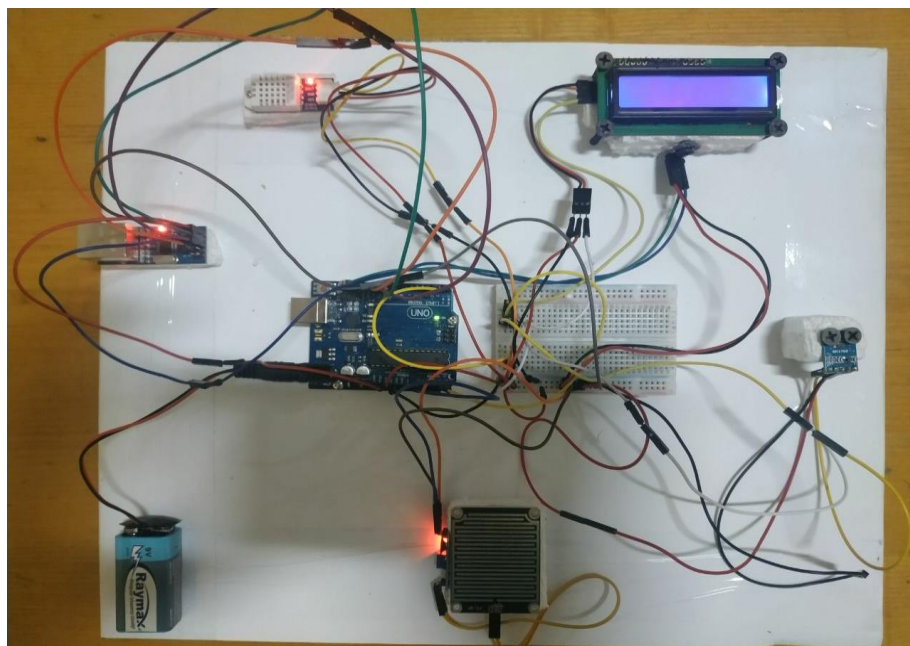


Рисунок 4.5 – Макет підключення барометричного датчика, датчика вологості та температури до плати Arduino nano 3.0

**Додаток Analyzer.** Створивши повноцінний датчик для збору таких показників як тиск, температура, вологість була проведена розробка додатка для використання у гаджетах з ANDROID-платформою.

Запропонована нами форма є зручною і дозволяє ознайомити користувача з зібраними показниками, надавши можливість записати інформацію. А також вивести інформацію на дисплей, або зберегти на зовнішньому носії з використанням платформи ANDROID.

Водночас з Android studio, як додаток на мобільний пристрій, високу ефективність демонструє додаток на платформі Xamarin. Останній надає більш витончений дизайн та інтерфейс у роботі. Xamarin дозволяє будувати мобільні додатки для усіх основних платформ зі спільною базою коду, що значно зменшує витрати часу і коштів. Такі додатки не просто виглядають добре (згідно з принципами дизайну кожної платформи), але також працюють ефективно. Для них є доступними усі API. І розробники, і користувачі можуть користуватися унікальними можливостями, які пропонує кожна платформа. Спільну мову між різними платформами було визначено. Працюючи з Xamarin, достатньо знати C# — мову, звичну для усіх розробників на .NET. В іншому випадку девелоперам нативних додатків під Android та iOS потрібно було б знати Java і Objective-C або Swift. Для цього, швидше за все, потрібні були б різні команди. А з Xamarin усе простіше. Для спільного коду в Xamarin існують спеціальні типи проєктів, такі як Shared Projects та Portable Class Libraries (PCLs), які в ході компіляції автоматично додаються до відповідних мобільних додатків. Також Xamarin надає змогу використовувати всі стандартні бібліотеки та технології .NET'у, що добре відомі кожному фахівцеві. Між іншим,

формуючи цей проект постало питання з'єднання датчиків з елементами програмної частини - ANDROID – додатку.

Отже, для підключення було обрано протокол типу TCP/IP що є досить зручним у роботі і доволі швидко під'єднує до локальної мережі всі потрібні пристрої, а саме за допомогою датчика тиску, вологості, або іншого чутливого елемента.

