Київський національний університет імені Тараса Шевченка

Факультет радіофізики, електроніки та комп’ютерних систем

Кафедра радіотехніки та радіоелектронних систем

**КУРСОВИЙ ПРОЕКТ**

**з цифрової схемотехніки та радіоелектроніки**

**Програмно-апаратний комплекс для вимірювання відносного рівня СО**

**у видиху людини**

|  |
| --- |
| Виконавець:  студент групи ІБТКСМ (ПАЗТК)  (3 курс, бакалавр)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ігор ЯШАН |
| Науковий керівник:  к.ф.-м.н., доцент кафедри радіотехніки та радіоелектронних систем, доцент  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ігор БЕХ |
|  |
| Завідувач кафедри радіотехніки та радіоелектронних систем,  д.ф.-м.н., професор  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ігор АНІСІМОВ |

Київ-2021

ЗМІСТ

|  |  |
| --- | --- |
| ВСТУП | 3 |
| РОЗДІЛ 1 ОГЛЯД ЛІТЕРАТУРИ | 5 |
| 1.1. Актуальність діагностики зовнішнього дихання | 5 |
| 1.2.Компонентна база електроніки для приладу діагностики зовнішнього дихання | 6 |
| 1.2.1. Мікроконтролер STM32 | 6 |
| 1.2.2. Датчик СО MQ7 | 12 |
| 1.2.3. Модуль LCD MAX7219 | 13 |
| РОЗДІЛ 2 ПРОГРАМНО-АПАРАТНИЙ КОМПЛЕКС ДЛЯ ВИМІРЮВАННЯ ВІДНОСНОЇ КОНЦЕНТРАЦІЇ ОКИСУ ВУГЛЕЦЮ У ВИДИХУ | 16 |
| 2.1. Структурна схема пристрою | 16 |
| РОЗДІЛ 3 ПРОГРАМНИЙ КОД ДЛЯ STM32F103 | 17 |
| 3.1. Середовище розробки | 17 |
| 3.1.1. STM32CubeProgrammer | 17 |
| 3.1.2. STM32CubeMX | 17 |
| 3.1.3. Keil uVision5 | 19 |
| 3.1.4. Створення проекту та кодування алгоритму дії аналізатора СО | 20 |
| ВИСНОВКИ | 23 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 24 |

**Вступ**

В сучасному світі дуже важливо слідкувати за станом здоров`я людини для високої працездатності та збільшення тривалості життя. І на допомогу нам в побутовому житті приходить багато різноманітних приладів для моніторингу показників життєдіяльності людини. Наприклад пульсометри, тонометри, глюкометри та різні аналізатори газів в організмі людини.

Аналіз повітря дихання пацієнта відноситься до неінвазивних методів діагностики, що й викликає підвищений інтерес. З розвитком інформаційних технологій стало легше створювати нові апаратні засоби та методики визначення процентного вмісту різних газів. Один з способів діагностики є аналіз складу видихуваного повітря, яке представляє собою суміш газів різного ендогенного походження з дихальних шляхів. Компоненти видихуваного повітря можуть свідчити про стан здоров’я людини. Для того щоб якось можна було фіксувати та аналізувати маркери певних газів, зокрема, моно оксиду вуглецю, створені різноманітні пристрої для діагностики зовнішнього дихання. Загалом, розробка апаратури і методик для віддаленого контролю стану здоров’я людини на основі мікропроцесорної техніки сприяє підвищенню ефективності діагностики і є актуальним напрямком у медицині.

Наприклад, у випадках хронічної обструкції легень чи ревматоїдного артриту, куріння хворого є фактором, що впливає на результат проведених аналізів. Тому перед взяттям проби хворих просять утриматися від куріння на термін 8 годин. Але, як показала статистика, частина пацієнтів ігнорує попередження, що впливає на результати. Тому важливо мати прилади, що можуть встановити, чи палили пацієнти перед проведенням процедури.

Ще один приклад використання датчиків для виявлення чадного газу у диханні був представлений в науковій статті «A mobile-phone-based breath carbon monoxide meter to detect cigarette smoking». У ній автор пропонує використовувати сенсори для вимірювання рівня CO у мобільних смартфонах та персональних комп’ютерах. А саме вводити їх в конструкцію гаджетів. Важливою особливістю цих «втручань» є об'єктивна оцінка стану курця, через частий контроль вмісту чадного газу в диханні. Необхідна об’єктивна оцінка куріння, оскільки курці часто помилково класифікують себе як некурців під час спроб кинути палити. Крім того, деякі дані свідчать про те, що лише моніторинг CO в диханні може сприяти зменшенню куріння [7].

Необхідно розробити портативний прилад, який буде мати досить хорошу чутливість для аналізу видиху людини і міг би показати коректні данні, які б показували рівень отруєння людини монооксидом вуглецю.

Враховуючи все вищезазначене при виконанні даного курсового проекту я зосередився на розробці пристрою для вимірювання відносного рівня монооксиду вуглецю у видиху людини.

**РОЗДІЛ 1**

* 1. **АКТУАЛЬНІСТЬ ДІАГНОСТИКИ ЗОВНІШНЬОГО ДИХАННЯ**

Оперативний контроль стану здоров’я та працездатності організму людини за складом видиху повітря полягає, в основному, у вимірюванні відносної концентрації СО2, О2 і СО. Більш інформативним є одночасні виміри й інших характеристик організму людини, таких як частота дихання, частота серцевих скорочень, вологість та температура повітря, що видихається. В роботі [1] наведено методику аналізу повітря видиху на наявність маркерних газів і наведено шляхи апаратного вирішення проблеми. Саме моніторинг концентрації СО2, О2 і СО, визначення об’ємів цих газів, виділених при видиху, їх співвідношення, має виконуватись в портативному інтелектуальному приладі на основі мікропроцесорної техніки з використанням електрохімічних та інших сенсорів [1].

