

Національний університет «Полтавська політехніка імені Юрія Кондратюка»

(повне найменування закладу вищої освіти)

Навчально-науковий інститут інформаційних технологій і робототехніки

(повне найменування інституту, назва факультету (відділення))

Кафедра автоматики, електроніки та телекомунікацій

(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до кваліфікаційної роботи

магістр

(ступінь вищої освіти)

на тему «Система моніторингу та керування параметрами мікроклімату в приміщеннях на основі стандартів ЄС»

Виконав: студент б курсу, групи 601-МЕ

спеціальності 141 «Електроенергетика,

електротехніка та електромеханіка»

шифр і назва напрямку підготовки, спеціальності

Косинков А.О.

(прізвище та ініціали)

Керівник Боряк Б.Р.

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Полтава - 2022 рік

ЗМІСТ

Вступ	5
Розділ 1. Аналіз стандартів параметрів мікроклімату України та країн ЄС та впливу зміни параметрів на людину	6
1.1. Стандарти параметрів мікроклімату України та країн ЄС	6
1.2. Дослідження впливу зміни параметрів мікроклімату на людину	10
1.3. Висновок до розділу 1	13
Розділ 2. Дослідження технічного забезпечення систем контролю мікроклімату	14
2.1. Аналіз існуючих систем контролю параметрів мікроклімату.....	14
2.2. Аналіз існуючих вимірювальних засобів систем контролю параметрів мікроклімату	21
2.2.1. Модуль датчика якості повітря MQ-135.....	25
2.2.3. Датчик CO ₂ (Arduino-сумісний) від DFRobot.....	32
2.2.4. Датчик вологості та температури DHT22 та DHT11.....	38
2.3. Висновок до розділу 2	42
Розділ 3. Розроблення конфігурації системи моніторингу та керування параметрами мікроклімату в приміщеннях	43
3.1. Підсистема моніторингу температури та вологості навколишнього середовища.....	43
3.2. Підсистема індикації параметрів навколишнього середовища.....	46
3.3. Підсистема вимірювання рівня діоксиду вуглецю	51
3.4. Підсистема керування виконавчими пристроями	55
3.5. Висновок до розділу 3	65
Загальні висновки.....	66
Література.....	67
Додатки.....	70
Додаток А. Section 3. development of the configuration of the indoor microclimate monitoring and management.....	70

Додаток Б. Програмний код системи моніторингу та керування параметрами мікроклімату	93
Додаток В. Презентація кваліфікаційної роботи	97

Вступ

Актуальність теми. Зараз, в час науково-технічного прогресу, багато компаній малого бізнесу та частина середнього в погоні за прибутком забувають про приміщення, в яких працюють їхні робітники. Не дотримуючись певних температурних норм, чи не слідкуючи за рівнем вуглекислого газу в приміщенні, здоров'я працівників знаходиться в зоні ризику, а також знижується ефективність праці. Актуальність теми визначається важливістю моніторингу і збереження здоров'я працівників. Дослідивши норми мікроклімату приміщень, реальним буде розроблення системи, яка не лише дозволить створити комфортні умови для працівників, поліпшити їхню працездатність, а і зберегти здоров'я.

Об'єкт дослідження. Процес моніторингу та керування параметрами мікроклімату у виробничих приміщеннях.

Предмет дослідження. Чисельні показники параметрів мікроклімату.

Мета дослідження. Метою роботи є розроблення та дослідження методів моніторингу та керування параметрами мікроклімату у виробничих приміщеннях.

Задачі дослідження:

1. Дослідження стандартів параметрів мікроклімату України та країн ЄС.
2. Аналіз впливу зміни параметрів мікроклімату на стан людини.
3. Дослідження технічних засобів для забезпечення моніторингу параметрів мікроклімату.
4. Розроблення системи моніторингу параметрів мікроклімату у виробничих приміщеннях.
5. Аналіз зміни параметрів мікроклімату у виробничих приміщеннях.
6. Розроблення моделі системи керування параметрами мікроклімату у виробничих приміщеннях.
7. Дослідження можливостей імплементації розробленої моделі системи керування параметрами мікроклімату у виробничих приміщеннях.

Розділ 1. Аналіз стандартів параметрів мікроклімату України та країн ЄС та впливу зміни параметрів на людину

1.1. Стандарти параметрів мікроклімату України та країн ЄС

Згідно з статтею[1] мікроклімат (від мікро- і клімат) — клімат приземного шару повітря, обумовлений мікромасштабними відмінностями земної поверхні усередині місцевого клімату. Наприклад, в місцевому кліматі лісового масиву розрізняють мікроклімат лісових полян, узлісь тощо; в місцевому кліматі міста — мікроклімат площ, провулків, скверів, дворів і ін. Фітоклімат — атмосферні умови в середовищі поширення рослин: в травостої, в кронах дерев тощо.

В даній магістерській роботі ми будемо розглядати мікроклімат в офісі. Для початку потрібно знайти стандарти на які ми будемо опиратися для оцінки мікроклімата в цьому приміщенні. В Україні норми регулюються Державними Санітарними Нормами – ДСН 3.3.6.042-99 прийнятими 01.12.1999 року[2]. Наразі цей стандарт досі діючий та є актуальним

Відповідно до ДСН є терміни я вибрав ті які будуть застосовуватись далі в тексті-

- Виробниче приміщення - замкнутий простір в спеціально призначених будинках та спорудах, в яких постійно (по змінах) або періодично (протягом частини робочого дня) здійснюється трудова діяльність людей.

- Постійне робоче місце - місце, на якому працюючий знаходиться понад 50% робочого часу або більше 2-х годин безперервно. Якщо при цьому робота здійснюється в різних пунктах робочої зони, то вся ця зона вважається постійним робочим місцем.