#### **4.4.Робота з використанням платформи ANDROID**

Наведемо далі послідовність виконання дій щодо отримання необхідної інформації з використанням платформи ANDROID.

Процес підключення відбувається в автоматичній формі: для цього потрібно натиснути на початковому екрані «почати роботу», і підключення відбувається автоматично та перейде до наступної вкладки.

Коли вже під'єднання відбулося зустрічаємо наступну сторінку, що дозволяє налаштувати процес зчитування та формування показників, є можливість налаштування інтервалу зчитування даних таких як: тиск; температура; вологість повітря у відсотках.

Додаток фіксує значення температури, що споживач отримує через бездротову мережу і виводить інформацію на дисплей гаджету на платформі ANDROID, в межах заданого часу. Ми бачимо значення температури за кожні 4 секунди, які виставили на початку. Відкриваючи наступну сторінку бачимо значення тиску, що також замірюються кожні чотири секунди рисунок та формують вже історію зчитування даних.

Далі є можливість контролю вхідних даних температур, що так само записуються у певний проміжок часу (рис 4.9).

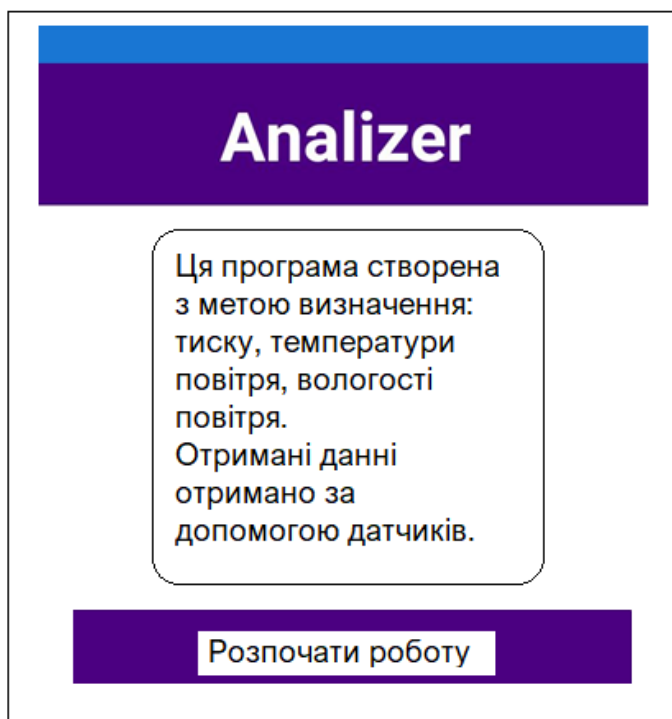


Рисунок 4.6 – Початок роботи (підключення до датчика)