Основною функцією мого приладу буде вимірювання відносного рівня монооксиду вуглецю у видиху людини. СО, або чадний газ, є одним з найбільш токсичних компонентів продуктів горіння. Чадний газ входить до складу диму і виділяється при тлінні та горінні майже всіх горючих речовин. У невеликих кількостях він навіть утворюється в організмі людини. Підступність чадного газу проявляється у тому, що він легко проходить через бар’єр легень, потрапляючи у кров, легко вступає в контакт з білком гемоглобіном. Найгірше те, що моноокис вуглецю набагато швидше та сильніше може зв’язуватися з гемоглобіном у порівнянні з киснем, витісняючи його і утворюючи досить стійку сполуку — карбоксигемоглобін. Кров при цьому втрачає здатність переносити і правильно використовувати кисень, що пошкоджує мозок та інші органи. В результаті спричиняється кисневе голодування організму в цілому, що при тривалому вдиханні великої концентрації чадного газу призводить до летального наслідку [1].

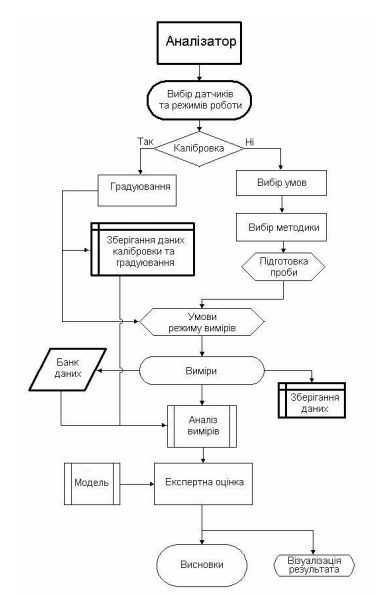


Рис. 1.1. Структурна схема загального алгоритму роботи газоаналізатора [1].

Загальний алгоритм роботи газоаналізатора можна розбити на декілька основних етапів, як це показано на рис. 1.1:

* вибір датчиків та їх калібрування, враховуючи умови режиму вимірів;
* виміри та зберігання даних;
* аналіз та візуалізація результатів.

**1.2. КОМПОНЕТНА БАЗА ЕЛЕКТРОНІКИ ДЛЯ ПРИЛАДУ ДІАГНОСТИКИ ЗОВНІШНЬОГО ДИХАННЯ**

1.2.1. Мікроконтроллер STM32

Інструмент простіше, ніж машина. Найчастіше інструментом працюють руками, а машину приводить у дію парова сила чи тварина.

**Чарльз Беббідж**

Стрімкий розвиток електроніки змінює наше життя на краще, тому ми помічаємо як світові компанії намагаються розробити все кращі і кращі процесори, чіпи та інші електроні датчики.

Мікропроцесор або мікроконтролер на його основі – це одна з найважливіших частин усіх електричних приcтроїв у нашому сучасному світі. Саме мікропроцесор можна вважати серцем будь-якого цифрового електронного пристрою. Перед тим як вибрати або почати роботу з мікроконтролером потрібно знати його основні характеристики та як саме працює процесор.

Є два основних компонента процесора. Один з них пристрій керування (control unit, CU), який допомагає процесору контролювати і виконувати інструкції, саме на нього в першу чергу надходять команди і він з відповідними інструкціями повідомляє, що саме потрібно робити іншим частинам комп`ютера, включно з другим не менш важливим компонентом – арифметико-логічним пристроєм (arithmetic logic unit, ALU). Також треба зазначити, що існує два варіанти реалізації CU. CU на жорсткій логіці (hardwired control units, HCU) та CU з мікропрограмним керуванням (microprogrammable control units, MCU). Функціональність HCU визначається його внутрішньою електричною будовою, тому без фізичного втручання модифікація цієї частини неможлива. А MCU може бути запрограмований для будь-яких цілей, тому і має більш гнучкіший функціонал і простіший в налаштуванні та корегуванні. Арифметико-логічний пристрій відповідає за виконання арифметичних та логічних операцій, наприклад додавання, віднімання, логічне “або” та інші. Він складається з логічних елементів, які можуть утворювати велику структурну схему для виконання більш складних задач. Для прикладу, на рис. 1.2 наведено структурну схему для повного однобітного суматора, складеного з п’яти логічних елементів XOR, AND, OR.

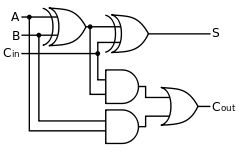


Рис. 1.2. Структурна схема повного однобітного суматора.

Також дуже важлива складова будь-якого мікроконтролера це його пам'ять, яка в основному ділиться на оперативну (RAM) та постійну внутрішню пам'ять (ROM). В свою чергу оперативна пам’ять процесора – це, як правило, набір певної кількості регістрів.

Почати своє ознайомлення з мікропроцесорами я вирішив саме з сімейства 32-бітних мікроконтролерів виробництва STMmicroelectronics -STM32. Чіпи STM32 групуються в серії, в рамках кожної з якої використовується одне і теж 32-бітне ядро ARM, наприклад [Cortex-M7F](https://ru.wikipedia.org/w/index.php?title=Cortex-M7F&action=edit&redlink=1), [Cortex-M4F](https://ru.wikipedia.org/w/index.php?title=Cortex-M4F&action=edit&redlink=1), [Cortex-M3](https://ru.wikipedia.org/w/index.php?title=Cortex-M3&action=edit&redlink=1) [4].