- Мікроклімат виробничих приміщень - умови внутрішнього середовища цих приміщень, що впливають на тепловий обмін працюючих з оточенням шляхом конвекції, кондукції, теплового випромінювання та випаровування вологи. Ці умови визначаються поєднанням температури, відносної вологості та швидкості

руху повітря, температури оточуючих людину поверхонь та інтенсивністю теплового (інфрачервоного) опромінення.

• Категорія робіт - розмежування робіт за важкістю на основі загальних енерговитрат організму.

Мікрокліматичні умови на робочому місці, у виробничих приміщеннях – найважливіший санітарно-гігієнічний фактор, від якого залежить стан здоров'я та працездатність людини.

Мікрокліматичні умови поділяють на оптимальні та допустимі.

Оптимальні умови – це поєднання параметрів мікроклімату, які при тривалому та систематичному впливі на людину забезпечують зберігання нормального теплового стану організму, без активізації механізмів терморегуляції. Вони забезпечують відчуття теплового комфорту та створюють передумови для високого рівня працездатності.

Таблиця 1.1 Оптимальні величини температури, відносної вологості та швидкості руху повітря в робочій зоні виробничих приміщень

Період року	Категорія робіт	Температура повітря, °С	Відносна вологість, %	Швидкість руху, м/с
Холодний період року	Легка 1а	22-24	60-40	0,1
	Легка 1б	21-23	60-40	0,1
	Середньої важкості 2а	19-21	60-40	0,2
	Середньої важкості 2б	17-19	60-40	0,2
	Важка 3	16-18	60-40	0,3
Теплий період року	Легка 1а	23-25	60-40	0,1
	Легка 1б	22-24	60-40	0,2
	Середньої важкості 2а	21-23	60-40	0,3
	Середньої важкості 2б	20-22	60-40	0,3
	Важка 3	18-20	60-40	0,4

Допустимі мікрокліматичні умови – поєднання параметрів мікроклімату, які при тривалому та систематичному впливі на людину можуть викликати зміни теплового стану організму, що швидко минають і нормалізуються та супроводжуються напруженням механізмів терморегуляції в межах фізіологічної адаптації. При цьому не виникає ушкоджень або порушень стану здоров'я, однак можуть спостерігатися дискомфортні тепловідчуття, погіршення самопочуття та зниження працездатності [3]

Таблиця 1.2 Допустимі мікрокліматичні умови

Період року Категорія робіт		Температура, °С				Відносна вологість (%) на робочих місцях – постійних і непостійних	Швидкість руху(м/сек.) на робочих місцях – постійних і непостійних
		Верхня межа		Нижня межа			
		На постійних робочих місцях	На постійних робочих місцях	На постійних робочих місцях	На непостійних робочих місцях		
Холодний період року	Легка Іа	25	26	21	18	75	Не більше 0,1
	Легка Іб	24	25	20	17	75	Не більше 0,2
	Середньої важкості Іа	23	24	17	15	75	Не більше 0,3
	Середньої важкості Іб	21	23	15	13	75	Не більше 0,4
	Важка ІІІ	19	20	13	12	75	Не більше 0,5
Теплий період року	Легка Іа	28	30	22	20	55-при 28 °С	0,2-0,1
	Легка Іб	28	30	21	19	60-при 27 °С	0,3-0,1
	Середньої важкості Іа	27	29	18	17	65-при 26 °С	0,4-0,2
	Середньої важкості Іб	27	29	15	15	70-при 25 °С	0,5-0,2
	Важка ІІІ	26	28	15	13	75-при 25 °С	0,6-0,5

Мікрокліматичні умови виробничих приміщень характеризуються такими показниками:

- температура повітря,
- відносна вологість повітря,
- швидкість руху повітря,
- інтенсивність теплового (інфрачервоного) опромінення,
- температура поверхні.

Вимірювання параметрів мікроклімату проводяться на робочих місцях і в робочій зоні на початку, в середині та в кінці робочої зміни. При коливаннях мікрокліматичних умов, пов'язаних з технологічним процесом та іншими причинами, вимірювання проводяться з урахуванням найбільших і найменших величин термічних навантажень протягом робочої зміни

Зазвичай вимірювання параметрів мікроклімату проводиться спеціальними приладами такими як – Аспіраційний психрометр із ртутними термометрами для вимірювання температури повітря, гігрограф для замірювання вологості повітря, електротермометри, термопари для вимірювання температури поверхонь, Актинометри, термостовбці для вимірювання інтенсивності інфрачервоного випромінювання.



Рис.1 Прилади для вимірювання параметрів мікроклімату

Існують вимоги до вимірювальних приладів табл.1.3

Таблиця 1.3 Вимоги до приладів які вимірюють параметри мікроклімату

Вимірювані величини	Діапазон вимірювань	Допустима похибка	Рекомендовані прилади
1.Температура повітря, °С	-30 до +5	+0,1	Аспіраційний психометр із ртутними термометрами
2. Відносна вологість повітря, %	15 до 100	+5,0	Ті ж самі та записуючі гігрографи
3. Температура поверхні, °С	-30 до 100	+1,0	Електротермометри,термопари і т. ін
4. Швидкість руху повітря, м/сек.	0,1-0,5 до 0,6 – 5,0	+0,1 +0,2	Анемометри ротаційної дії
5. Інтенсивність інфрачервоного опромінення	10,0- 20000,0	+10%	Актинометри, термостовбці, болометри, радіометри зі спектральною чутливістю в діапазоні 0,30 – 20,0 мкм

1.2. Дослідження впливу зміни параметрів мікроклімату на людину

Досліджуючи [4] За оптимальних мікрокліматичних умов в організмі працівника, завдяки терморегуляції, підтримується постійна температура тіла (36,6°С). Кількість тепла, що утворюється в організмі, залежить від фізичного навантаження працівника, а рівень тепловіддачі – від мікрокліматичних умов виробничого середовища. Віддача тепла здійснюється за допомогою випромінювання та випаровування вологи з поверхні шкіри. Чим нижча температура повітря і більша швидкість його руху, тим більше тепла віддається організмом. При високій температурі повітря значна частина тепла втрачається

випаровуванням. Разом з потом організм втрачає воду, вітаміни, мінеральні солі. Таким чином, внаслідок зневоднювання, порушується обмін речовин. Тому працівники «гарячих» цехів повинні забезпечуватись газованою підсоленою водою.