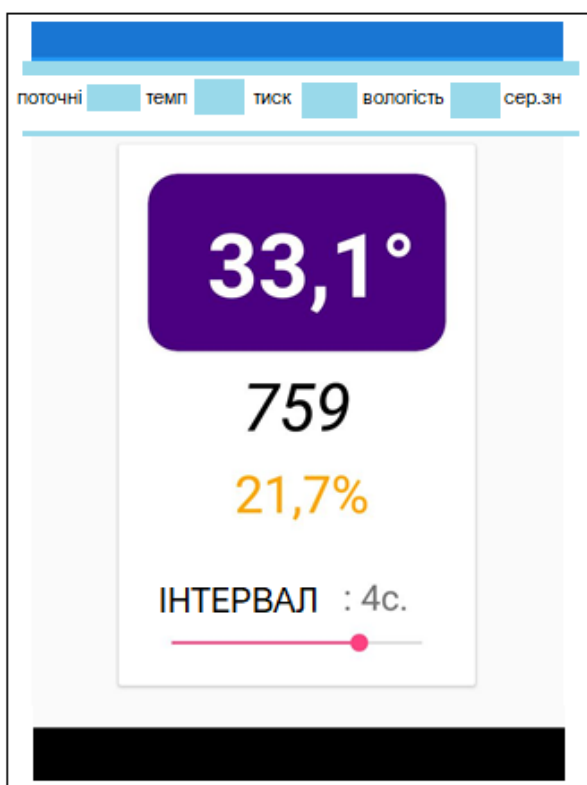


Рисунок 4.7 – Налаштування збору даних



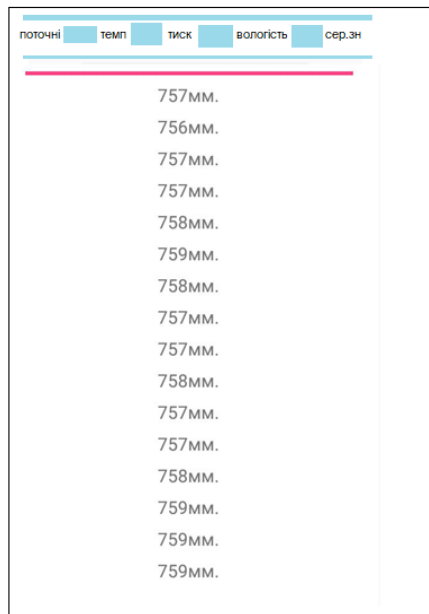


Рисунок 4.8 – Історія зчитаного тиску за останній проміжок часу

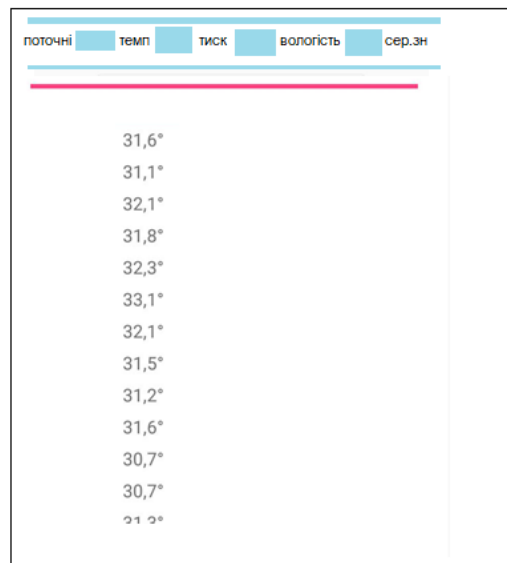


Рисунок 4.9 – Історія зчитаного температури за останній проміжок часу

Усі зібрані данні є високого рівня, адже застосування інтелектуального датчику дозволяє зменшити похибку вимірювання до мінімуму. А вбудований термометр і калібрувальні коефіцієнти, що зберігаються в датчику, дозволяють отримати недосяжні раніше, для такого класу пристроїв, характеристики.

Датчик має високу абсолютну точність і впевнено реєструє зміна тиску атмосфери при переміщенні по висоті на 0,2..0,3 м. Запропонований апартний модуль можна застосувати у метеорологічних станціях, у висотомірах та датчиках вертикальної швидкості літальних апаратів, для контролю тиску в медицині та системах вентиляції. Знімаючи навіть вологість повітря було отримано досить якісні показники.

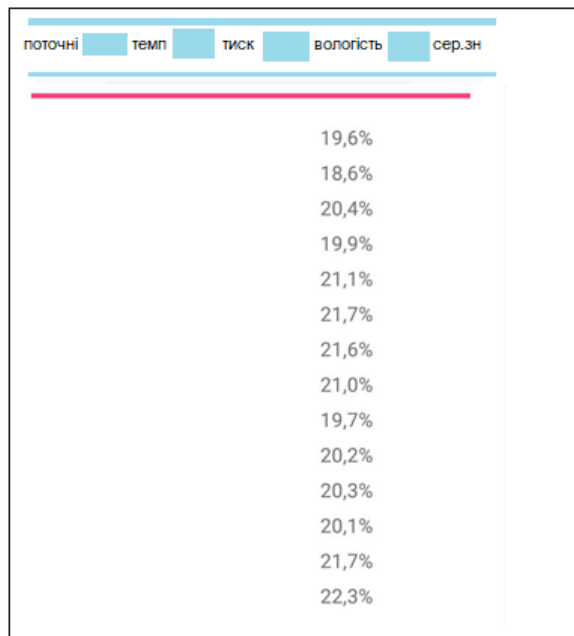


Рисунок 4.10 – Історія зчитаного вологсті повітря за останній проміжок часу у відсотковому співвідношені

Середні значення параметрів, які отримали з датчиків виводяться на скріншот. Це дає можливість визначити більш точні показники, адже електроніка має завжди похибку. Отримавши середнє значення маємо можливість отримувати більш якісні дані та формувати на їх базі необхідні метеорологічні дослідження.