Виробник ділить всі серії мікроконтролерів STM32 на 4 платформи, наведені у табл. 1.

Таблиця 1.

|  |  |
| --- | --- |
| **Платформа контролера** | **Назва серії** |
| Високопродуктивні | F2, F4, F7, H7 |
| Широкого застосування | F0, F1, F3, F4, G0 |
| Наднизького споживання | L0, L1, L4, L5 |
| Бездротові | WB, WL |

Сімейство лінії STM32F103 включає в себе високопродуктивне 32-розрядне ядро ARM Cortex-M3, що працює на частоті 72 МГц, має високошвидкісну вбудовану пам’ять (флеш-пам’ять до 128 Кбайт і SRAM до 20 Кбайт) та широкий спектр вводів-виводів для периферійних пристроїв. Також моделі цього сімейства мікроконтролерів обладнані двома 12-розрядними АЦП, трьома 16-розрядними таймера загального призначення, а також стандартними та вдосконаленими інтерфейсами зв'язку I2C, SPI, USART, USB та CAN. STM32F103 працює від джерела живлення від 2.0 до 3.6 В. Також має рекомендований температурний діапазон від -40 до +105 градусів за Цельсієм. Завдяки цим характеристикам мікроконтролери сімейства STM32F103 досить добре підходять для широкого спектру застосувань, таких як приводи двигунів, ПК та різні ігрові пристрої, GPS платформи, промислове обладнання та медичне портативне обладнання, а це те, що відповідає поставленій в роботі задачі. Позначення окремих контактів мікроконтролера STM32F103C8 наведено на рис. 1.3

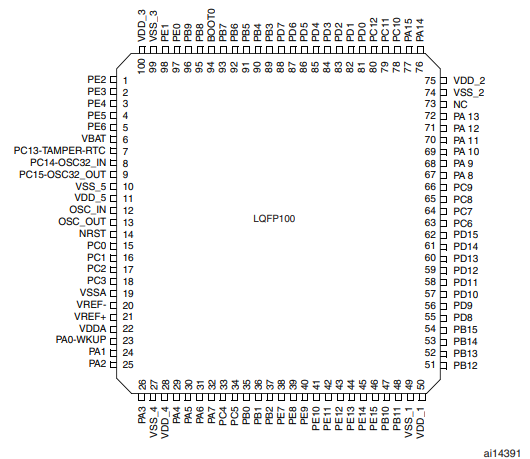


Рис. 1.3. Розміщення контактів на платі контролеру STMF103C8 [4].

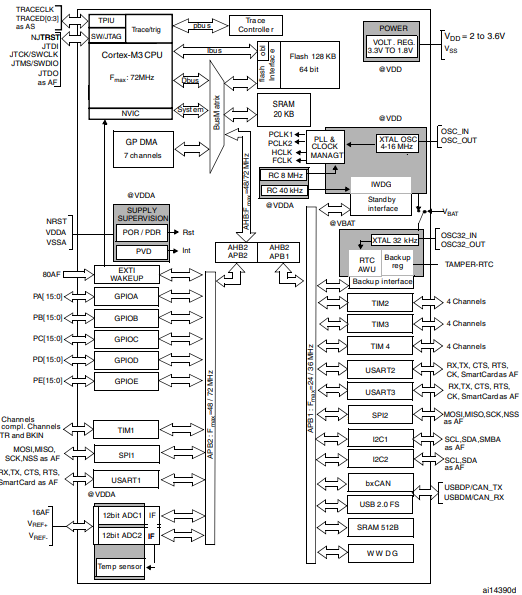


Рис. 1.4 Структурна схема контролеру STM32F103C8 [4].

На рис. 1.4 наведено структурну схему мікроконтролера STM32F103, де показані усі вищезазначені інтерфейси зв’язку та деякі характеристики.

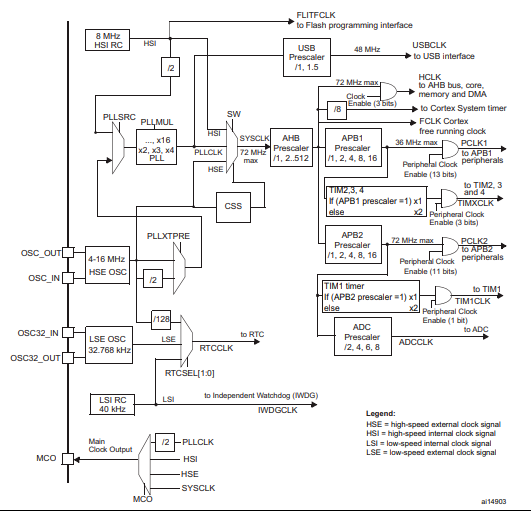


Рис. 1.5 Дерево часових конфігурацій контролеру STM32F103C8 [4].

На рис. 1.5 зображено дерево часових конфігурацій мікроконтролера STM32F103. В ході виконання курсового проекту цим часовим діаграмам буде приділятись мало уваги, бо програмне забезпечення STM32 надає змогу легко організувати роботу з часовими характеристиками в графічному інтерфейсі.

На підставі детального аналізу характеристик мікроконтролерів сімейства STM32F103 для виконання поставленого в проекті завдання було обрано мікроконтролер STM32F103C8.

Перед тим, як почати працювати з контролером STM32F103C8 на пряму, потрібно завантажити та встановити відповідне програмне забезпечення та здійснити його налаштування. Для створювання програмного коду для контролерів STM32, компанія STMicroelectronics випустила STM32CubeIDE – інтегроване середовище розробки, де є майже всі інструменти для початку роботи. Але в курсовій роботі розглянуто стек розробки CubeProg, CubeMX та редактор коду Keil uVision5 для того, щоб краще розібратись в створенні програм на низькому рівні.