Вологість повітря істотно впливає на самопочуття та працездатність. Через високу вологість зменшується віддача тепла за допомогою випаровування. Зниження вологості покращує процес тепловіддачі. Однак, і надто низька вологість викликає висихання слизових оболонок дихальних шляхів. Для забезпечення допустимих параметрів мікроклімату на виробництві впроваджується механізація важких робіт, обов'язкова наявність припливно-витяжної вентиляції з механічним спонуканням, а також додатково, кондиціонування повітря. У відповідності до ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень», фізіологічно оптимальна відносна вологість становить 40-60%, допустиме значення не більше 75%.

Таблиця 1.4 Класифікація за рівнем концентрації CO₂ у повітрі будівлі/приміщення відповідно ДБН В.2.5-67:2013 “Опалення, вентиляція, кондиціонування”.

Умови мікроклімату	Рівень концентрації CO ₂ у приміщенні понад рівень у зовнішньому повітрі, ppm	
	Типовий діапазон	Типове значення
Підвищені оптимальні	<400	350
Оптимальні	400-600	500
Допустимі	600-1000	800
Обмежено допустимі	>1000	1200

CO₂ — вуглекислий газ або діоксид вуглецю — це безбарвний газ, який не має запаху при малих концентраціях. Він виділяється людьми, тваринами та рослинами, наприклад, організм людини здатний виділити близько 1 кг вуглекислого газу на добу. Рекомендований склад повітря у приміщенні: 20% кисню, 79% азоту, 1% інших газів. У цей 1%, що залишився, входять різноманітні сполуки, одна з яких —

CO₂, вуглекислий газ. Його кількість від загальної маси — 0,035-0,04%, або 350-400 ppm (за європейською системою позначень).

В Україні державні будівельні норми визначають допустимі концентрації CO₂ в повітрі з урахуванням вмісту двоокису вуглецю в повітрі зовні будівель.

При надмірному рівні CO₂ може відбутися погіршення стану здоров'я працівника (відчуття задухи, пришвидшення пульсу, збільшення стомлюваності, проблеми з концентрацією уваги, падіння продуктивності праці, непритомність, нудота, головні болі, синдром хронічної втоми)

Таблиця 1.5 Вираженість симптомів залежно від концентрації рівня CO₂

Рівень CO ₂ (ppm)	Вплив на людину
300-400 ppm	Оптимальне значення
400-600 ppm	Норма для роботи в офісі без зниження продуктивності праці. На організм ніяк не впливає
До 600 ppm	Норма для спальні, дитячого садка і школи
600-1000 ppm	Падає продуктивність праці, складніше концентрувати увагу і сприймати інформацію. З'являється головний біль
1000-1500 ppm	Людина стає млявою, падає концентрація уваги, складніше приймати рішення, збільшується кількість помилок в роботі. Виникає ризик розвитку хронічної втоми
Від 1500	Сильний головний біль. Зникає бажання працювати через втому.

Також проаналізувавши [5] Коли температура навколишнього середовища знижена, активується хімічна терморегуляція — зростає кількість тепла, яке виробляє організм. Одночасно звужуються шкірні судини. Через це зменшується віддача тепла у навколишнє середовище.

Коли температура навколишнього середовища підвищена, зменшується тепловіддача внаслідок конвекції, кондукції, радіації і зростає тепловіддача через випаровування (потовиділення) вологи з поверхні тіла.

За підвищеної температури повітря від 16 до 30 °С і роботи середнього ступеня тяжкості в одягненої людини частка тепловіддачі через конвекцію і радіацію знижується з 73 до 8%. Тепловіддача через випаровування вологи зростає з 27 до 92%. Віддача тепла з поверхні тіла через конвекцію і радіацію припиняється, коли температура навколишнього середовища становить 33 °С.

1.3. Висновок до розділу 1

Проаналізувавши норми мікроклімату у виробничих приміщеннях відповідно до нормативних документів України та країн ЄС варто відзначити, що діючий стандарт в Україні є ДСН 3.3.6.042-99. Згідно нього є зазначені оптимальні та допустимі норми показників температури, вологості та швидкості руху повітря на робочому місці. При порушенні оптимального мікроклімату, змінюється працездатність робітників, чим холодніше, тим більше людина витрачає енергії на зігрівання, та більше робить помилок при набору тексту. Також аналізуючи стандарти, є ще один параметр мікроклімату за яким варто слідкувати – рівень CO₂ при його значеннях від 300 до 600 ppm людина не бачить різниці, при значеннях від 600 до 1500 і вище, поступово може падати продуктивність праці, людина може стати млявою, збільшується кількість помилок, при високих показниках може виникнути головний біль. Стандарти CO₂ в Україні та ЄС, є однаковими. Щоб нормалізувати мікроклімат можна використовувати опалення, кондиціонування повітря, або вентиляцію.

Розділ 2. Дослідження технічного забезпечення систем контролю мікроклімату

2.1. Аналіз існуючих систем контролю параметрів мікроклімату

Для початку варто провести аналіз існуючих систем керування та моніторингу параметрів мікроклімату в приміщеннях. Серед існуючих рішень є як професійні так і так звані DIY (англ. DIY також D. I. Y. від англ. Do It Yourself – «зроби це сам»). Розглянемо кілька прикладів систем, які можливо розробити самостійно або купити в мережі Internet.

Одним із проєктів, які варто розглянути є DIY Монітор якості повітря–вимірювач PM2,5, CO₂, VOC, озону, температури та вологості на платформі Arduino [7].



Рисунок 2.1 – Система моніторингу якості повітря

Я поясню, як кожен із параметрів якості повітря впливає на нас і як працюють датчики. Головним компонентом цього проєкту є плата Arduino Pro Mini, яка в

поєднанні з 2,8-дюймовим сенсорним дисплеєм Nextion забезпечує гідний інтерфейс користувача.