До запропонованої платформи був підключений дисплей, що дозволило отримувати значення метеорологічних характеристик у цифровому вигляді (рис.4.11).

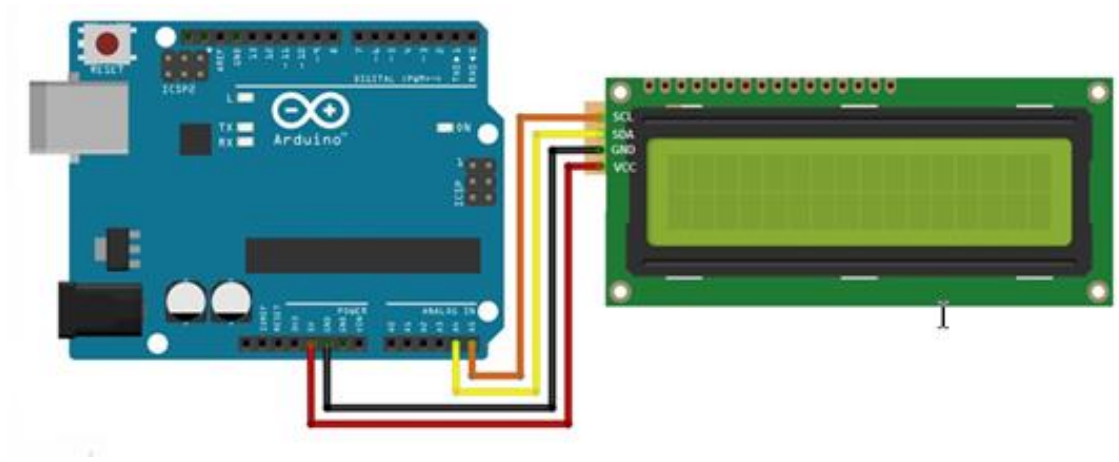


Рисунок 4.11 – Підключення дисплею до плати

## ВИСНОВКИ

Основною задачею дипломного проекту стала розробка пристрою на базі плати ARDUINO (для ANDROID платформи), а також додатку, що дасть змогу аналізувати данні тиску, температури та вологості повітря в реальному часі.

Були вирішені наступні задачі.

1. Проведено аналітичний огляд та розглянуто коло задач, що вирішують сучасні інформаційно-вимірювальні системи (розділ 1).
2. Проведена практична робота щодо розробки чутливого елемента інтелектуального сенсора для створення датчика тиску та інших метеорологічних характеристик (розділ 2).
3. Проаналізовано можливості апаратно обчислювальної платформи ARDUINO та визначені сфери її застосування для гаджетів з ANDROID-процесором (розділ 3).
4. Виокремлено основні компоненти апаратно обчислювальної платформи ARDUINO та створено програмне середовище для гаджетів з ANDROID-процесором (розділ 4).
5. Доведено до практичної реалізації пристрій, що дозволяє виміряти метеорологічні характеристики (тиск, температура, вологість повітря). Зроблено його зручним у використанні для гаджетів з ANDROID-процесором, завдяки гнучкій системи налаштування (що дозволить змінювати проміжок часу між даними) (Додаток).

**Практичне** значення результатів дослідження полягають у тому, що результати аналізу обмежень та погрешностей вимірювання метеорологічних характеристик (тиску, температури, вологості повітря) можуть проведені на основі схеми кодування з використанням апаратно-обчислювальної платформи ARDUINO для гаджетів з ANDROID-процесором.

## Додаток

Для повноцінного запуску необхідно провести завантаження скетча у пам'ять мікроконтролера для подальшого отримання даних для гаджетів з ANDROID-процесором.