Зазначимо різницю між STM32F103 і іншими контролерами, які пропонують для початківців. Насамперед, це контролер з сімейства Arduino. Arduino представлена у різноманітних варіантах, наприклад, Arduino Nano, Arduino Uno та інші. Їх підключення відбувається за допомогою USB кабелю та настройкою певних характеристик у самому середовищі Arduino IDE. Тоді як STM32F103 підключення відбувається через програматор, а в нашому випадку це STlinkV2.



Рис.1.6 Программатор STlinkV2.

Одною з найбільших переваг STM32 над Arduino є те, що STM32 дозволяє «глибше» налаштовувати певні команди та регістри, які використовує мікропроцесор. Це дає змогу підійти ближче до розуміння роботи процесорів.

1.2.2. Датчик моно оксиду вуглецю MQ7

В даній роботі, для вимірювання відносної концентрації СО у видиху буде використано серію сенсорів MQ, оскільки вони є єдиним доступним рішенням як для промислових потреб, так і для більш вузьких сфер використання.

MQ7 – нормований датчик для детектування рівня чадного газу, також має досить непогану чутливість до природного газу, побутового газу і різних зріджених попутних нафтових газів (пропан, пропілен, бутан та інші). Основним робочим елементом датчика є нагрівальний елемент, за рахунок якого відбувається хімічна реакція, в результаті якої на аналоговий вихід надходить сигнал, пропорційний до концентрації газу.

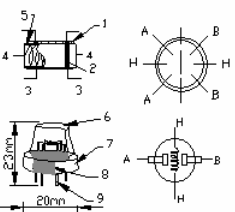


Рис. 1.6. Схематична будова датчика СО MQ7 [5].

Основні характеристики датчика MQ7 наведені у табл. 2.

Таблиця 2.

|  |  |
| --- | --- |
| Напруга живлення | 5 В |
| Струм споживання | 160 мА |
| Час розжарення нагрівального елементу | 60-90 с |
| Потужність нагрівача | 350 мВт |
| Температурний діапазон | -10 - 50 °C |
| Діапазон чутливості | 10 – 10000 ppm |

1.2.3. Модуль LCD MAX7219

Для відображення показів датчика СО знадобиться додаткова периферія, на яку будуть виводитися зчитані дані. Для цієї задачі я вибрав цифровий восьмирозрядний семисегметний модуль на основі драйвера MAX7219. Драйвер керується за допомогою SPI інтерфейсу та підтримує частоту роботи інтерфейсу до 15 МГц. Мікросхема MAX7219 має три лінії, по яким здійснюється передача даних. Лінія DIN – для інформаційного сигналу, CLK – для тактового та LOAD, який дає дозвіл на приймання даних. Також є ще одна лінія DOUT, яка може використовуватись для виведення даних при під’єднані декількох мікросхем. На рис. 1.7 показано функціональну схему драйверу MAX7219, на якій позначено усі вище зазначені входи та виходи у вигляді цифр на дисплеї.

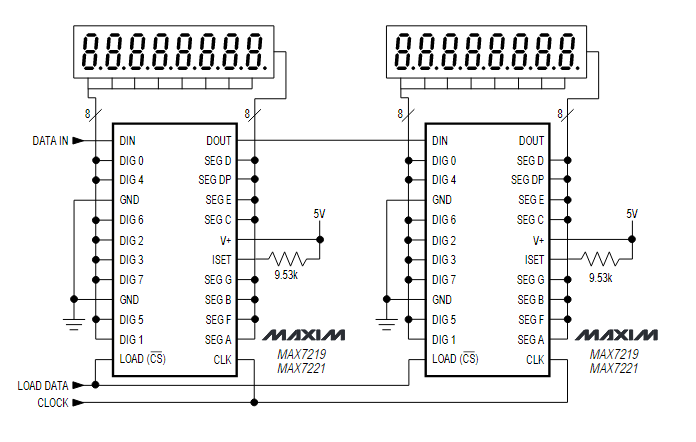


Рис.1.7. Функціональнаа схема драйверу MAX7219 [6].

Якщо глибше зануритись у роботу дисплею, то потрібно знати як саме відбуватся показ того чи іншого символу. Для цього знадобиться лише один розряд семисегметного дисплею. На рис. 1.8 зображено дисплей, який поділений на 7 сегметів від A до G і, при певному порядку подачі сигналів на його входи, сегменти будуть засвічені. Для прикладу, для отримання зображення нуля, потрібно подати сигнали високого рівня на всі сегменти, крім G, тоді й буде відображатися символ «0». Також варто зазначит, що дисплеї можуть працювати по різному: в одному випадку для того, щоб засвітився будь-який сегмент потрібно подати сигнал високого рівня, а в іншому – навпаки. Це пов'язано з апаратною частининою, бо у деяких дисплеїв елементом відображення є катод, а у інших - анод.

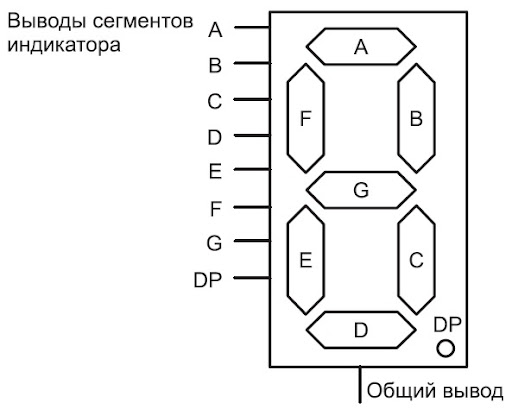


Рис. 1.8. Семисегметний однорозрядний дисплей [6].