В проєкті використовується датчик PM 2.5-PMS5003 який зображено на рисунку 2.2



Рисунок 2.2 – Датчик PM 2,5 PMS5003

Його було використано для вимірювання PM 2,5 (або Particulate Matter 2,5 – тверді частинки у повітрі діаметром приблизно 2,5 мікрона) Ці частки є найшкідливішою формою забруднення повітря, так як можуть проникати глибоко в легені, кровотік, мозок, викликаючи багато проблем зі здоров'ям людини. Датчик PM 2,5 працює за принципом лазерного розсіювання. Він має вентилятор, який створює контрольований потік, щоб частинки навколишнього середовища проходили через сфокусований лазерний промінь. Частинки викликають розсіювання світла, яке виявляється фотодіодом, а потім перетворюється на концентрацію твердих частинок за допомогою його мікропроцесора. Автор виявив, що результати цього датчика досить надійні, і разом із PM2.5 він також може видавати значення PM1 та PM10.

Далі використовується Датчик CO₂-MH-Z19 для вимірювання CO₂ або вуглекислого газу (зображений на рисунку 2.3)



Рисунок 2.3 – Датчик MH-Z19B для вимірювання CO₂

Оскільки люди під час дихання виділяють вуглекислий газ, концентрація CO₂ в приміщенні може легко стати дуже високою. Газ CO₂ не тільки небезпечний у високих концентраціях, але також може викликати сонливість, втому, знизити рівень нашої продуктивності, тощо.

Датчик використовує недисперсійний інфрачервоний принцип для вимірювання CO₂ у повітрі. Джерело інфрачервоного випромінювання направляє світло через трубку, наповнену повітрям, яке ми вимірюємо. З іншого боку від джерела інфрачервоного випромінювання є оптичний фільтр і ІЧ-детектор, який вимірює кількість ІЧ-світла, яке проходить крізь нього.

Молекули газу CO₂, присутні в повітрі, яке ми вимірюємо, поглинають певну смугу ІЧ-променів, пропускаючи хвилі певної довжини. Отже, рівень CO₂ розраховується відповідно до різниці між кількістю випромінюваного світла та кількістю ІЧ-світла, отриманого детектором. Результати цього датчика також досить точні.

Також в проєкті використовується Датчики VOC (VOC це легкі органічні сполуки) та озону MP503 та MQ131 зображені на рисунку 2.4.



Рисунок 2.4 – Датчик MQ131 - Модуль тестування виявлення газу (1) та датчик забруднення повітря ZP07 - MP503-10 VOC Модуль тестування якості повітря (2)

Принцип їх в тому, що через металеву підкладку проходить певний електричний струм, і опір які змінюються відповідно до кількості присутнього газу.

Датчик MQ131 є озон, який у звичайному домогосподарстві може утворюватись такими продуктами, як очищувачі повітря, відпарювачі для обличчя, бактерицидні лампи, що виділяють ультрафіолетове світло.

Датчик MP503 має кілька основних газів які може улавлювати – це алкоголь, дим, ізобутан, метанал та інші.

Для того, щоб зібрати схему потрібно такі комплектуючі-

Комплектуючі потрібні для збору проєкту:

- PMS5003 Датчик РМ
- МН-Z19 CO₂ датчик
- Датчик озону MQ-131
- Датчик VOC MP503
- DHT22 Датчик температури та шуму
- Дисплей Nextion 2,8
- DS3231 RTC
- Arduino Pro Mini
- Гайки М3
- Роз'єм Mini USB

- Штифтові роз'єми для Arduino
- 2-позиційний перемикач
- Конденсатори: 0.1 мкФ керамічний і 10 мкФ електролітичний
- Транзистор – 2N3904
- Перемикач

Датчик PM2.5 передає сигнал до Arduino через послідовний інтерфейс. Він працює при напрузі 5 В, але логічний рівень прийому RX працює при напрузі 3,3 В, тому для нього потрібен дільник напруги. Датчик CO₂ і дисплей Nextion також використовують послідовний зв'язок. Для зчитування датчиків VOC та озону ми використовуємо аналогові входи Arduino, тоді як датчик температури та вологості DHT22 використовує для цієї мети цифровий контакт.

Два транзистори використовуються для активації нагрівачів датчиків. Ми також використовуємо модуль годинника реального часу для відстеження часу під час зберігання значень датчиків, і він використовує зв'язок I2C . Весь пристрій живиться 5 В через роз'єм Mini USB.

Для завантаження коду в плату Arduino ми підключимо її до комп'ютера через USB інтерфейс. Зауважте, що для датчика VOC ми зчитуємо лише необроблені дані з цього датчика, а не значення часток на мільйон або частин на мільярд. Лише аналогові значення від 0 до 1024. Більш високі значення означають наявність ЛОС.

Що стосується датчика озону, щоб отримати точніші результати, ми повинні правильно встановити значення `setTimeToRead()` і `setR0()` відповідно до прикладу калібрування бібліотеки. Однак довший час `setTimeToRead` означає, що програма буде заблокована під час вибірки, а все інше буде заморожено.

Другим проектом буде більш простіше рішення - з назвою Монітор CO₂ [7]. Автора турбувала недостатня вентиляція в домі, тому він вирішив зібрати цей проект. Головною комплектуючими тут є плата Arduino UNO та датчик MQ-135 який може вимірювати якість повітря. Для того, щоб зібрати схему, потрібно використати такі комплектуючі як наведено в таблиці 2.1

Таблиця 2.1. Комплектуючі для збірки – Монітор CO₂

Arduino UNO	1 шт
Буквено-цифровий LCD дисплей, 16x2	1 шт.
Quad операційний підсилювач загального призначення від Texas Instruments	1 шт.
Датчик температури з терморезистором VMA-320	2 шт
Роз'єм для SD карт VMA – 304	1 шт
Резистор 10 кОм	1 шт
Резистор 1 кОм	1 шт
Резистор 221 Ом	1 шт

Ми провели аналіз вже існуючих систем контролю параметрів мікроклімату наприклад [7] де автор зібрав датчик CO₂ та контролю температури на основі мікроконтролера Arduino та датчика MQ-135.