В Arduino IDE завантажувався скетч, що наведений у цьому додатку.

```
//ПІДКЛЮЧЕННЯ БІБЛІОТЕК
#include <DHT.h> //Бібліотека датчика температури та вологості DHT22
#include <ETHER_28J60.h> // Бібліотека Ethernet модуля ENC28J60
#include <Wire.h> // Бібліотека для взаємодії з різними пристроями по інтерфейсу
I2C
#include <LiquidCrystal_I2C.h> // Бібліотека для роботи с LCD дисплеєм
#include <SFE_BMP180.h> // Бібліотека цифрового барометру BMP180
#include <BH1750.h> // Бібліотека датчика світла BH1750
//=====

//ВИПРАВЛЕННЯ ПОХИБОК ВИВЕДЕННЯ НА ДИСПЛЕЇ
//Код дозволяє виведення на дисплеї строку цілою, а не лише окремий символ
#if defined(ARDUINO) && ARDUINO >= 100
#include "Arduino.h"
#define printIIC(args) Wire.write(args)
inline size_t LiquidCrystal_I2C::write(uint8_t value) {
    send(value, Rs);
    return 1;
}
#else
#include "WProgram.h"
#define printIIC(args) Wire.send(args)
inline void LiquidCrystal_I2C::write(uint8_t value) {
    send(value, Rs);
}
#endif
//=====
```

```

//ІДЕНТИФІКАТОРИ І ГЛОБАЛЬНІ ЗМІННІ
#define DHTPIN 2 // номер цифрового піна, до якого під'єднано датчик Темпера-
тури та вологості
#define rainPin A0 //номер аналогового піна, до якого під'єднано датчик Вологи і дощу
static uint8_t mac[] = { 0x54,0x55,0x58,0x10,0x00,0x24 }; //задаємо мак адрес при-
строю
static uint8_t ip[4] = { 192, 168, 1, 155}; //надаємо статичну адресу
static uint8_t port = 80; //надаємо порт для підключення
int timer = 0; //змінна для визначення затримки між розрахунками
//=====================================================

//НАЗВА ОБ'ЄКТІВ КЛАСІВ, ЩО ВИКОРИСТОВУЮТЬ ДАТЧИКИ І МОДУЛІ
//Датчик температури і вологості
//створюємо об'єкт класу DHT
//передаємо номер піна до якого підключений датчик і тип датчика
//типи сенсорів: DHT11, DHT21, DHT22
DHT dht(DHTPIN, DHT22);

//Ethernet модуль
//створюємо об'єкт класу ETHER_28J60
ETHER_28J60 client;

//LCD дисплей
//створюємо об'єкт класу LiquidCrystal_I2C
//показуємо I2C адрес LCD дисплея, а також параметри екрана
//у випадку LCD 1602 – 2 строки по 16 символів у кожній
LiquidCrystal_I2C lcd(0x27,16,2);

//Датчик тиску
//створюємо об'єкт класу SFE_BMP180
SFE_BMP180 pressure;

//Датчик світла

```

```

// створюємо об'єкт класа BH1750
BH1750 lightMeter;

//=====================================================

//ІНІЦІАЛІЗАЦІЯ СИСТЕМИ
void setup()
{
  //Ініціалізуємо спілкування з комп'ютером за послідовним інтерфейсом
  //задаємо швидкість передачі даних у бітах за секунду
  Serial.begin(9600);

  // Ініціалізуємо датчик Температури і вологості
  dht.begin();

  //Запускаємо Сервер
  client.setup(mac, ip, port);

  // Ініціалізуємо дисплей
  //вмикаємо підсвітку
  //встановлюємо курсор на початок першої строки
  //набираємо текст
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("(c) OSENU 2019");
  //Встановлюємо режим роботи заданного pin датчика Вологи і дощу, як входу
  pinMode(rainPin, INPUT);
  // Ініціалізуємо датчик Освітленості
  lightMeter.begin();

  // Ініціалізуємо Барометр
  pressure.begin();
}

//=====================================================