Основні характеристики драйверу MAX7219 наведені у табл. 3.

Таблиця 3:

|  |  |
| --- | --- |
| Частота тактування інтерфейсу | 10 МГц |
| Напруга живлення В | 4.5-5 В |
| Споживання по шині +5В в сплячому режимі | 150 мкА |
| Середній струм через один сегмент | До 40 мА |
| Частота проходження імпульсів включення символів | >500 Гц |
| Затримка , мс | 2.2 мс |

**РОЗДІЛ 2**

**ПРОГРАМНО-АПАРАТНИЙ КОМПЛЕКС ДЛЯ ВИМІРЮВАННЯ ВІДНОСНОЇ КОНЦЕНТРАЦІЇ ОКИСУ ВУГЛЕЦЮ У ВИДИХУ**

2.1. Структурна схема пристрою

На рис. 2.1 зображено функціональну схему підключення елементів розроблюваного пристрою. Датчики MQ7 мають живлення +5 В від самого мікроконтролера. Їхні аналогові виводи А0 під'єднані до виводів PA0 та PA1 мікроконтролера, саме для них налаштовується інтерфейс ADC для зчитування аналогового сигналу. Модуль дисплею MAX7219 також має живлення у +5 В від самого мікроконтролера та підключення до нього здійснюється таким чином: PA7 – DIN (MOSI), PA4 – LOAD (Chip Select), PA5 – CLK.

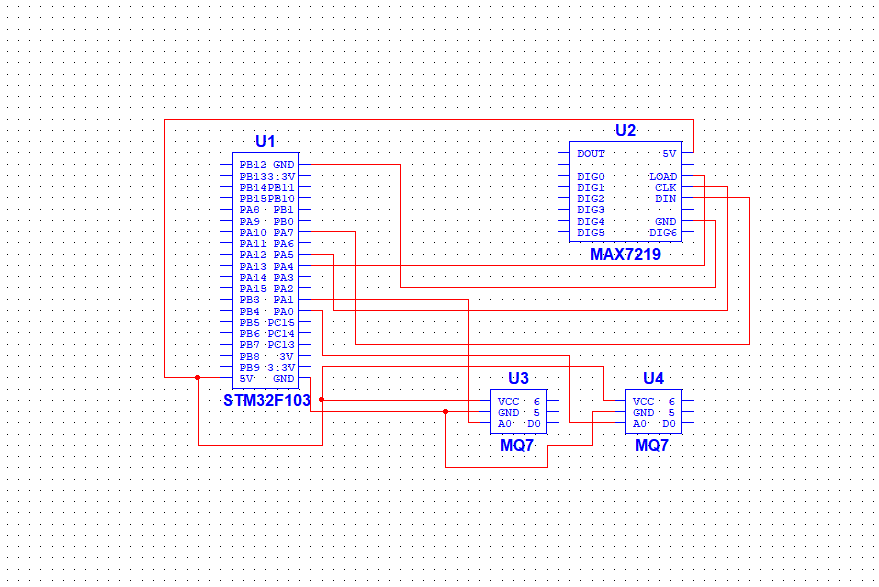
****

Рис. 2.1. Функціональна схема розроблюваного пристрою.

**РОЗДІЛ 3**

**ПРОГРАМНИЙ КОД ДЛЯ STM32F103**

3.1. Середовище розробки

3.1.1. STM32CubeProgrammer

STM32CubeProgrammer – це універсальний програмний інструмент для програмування продуктів STM32. CubeProgrammer надає ефективне середовище для читання, запису та перевірки пам’яті пристрою як через інтерфейс налагодження (JTAG і SWD), так і через інтерфейс завантажувача (UART, USB DFU, I2C, SPI, CAN). CubeProgrammer надає широкий спектр функцій для програмування внутрішньої та зовнішньої пам’яті мікроконтролера. Насамперед, після підключення програматора і перед завантаженням програмного коду потрібно переконатись, що мікроконтролер справний і має очищену пам'ять. Це можна дізнатись через опцію Memory & FileEdition, де зображено карту прошивки його пам’яті.