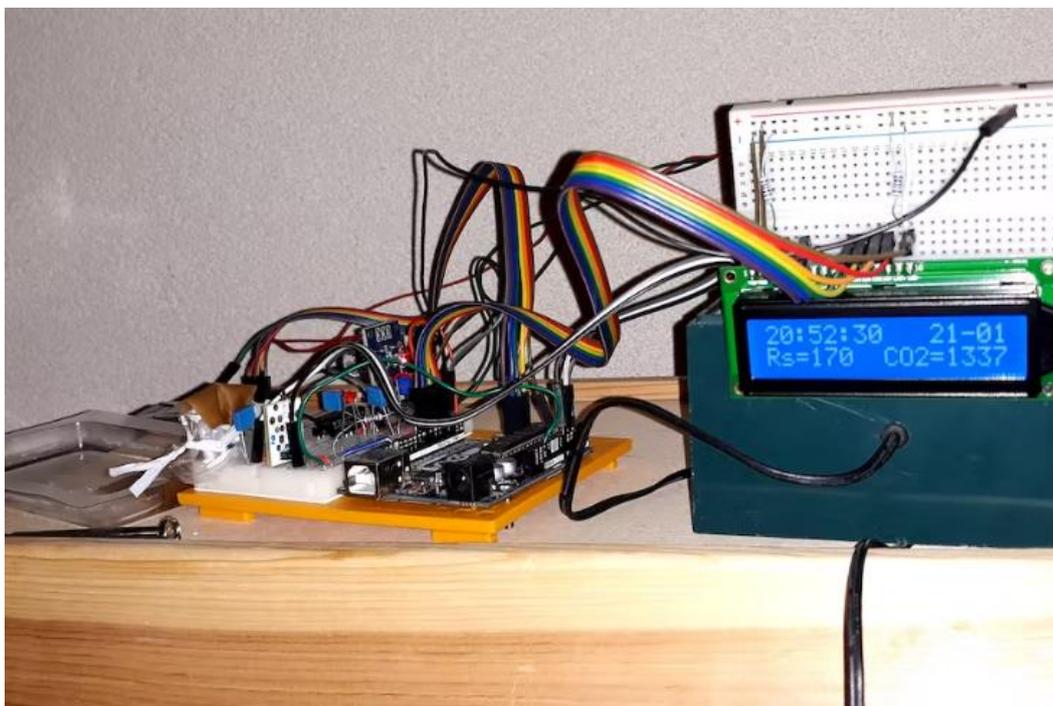


Рисунок 2.5 – Фото проекту автора з датчиком вуглекислого газу та LCD екраном.

Автор зазначив, що датчик MQ-135 досить чутливий, якщо в кімнату з установкою заходить людина то показники значно зміняться.

Також було проаналізовано різні проекти по контролю температури в приміщенні таких як це зробив автор тут [8].

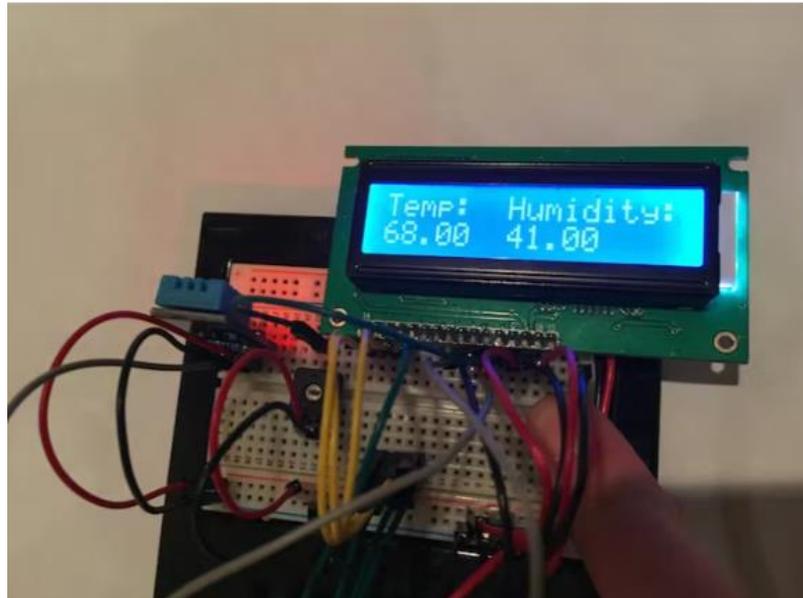


Рисунок 2.6 – Проект по контролю температури

Автор використав плату Arduino UNO, датчик температури DHT11 LCD дисплей та додаткові компоненти, такі як кнопки , резистор та проводи.

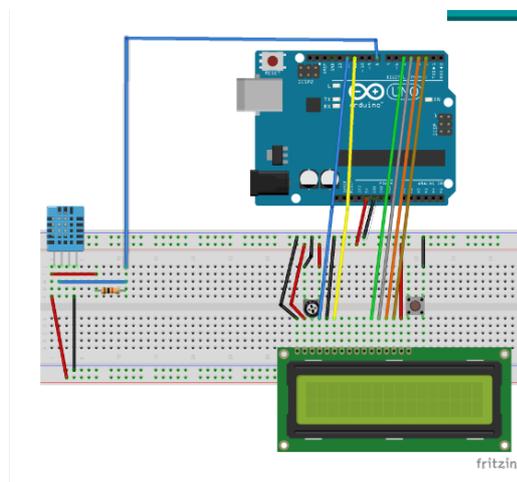


Рис.1.7 Схема проекту контролю температури на основі датчика DHT11

2.2. Аналіз існуючих вимірювальних засобів систем контролю параметрів мікроклімату

Для того, щоб проаналізувати вже існуючі системи контролю параметрів мікроклімату, я спробував знайти їх в відомих інтернет магазинах наприклад як [10] зображений на рисунку 2.8.