```

```

//ОСНОВНИЙ ЦИКЛ
void loop()
{

// Затримка 1/1000 секунди між вимірами
delay(1);

//Збільшення значення таймеру
timer++;

//Локальні змінні
uint16_t lux; //освітленність
int rainValue; //датчик дощу
float h; //вологість
double t; //температура
double P; //тиск
char* rain; //показання датчика дощу
char light; //показання датчика освітленості

//Вимірюємо вологість
h = dht.readHumidity();
//Изверяю температуру
t = dht.readTemperature();

//Спрацьовує раз у 2-4 секунди
if(timer==2000)
{
//Зчитуємо показник з датчика Дощу і вологості
rainValue = analogRead(rainPin);
//Обробляємо отримані показники
if(rainValue <= 500)
if(t<=0)

```



```

    rain = "Snow";
else
    rain = "Rain";
else
    rain = "Dry";

// Зчитуємо показник з датчика Освітленості
lux = lightMeter.readLightLevel();
// Обробляємо отримані показники
if(lux<=10)
    light = 'N';
else
    if(lux<60 && lux>10 )
        light = 'C';
    else
        light = 'S';

char status; //змінна для обробки статусу Барометра
//Зчитуємо показники Барометра
status = pressure.startPressure(3);
if (status != 0)
{
    //чекаємо на закінчення вимірювань:
    delay(status);
    //Тепер отримуємо тиск у змінну P
    //функція повертає 1 якщо все ОК, 0 якщо не ОК.
    status = pressure.getPressure(P,t);
}

//Виведення показань на екран
byte          simvol[8]
{B11100,B10100,B11100,B00000,B00000,B00000,B00000,B00000,}; // Символ градуса
lcd.clear(); //очистка екрана
lcd.setCursor(0,0); //установка курсора на початок першої строки

```

```

        lcd.print("T:"+(!isnan(t)?((String)(int)t+ "\1C"): "Err ") + " H:"+(!is-
nan(h)?((String)(int)h+ "% "): "Err ") + light); //друкуємо строку показників
        lcd.createChar(1, simvol); //змінюємо символ градуса
        lcd.setCursor(0,1); //установка курсора на початок другої строки
        lcd.print((String)"P:"+(!isnan(P)?((String)(int)P+ "mbar "): "Err ") + rain); //друкуємо
строку показників

```

```

//Виводимо показники по Serial
Serial.println("Temperature: "+(!isnan(t)?((String)t+ " C"): "Error"));
Serial.println("Humidity: "+(!isnan(h)?((String)(int)h+ " % "): "Error"));
Serial.println("Pressure: "+ (!isnan(P)?((String)(int)P+ " mbar"): "Error"));
Serial.println("Light: " + " +
((light=='S')?(String)"Sun":((light=='C')?(String)"Cloudy":(String)"Night")) + " (" + lux + " lux)");
Serial.println("Rain: " + (String)rain);

```

```

//Обнуляємо счетчик таймера
timer = 0;
}
//Виведення даних на Web-сторінку
//отримання запиту до сервера
if (client.serviceRequest())
{
//Побудова Web-сторінки з автооновленням раз у 4 секунди
client.print("<html>");
client.print("<head><title>OSENU Wether Station</title><meta http-equiv=\"re-
fresh\" content=\"4\" ></head>");
client.print("<h2>OSENU Wether Station</h2></br>");
client.print("<b>Temperature: ");
client.print(t);
client.print(" &deg;C </br>");
client.print("Humidity: ");
client.print(h);
client.print(" %</br>");
client.print("Pressure: ");

```

```
client.print(P);
client.print(" mbar</br>");
client.print("Light: ");
if(light=='S')
    client.print("Sun");
else if(light=='C')
    client.print("Cloudy");
else
    client.print("Night");
client.print(" (");
client.print(lux);
client.print(" lux)</br>");
client.print("Rain: ");
if(rain=="Rain")
    client.print("Rain");
if(rain=="Snow")
    client.print("Snow");
if(rain=="Dry")
    client.print("Dry");
client.print(" </b></br>");
client.print("</html>");
client.respond();
}
}
```