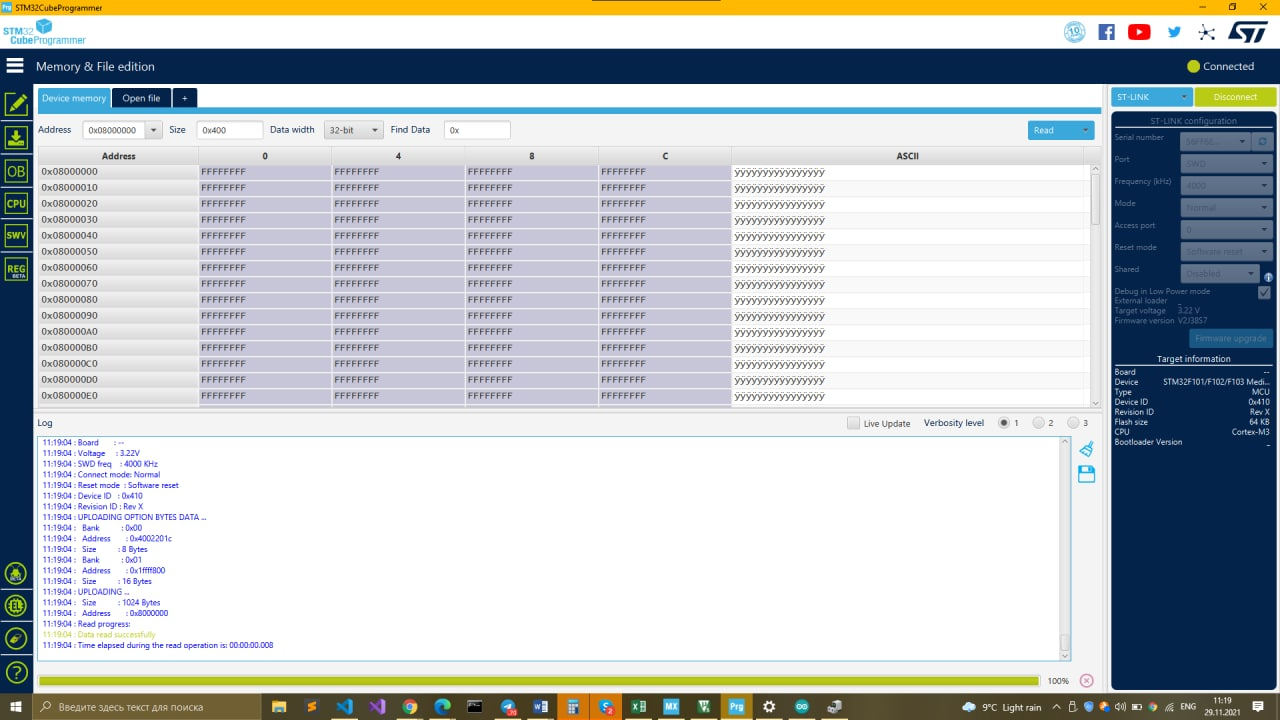


Рис. 2.2. Скріншот екрану програмного середовища CubeProgrammer.

Опція Memory & FileEdition.

3.1.2. STM32CubeMX

STM32CubeMX – це графічний інструмент, який дозволяє легко конфігурувати мікроконтролери та мікропроцесори, а також генерувати відповідний код ініціалізації на мові програмування «С» для ядра ARM Cortex-M за декілька кроків.

Першим кроком потрібно обрати мікроконтролер – STM32F103C8, як це показано на рис.2.3. Обираємо STM32F103C8.

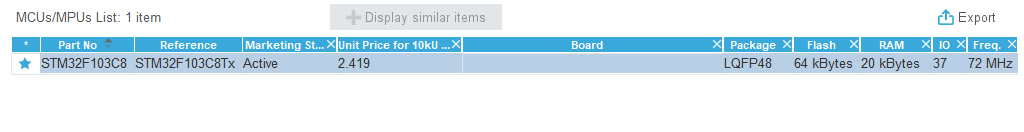


Рис. 2.3 Опція для вибору мікроконтролерів у CubeMX.

Другим кроком потрібно налаштувати усі периферії та канали зв'язку через виводи на платі. В меню Categories, A->Z перелічено усі інтерфейси. ADC1, ADC2, GPIO, NVIC, SPI1, SYS, RCC (див. рис. 2.4). Коректність підключення виводів відображено в графічному інтерфейсі в опції Pinout view, як це показано на рис. 2.5.

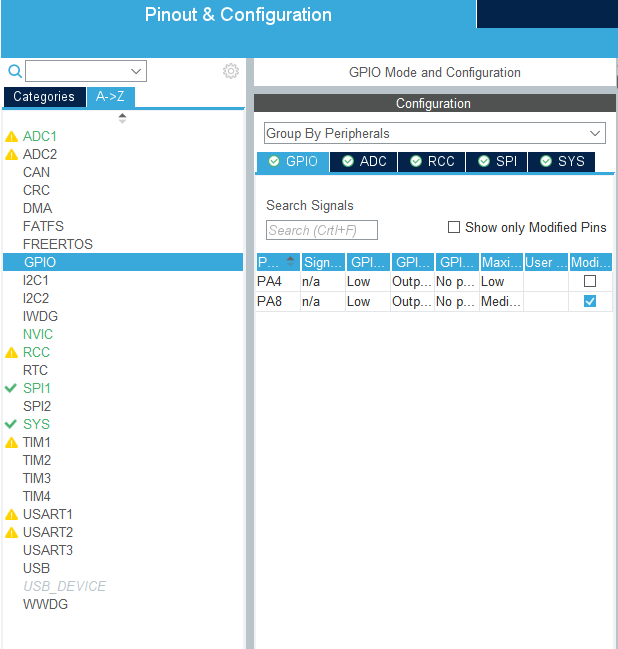


Рис. 2.4. Меню Categories, A->Z у CubeMX.