Рисунок 2.8 – Термогігрометр з датчиком вимірювання CO2SBTR PTH-9с

Як зазначено на сайті, термогігрометр можна використовувати в будинку, офісі та інших приміщеннях. Може вимірювати рівень вуглекислого газу, вологості повітря та температури. Визначення вмісту вуглекислого газу здійснюється за допомогою інфрачервоного недисперсійного методу, на основі якого працює датчик.

Переваги CO₂ PTH-9с в тому, що він

- Вимірює рівень вуглекислого газу в межах 400-5000 ppm
- Показує температуру: 10 ~ 40 °C (± 1 °C)

- Показує вологість: 20%-80% ($\pm 5\%$ відносної вологості)
- Принцип визначення рівня CO_2 заснований на технології NDIR, вимірювання відбувається за допомогою недисперсійного інфрачервоного датчика;
- Є портативним приладом;
- Працює на літєвій батареї 2400 мАг, що заряджається за допомогою Micro-USB кабелю, що входить до комплекту. Час заряджання 4-6 годин.
- Висота 98 мм, ширина 90, глибина 40мм, вага 300 г.
- Ціна 2 250 грн

Також є досить дорогий пристрій, а саме 13 972 грн, для вимірювання CO_2 , вологості та температури в приміщенні [2.3], також компактний, а саме – 66*120*33 та вагу 103 грами, що є ще більш легшим чим перший пристрій. Зображений на рисунку 2.9.



Рисунок 2.9 – Вимірювач рівня CO_2 TFA «AIRCO2NTROL 5000»

На дисплеї відображаються поточні рівні CO_2 , а також графік з попередніми значеннями та відповідними максимальними та мінімальними значеннями. Тимчасова шкала гістограми може бути змінена на одну хвилину, годину, день або

тиждень. Як варіант, останні виміряні значення температури та вологості також можуть бути чітко відображені на графіку.

Кольоровий дисплей світлофора дозволяє швидко інтерпретувати значення CO₂ та змінює колір із зеленого на жовтий та на червоний залежно від якості повітря та налаштування граничних значень. Якщо вказане максимальне значення перевищено, може бути надано додатковий попереджувальний сигнал, якщо рівні концентрації CO₂ надто високі.

Обширна функція реєстрації дозволяє зберігати більше 1 мільйона наборів даних (CO₂ / вологість / температура з датою і часом) на карті Micro-SD, що додається у вигляді файлу CSV. Інтервал стандартного вимірювання становить п'ять секунд. Збережені дані можна легко та швидко оцінити за допомогою Microsoft Excel. Живлення AirControl 5000 здійснюється через USB-кабель, що додається.

Характеристики:

- Діапазон вимірювання концентрації вуглекислого газу 0...5000 ppm (кількість частинок на мільйон)
- Роздільна здатність 1 ppm (0-1000), 5 ppm (1000-2000), 10 ppm (>2000)
- Інтервал поновлення 5 секунд
- Час відгуку < 30 секунд
- Спосіб вимірювання Non-Dispersive-Infrared Technology (NDIR)
- Діапазон вимірювання температури 0...+50°C
- Роздільна здатність 0,1 °C
- Діапазон вимірювання вологості 5...95%
- Роздільна здатність 1%
- Робоча температура 0...+50°C
- Температура зберігання -20...+60°C
- Годинник Кварцовий
- Матеріал - Пластик
- Колір білий

- Установка Підвішений або окремо стоїть
- Живлення Micro-USB
- Розміри (L) 120 x (B) 33 x (H) 66 мм
- Вага 103 г

Функції:

- Для моніторингу концентрації CO₂ у будинках, де присутні люди, наприклад, школи, офіси, громадські об'єкти
- Індикація рівня CO₂ у вигляді світлофора
- 2-канальний датчик NDIR з низьким дрейфом
- Відображення часу, температури та вологості у приміщенні
- Графік історичних значень (хвилина, година, день, тиждень)
- Макс.-мін.-функція
- Сигнал тривоги, що налаштовується
- Функція логгера зберігає понад 1 мільйон записів у вигляді файлу CSV на карті micro SD (у комплекті)
- Живлення через кабель micro-USB (входить до комплекту)

Комплектація:

- CO₂-монітор
- Карта micro SD
- Кабель micro-USB
- AC адаптер
- Інструкція з експлуатації
- Виробник: TFA Dostmann GmbH & Co. KG, Німеччина

Після пошуку вже готових пристроїв на інтернет ринку в Україні, можу сказати, що мною не було знайдено великого різноманіття цих пристроїв, є декілька марок та різні моделі з мінімальними змінами в них, серед них є досить непогані варіанти які можна встановити наприклад в офісі такий як CO₂SBTR PTH-9c, це не дорогий пристрій , а саме 2250 грн, функціоналом схожий на дорожчі аналоги. Але ми вирішили зібрати свій, так як головною його перевагою є модульність, завдяки

цьому можна змінювати як функціонал конструкції, так і ціну готового варіанту. Для початку потрібно оцінити характеристики модулів які ми вибрали для цього проекту.

2.2.1. Модуль датчика якості повітря MQ-135

MQ-135 згідно з [12] в основному використовується для визначення концентрації вуглекислого газу, алкоголю, бензолу, оксидів азоту, аміаку та інших газів у повітрі. На платі модуля передбачений компаратор для визначення порогового значення концентрації шкідливих речовин. Поріг спрацьовування задається потенціометром[13].

Характеристики:

- Вихідний сигнал TTL рівня
- Аналоговий вихід датчика 0 - 5В. Чим вище концентрація газів тим вище напруга на виході
- Напруга живлення модуля: 5В постійного струму
- Розміри: 32мм X22мм X27мм
- Тип датчика: MQ-135
- Ціна цього модуля варіюється від 65грн(Prom) до 229 грн на сайті evo.net.