## Література

1. Голіцина, О.Л. Інформаційні технології: Підручник / О.Л. Голіцина, Н.В. Максимов, Т.Л. Партика, І.І. Попов. - К .: Дім, ІНФРА-К, 2018. – 608 с.
2. Голубенко, Н.Б. Інформаційні технології в освітній сфері / Н.Б. Голубенко. - К: Фенікс, 2020. - 282 с.
3. Ковальчук В.В., Громов В.А., Маслий О.Н. Физические основы полупроводниковой электроники. – Одесса: «Видавець Букаєв Вадим Вікторович», 2012. – 180 с.
4. Находкін М.Г., Сізов Ф.Ф. Елементи функціональної електроніки. - К.: ВД «Професіонал», 2012. – 201 с.
5. Ковальчук В.В., Міроненко С.В., Смерж М.В. , Заїка Г.В. Управління якістю тепловізійних зображень // 7-ма Міжнародна НПК «Інтегровані інтелектуальні робототехнічні комплекси» (ІРТК-20194). Київ, 19-20 травня, 2019. – С.148-149.
6. Ковальчук В.В., Моисеев Л.М. Мезомеханіка регулярного деформаційного рельєфу // Наукове видання / Під ред. Проф. Лашкарьова Г.В. К.: Професіонал, 2004. - 275 с.
7. Ковальчук В.В., Костенко В.Л., Жаровцев С.О., Панченко О.О. Сенсорні пристрої та датчики // Наукове видання Одеса: видавець Букаєв Вадим Вікторович, 2013. – 200 с.
8. Ковальчук В.В., Смерж М.В. Твердотельный фото датчик для информационно-измерительных систем // Вестник ИАН Украины. – 2015. – № 3. – С.120-125.
9. Kovalchuk V.V. Nanoelectronic's material for optic system // Photoelectronics. – 2012. – № 21.P.65 – 68.
- 10.Лаздіна С.І., Лаздін В.П., Голомовзий Є.М., Олексієнко К.М. Температурні та теплофізичні вимірювання. м. Одеса: Стандартъ, 2007. – 223 с.
- 11.Ковальчук В.В., Кластерная модификация полупроводниковых гетеро структур // Наукове видання К.: Ні-Tech., 2007. – 309 с.

12. Характеристики та робота платформи <http://Arduino-project.net> 30.03.18 р.
13. Блум Дж. Вивчаємо Arduino [пер. з англ. Петін В.] - К.: БХВ- 2019.336 с.
14. Еванс Б. Arduino. Блокнот програміста. [пер. з англ. Голобов В.] – К: Освіта, 2017. 40 с.
15. Платформа Arduino <http://Arduino-diy.com> 25.04.2021.
16. Современные операционные системы, Таненбаум Э., Бос Х., К.: ЛОСЬ, 2020. – 221 с.
17. Ковальчук В.В., Панченко А.А., Сморгж М.В. Фізична модель чутливого елементу термометрів опору// Вісник Інженерної академії України. – 2015. – № 4.– С.111-115.
18. Ковальчук В.В., Громов В.О., Коломієць Б.С. Вимірювання метеорологічних характеристик з використанням апаратно-обчислювальної платформи ARDUINO // Матеріали ІХ Міжнародної науково-практичної конференції «Study of world opinion regarding the development of science», 22-25 листопада 2022 р., Прага, Чехія.
19. Gerz E. di Justo P. Atmospheric Monitoring with Arduino - O'Reilly Media, Inc. 2013 89с.
20. Шишкіна Ю.М Технології в управлінні підприємством Актуальні питання економічних наук. 2021. № 17-2. С. 263-265
21. Доступ до навчальних ресурсів <http://window.edu.ru/> 12.04.2021.
22. Навчальні конструктори <http://hobbytech.com.ua> 14.04.02022.
23. Механіка інтернет магазин <http://robom.ru> 15.04.2021.
24. Амперка <http://wiki.amperka.ru> 20.04.2021.
25. Програмування платформи <http://forum.Arduino.ua> 12.04.2021.
26. Стартові набори Arduino [http:// Arduino-diy.com](http://Arduino-diy.com) 10.04.2021.
27. Каталог енциклопедій. [www.dic.academic.ru](http://www.dic.academic.ru) 11.05.2021.
28. Архивы нормативно-технической, проектно-конструкторской, финансовой и др. документации. [www.alee.ru](http://www.alee.ru) 20.04.2021.