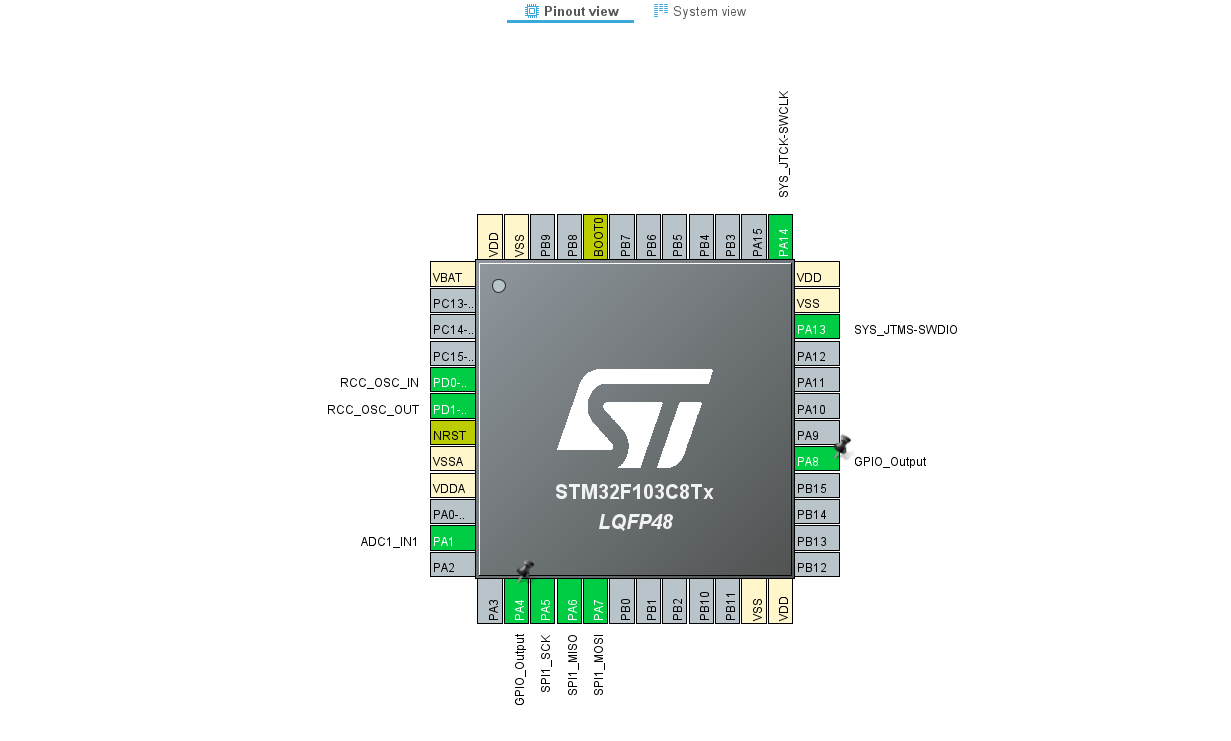


Рис. 2.5. Меню Pinout view у CubeMX.

3.1.3. Keil uVision5

Keil uVision – найкращий вибір для розробки програмного забезпечення під мікроконтролери на базі ARM. Keil uVision включає в себе всі необхідні компоненти для створення та налагодження вбудованих програм.

Файлова система проекту складається з чотирьох головних гілок, де зберігаються усі файли заголовків і файли програмного коду на мові програмування «С», як показано на рис. 2.7.

|  |  |
| --- | --- |
|  |  |
| Рис. 2.7. Меню Project у Keil uVision. | Рис. 2.8. Папка Application/User у Keil uVision |

Оскільки мікроконтролери STM32 дозволяють глибше зазирнути до налаштувань окремих частин, на рис. 2.8 зображено файли, які відповідають за конфігурацію інтерфейсів, які використовуються в CubeMX.

3.1.4. Створення проекту та кодування алгоритму роботи аналізатора СО.

Для початку потрібно підключити дисплей MAX7219. Для цього потрібно налаштувати функцію для інтерфейсу SPI. Першим кроком створюється статична функція. Першим та другим рядком створюється дві структури, одна з яких слугує для вводу-виводу, а інша для шини SPI. Третім та четвертим рядком налаштовується сигнал тактування окремого порту та налаштування виводів мікроконтролера. В останню чергу викликається бібліотечна функція LL\_SPI\_Init, куди передаються аргументи: структура та об’єкт SPI1 для того щоб програма розуміла, який номер інтерфейсу використовується [2, 3]. В результаті отримується код, наведений нижче.

В main.c :

**static** **void** **MX\_SPI1\_Init**(**void**)

{

/\*рядок 1-2 \*/

LL\_SPI\_InitTypeDef SPI\_InitStruct = {**0**};

LL\_GPIO\_InitTypeDef GPIO\_InitStruct = {**0**};

/\*рядок 1-2 \*/

/\*рядок 3-4 \*/

LL\_APB2\_GRP1\_EnableClock(LL\_APB2\_GRP1\_PERIPH\_SPI1);

LL\_APB2\_GRP1\_EnableClock(LL\_APB2\_GRP1\_PERIPH\_GPIOA);

/\*рядок 3-4 \*/

/\* Налаштування режиму роботи ніжок мікроконтролера\*/

GPIO\_InitStruct.Pin = LL\_GPIO\_PIN\_5|LL\_GPIO\_PIN\_7;

GPIO\_InitStruct.Mode = LL\_GPIO\_MODE\_ALTERNATE;

GPIO\_InitStruct.Speed = LL\_GPIO\_SPEED\_FREQ\_HIGH;

GPIO\_InitStruct.OutputType = LL\_GPIO\_OUTPUT\_PUSHPULL;

LL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);

GPIO\_InitStruct.Pin = LL\_GPIO\_PIN\_6;

GPIO\_InitStruct.Mode = LL\_GPIO\_MODE\_FLOATING;

LL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);

/\* Налаштування режиму роботи ніжок мікроконтролера\*/

/\* Заповненняструктуру та викликаємо функцію SPI\_Init \*/

SPI\_InitStruct.TransferDirection = LL\_SPI\_FULL\_DUPLEX;

SPI\_InitStruct.Mode = LL\_SPI\_MODE\_MASTER;

SPI\_InitStruct.DataWidth = LL\_SPI\_DATAWIDTH\_16BIT;

SPI\_InitStruct.ClockPolarity = LL\_SPI\_POLARITY\_HIGH;

SPI\_InitStruct.ClockPhase = LL\_SPI\_PHASE\_2EDGE;

SPI\_InitStruct.NSS = LL\_SPI\_NSS\_SOFT;

SPI\_InitStruct.BaudRate = LL\_SPI\_BAUDRATEPRESCALER\_DIV16;

SPI\_InitStruct.BitOrder = LL\_SPI\_MSB\_FIRST;

SPI\_InitStruct.CRCCalculation = LL\_SPI\_CRCCALCULATION\_DISABLE;

SPI\_InitStruct.CRCPoly = **10**;

LL\_SPI\_Init(SPI1, &SPI\_InitStruct);

/\* Заповнення структуру та викликаємо функцію SPI\_Init \*/

}

Викликом функції MX\_SPI1\_Init мікроконтролер починає «спілкування» через інтерфейс SPI.