Датчик наведений на рисунку 2.10.



Рисунок 2.10 – Модуль датчика якості повітря MQ-135

Цей датчик підключається до Плати Arduino згідно розпіновки зазначеної на таблиці 2.2.

Таблиця 2.2. Підключення датчика до Arduino

MQ-135	Arduino
VCC	5V
AOUT	A0
GND	GND

Зразок програми написаної для датчика MQ 135

```
const int gasSensor =0;
void setup(){
    Serial.begin(9600);// Встановлює на послідовному порті значення 9600
}
void loop(){
    float voltage;
    voltage = getVoltage(gasSensor);
    Serial.println(voltage);
    delay(1000);
}
float getVoltage(int pin){
    return (analogRead(pin) * 0.004882814);
}
```

Перед тим, як почати використовувати датчик, його потрібно підключити до напруги і попередньо прогріти протягом 12-24 годин. Після цього потрібно виконати наступні процедури в середовищі з температурою в 20 градусів С/35% температури повітря та прочитати значення строки gzero. Але для початку потрібно написати декілька рядків коду, щоб датчик працював.

```
#include "MQ135.h"
const int ANALOGPIN=0;
```

```

MQ135 gasSensor = MQ135(ANALOGPIN);
void setup(){
Serial.begin(9600);// Встановлює на послідовному порті значення 9600
}
void loop(){
float rzero = gasSensor.getRZero();
Serial.println(rzero);
delay(1000);

```

Після того як ми визначили значення строки rzero потрібно його записати в файл з назвою MQ135.h, у автора вийшло значення RZERO 76.63.

```

#include "MQ135.h"
const int ANALOGPIN=0;
MQ135 gasSensor = MQ135(ANALOGPIN);
void setup(){
Serial.begin(9600);
}
void loop(){
float ppm = gasSensor.getPPM();
Serial.println(ppm);
delay(1000);}

```

Після цього написавши даний код можна вже починати вимірювати якість повітря.

- Експерименти показують, що MQ-135 може виявляти вищезгадані гази, але не розрізняти їх. Якщо ви хочете перевірити вміст певного газу в повітрі, можливо, вам знадобиться знайти інші кращі датчики.

- MQ-135 використовує нагрівальний модуль для попереднього нагрівання датчика, тому передбачається, що блок живлення з більшою потужністю не розрядився швидко.

2.2.2. Датчик CO₂ MH-Z19B

Далі був вибраний нами датчик CO₂ MH-Z19B, згідно з [14], MH-Z19B — універсальний невеликий датчик, який дозволяє використовувати недисперсійне інфрачервоне випромінювання (NDIR) для виявлення вуглекислого газу в повітрі з хорошою вибірковістю. Цей датчик простий у використанні та має вбудований датчик температури, який може здійснювати температурну компенсацію. Цей датчик має послідовний інтерфейс UART (англ. universal asynchronous receiver/transmitter — універсальний асинхронний приймач/передавач) і виходи широтно-імпульсної модуляції. Його можна широко використовувати в системах HVAC (Опалення, вентиляція та кондиціонування повітря, скор. ОВК, або HVAC від англ. Heating, Ventilation, & Air Conditioning), моніторингу якості повітря в холодильних приміщеннях, датчик зображений на рисунку 2.11.

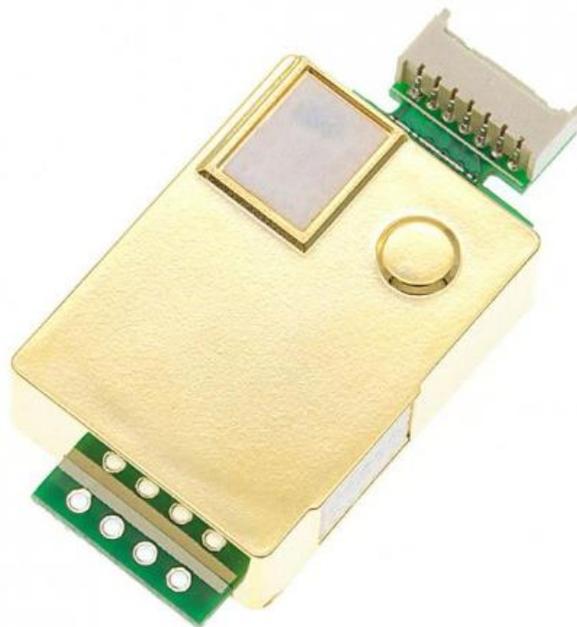


Рисунок 2.11. Датчик CO₂ MH-Z19B

Застосування:

- Побутові та промислові вимірювачі концентрації CO₂

Особливості:

- Висока чутливість і роздільна здатність
- Низьке споживання
- Вихідні інтерфейси: UART і ШІМ
- Температурна компенсація і чудова лінійність
- Хороша стабільність
- Довгий термін служби

Характеристики:

- Модель датчика: MH-Z19B
- Вимірюваний газ: CO₂
- Робоча напруга: від 3.6 до 5.5 В
- Напруга логічних рівнів: 3,3
- Середній споживаний струм: <18 мА
- Рівень логічних сигналів: 3.3 В
- Вимірюваний діапазон: від 0 до 0.5%
- Вихідні сигнали: UART, PWM
- Час розігріву: 3 хвилини
- Час вимірювання: T90 <60 s
- Робоча температура: від 0 до 50°
- Робоча вологість: від 0 до 95% RH
- Розміри: 33 мм × 20 мм × 9 мм
- Вага: 21 грам
- Гарантована напрацювання на відмову:> 5 років
- Ціна від 700 до 1250 грн[15].

Для того щоб виміряти температуру, вологість і концентрацію вуглекислого газу, ми підключимо датчик температури DHT і датчик MHZ-19B до Arduino Uno і відобразимо показання на модулі OLED-дисплея SSD1306 i2c.