MX\_SPI1\_Init();

Щоб використовувати функції дисплею, потрібно підключити бібліотеку max7219

#include "max7219.h"

Init\_7219(); // ініциалізація

Функції виводу:

Number\_7219();

Clear\_7219();

NumberR\_7219();

NumberL\_7219();

Тепер щоб зчитувати данні з датчика потрібно налаштувати інтерфейс ADC.

**static** **void** **MX\_ADC1\_Init**(**void**)

{

/\* Створення структур для окремих регістрів\*/

LL\_ADC\_InitTypeDef ADC\_InitStruct = {**0**};

LL\_ADC\_CommonInitTypeDef ADC\_CommonInitStruct = {**0**};

LL\_ADC\_REG\_InitTypeDef ADC\_REG\_InitStruct = {**0**};

LL\_GPIO\_InitTypeDef GPIO\_InitStruct = {**0**};

/\* Створення структур для окремих регістрів\*/

/\* Тактування переферії та тактування порта\*/

LL\_APB2\_GRP1\_EnableClock(LL\_APB2\_GRP1\_PERIPH\_ADC1);

LL\_APB2\_GRP1\_EnableClock(LL\_APB2\_GRP1\_PERIPH\_GPIOA);

/\* Тактування переферії та тактування порта\*/

/\* Налаштування ніжки \*/

GPIO\_InitStruct.Pin = LL\_GPIO\_PIN\_1;

GPIO\_InitStruct.Mode = LL\_GPIO\_MODE\_ANALOG;

LL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);

/\* Налаштування ніжки \*/

/\* Заповнення структур та виклик функцій ADC\_REG\_Init \*/

ADC\_InitStruct.DataAlignment = LL\_ADC\_DATA\_ALIGN\_RIGHT;

ADC\_InitStruct.SequencersScanMode = LL\_ADC\_SEQ\_SCAN\_DISABLE;

LL\_ADC\_Init(ADC1, &ADC\_InitStruct);

ADC\_CommonInitStruct.Multimode = LL\_ADC\_MULTI\_INDEPENDENT;

LL\_ADC\_CommonInit(\_\_LL\_ADC\_COMMON\_INSTANCE(ADC1), &ADC\_CommonInitStruct);

ADC\_REG\_InitStruct.TriggerSource = LL\_ADC\_REG\_TRIG\_SOFTWARE;

ADC\_REG\_InitStruct.SequencerLength = LL\_ADC\_REG\_SEQ\_SCAN\_DISABLE;

ADC\_REG\_InitStruct.SequencerDiscont = LL\_ADC\_REG\_SEQ\_DISCONT\_DISABLE;

ADC\_REG\_InitStruct.ContinuousMode = LL\_ADC\_REG\_CONV\_SINGLE;

ADC\_REG\_InitStruct.DMATransfer = LL\_ADC\_REG\_DMA\_TRANSFER\_NONE;

LL\_ADC\_REG\_Init(ADC1, &ADC\_REG\_InitStruct);

/\* Заповнення структур та виклик функцій ADC\_REG\_Init \*/

}

Користуючись стандартною бібліотекою LL відбувається зчитування аналогового сигналу , який надходить до мікроконтролера з датчика.

//початок конвертування аналогового сигналу

LL\_ADC\_REG\_StartConversionSWStart(ADC1);

//кінець конвертації

**while** (!LL\_ADC\_IsActiveFlag\_EOS(ADC1)) {}

LL\_ADC\_ClearFlag\_EOS(ADC1);

//зчитування даних в мілівольтах

ADC\_Data = LL\_ADC\_REG\_ReadConversionData12(ADC1);

ADC\_mVolt = \_\_LL\_ADC\_CALC\_DATA\_TO\_VOLTAGE((**uint32\_t**)**3275**, ADC\_Data, LL\_ADC\_RESOLUTION\_12B);

// зчитування даних в мілівольтах

//Вивід данні на дисплей

Number\_7219(ADC\_mVolt);

// Вивід данні на дисплей

ВИСНОВКИ

У ході виконання курсового проєкту:

* розроблено структурну схему та пристрою для вимірювання відносної концентрації СО у видиху людини;
* обрано компоненти схеми:
  + мікроконтролер STM32F103;
  + датчик СО семисегметний дисплей MAX7219;
  + семисегметний дисплей MAX7219;
* досліджено роботу процесора мікроконтролера та зібрано прототип пристрою;
* написано програмний код мікроконтролера на мові програмування «С»;
* для створення інтерфейсу зв’язку між платою мікроконтролера і датчиками використано програмне забезпечення STM32CubeMX.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. С.І Лукаш , Л.П. Вакал «Комп’ютерні засоби , мережі та системи С.І Лукаш, Л.П. Вакал»

2. Брайана Кернигана «Мова програмування С»

3. Уоррен Гей «Beginning STM32»

4. Технічна документація мікроконтролeр STM32F103С8 URL :

https://www.st.com/resource/en/datasheet/STM32F103c8.pdf

5.Технічна документація датчик детектування рівня чадного газу MQ7 URL : https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf

6.Технічна документація драйевра дисплею MAX7219 URL : <https://category.alldatasheet.com/index.jsp?components=DISPLAY>

7. Annsofi Sandberg, Johan Grunewald «Assessing Recent Smoking Status by Measuring Exhaled Carbon Monoxide Levels»