Після підключення модулів до Arduino , потрібно встановити бібліотеку MH-Z19. Далі вже можна починати завантажувати програму для роботи датчика. Програмування MH-Z19B Arduino:

```
// CO2
#include <Arduino.h>
#include <MHZ19.h>
#include <SoftwareSerial.h>
// Temp
#include <DFRobot_DHT20.h>
// Display
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h >
#include <Wire.h>
// CO2
#define RX_PIN 3 // Rx контакт MHZ19 Tx приєднаний до D2
#define TX_PIN 2 // Tx контакт MHZ19 Rx приєднаний до D3
#define BAUDRATE 9600 // Пристрій до MH-Z19 Послідовна швидкість
передачі (не слід змінювати)
MHZ19 myMHZ19; // Конструктор для бібліотеки
SoftwareSerial mySerial(RX_PIN, TX_PIN); // (Без прикладу) створити
пристрій для послідовного MH-Z19
// Температура
DFRobot_DHT20 dht20;
// Дисплей
#define SCREEN_WIDTH 128 // Ширина дисплея OLED, у пікселях
#define SCREEN_HEIGHT 64 // Висота дисплея OLED, у пікселях
// Декларація для дисплея SSD1306, підключеного до I2C (SDA, SCL
контакти)
#define OLED_RESET -1 // Скинути PIN-код # (або -1, якщо спільний контакт
Arduino для скидання)
```

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);
```

```
void setup()
{
  Serial.begin(9600);// Запуск CO2
  mySerial.begin(BAUDRATE);// (Без прикладу) пристрій для послідовного
запуску MH-Z19
  myMHZ19.begin(mySerial); // *Посилання Serial(Stream) необхідно передати
в бібліотеку begin().
  myMHZ19.autoCalibration(); // Увімкнути автоматичне калібрування (OFF
autoCalibration(false))
  // Start Temp
  while (dht20.begin()) {
    Serial.println("Помилка ініціалізації датчика");
    затримка (1000);
  }
  // Запуск відображення
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  затримка (2000);
  display.clearDisplay();
  display.setTextColor(WHITE);}
void loop()
{
  //Отримати CO2
  int CO2;
  CO2 = myMHZ19.get CO2 ();// Запит CO2 (як проміле)
  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(WHITE);
```

```

display.setCursor(0, 0);
display.print(dht20.getTemperature());
display.print("\n");
display.print(dht20.getHumidity());
display.print("%");
display.print("\n");
display.print(CO2);
display.print(" ppm");
display.display();
затримка (2000);}

```

Після завантаження коду концентрація вуглекислого газу та температура будуть відображатися на OLED.

2.2.3. Датчик CO₂ (Arduino-сумісний) від DFRobot

Це перший датчик CO₂ на ринку Arduino. Вихідна напруга модуля падає зі збільшенням концентрації CO₂. Потенціометр на платі призначений для встановлення порогу напруги. Поки концентрація CO₂ є достатньо високою (напруга нижча за порогове значення), буде випущено цифровий сигнал (ON/OFF).

Він має сенсорний модуль MG-811, який дуже чутливий до CO₂ і менш чутливий до алкоголю та CO, низької вологості та залежності від температури. Оснащений світлодіодною індикацією для відображення граничного значення концентрації вуглекислого газу. Analog CO₂ Gas Sensor – електрохімічний датчик, тому його обов'язково необхідно відкалібрувати перед фактичним виміром. Працюючи датчик гріється, це нормально.

- Бортовий контур опалення забезпечує найкращу температуру для роботи датчика. Внутрішнє підвищення напруги до 6 В для найкращої роботи датчика нагріву.
- Цей датчик має вбудовану схему кондиціювання для посилення вихідного сигналу[16].

Датчик зображений на рисунку 2.12.



Рисунок 2.12. Датчик SKU:SEN0159

Характеристики цього датчика зазначені нижче-

- Робоча напруга: 4,5 ~ 5,5 В постійного струму
- Виявлення газу: вуглекислий газ (CO₂)
- Середній струм: <85 мА
- Вихідний сигнал: UART/PWM/аналоговий вихід (останні два не розроблені)
- Діапазон вимірювання: 0 ~ 50000ppm
- Точність: ± (50 ppm + 5% показання)
- Час попереднього нагріву: 3 хв
- Час відгуку: T90 < 30 с
- Робоча температура: 0 ~ 50 °С
- Робоча вологість: 0 ~ 95% RH (без конденсації)
- Термін служби: >5 років
- Розмір плати: 21 * 27,11 мм / 0,83 * 1,06 дюйма

- Вага: 42 г
- Високоякісний роз'єм DFRobot Gravity
- Вбудована схема нагрівача датчика
- Комплектується зручним кабелем для підключення
- Ціна колеблється від 1687 до 1766 грн

ОСОБЛИВОСТІ

- Висока чутливість Висока роздільна здатність
- Низьке енергоспоживання
- Відмінна стійкість
- Температурна компенсація
- Відмінний лінійний вихід
- Високий цикл життя
- Захист від водяної пари
- Ніякого отруєння
- Потрібне джерело живлення 5 В, навіть якщо ви використовуєте мікроконтролер 3,3 В, наприклад Genuino 101/ Raspberry Pi тощо.

Калібрування модуля проходить деякий час під час цього слід забезпечити стабільне живлення цього модуля, під час роботи датчик буде нагріватися. Потрібно, розмістити цей модуль у місці з чистим повітрям. Після безперервної роботи протягом приблизно 48 годин можна виміряти вихідну напругу цього модуля. Потім потрібно змінити визначення в коді, розділивши значення напруги (одиниця: В) на 8,5.

```
#define ZERO_POINT_VOLTAGE (voltage/8.5)
```

Наприклад, напруга, яку ви виміряли на модулі, становить 2,4 В, тоді $2,4/8,5=0,282$. Тому заміняємо визначення, як показано нижче:

```
#define ZERO_POINT_VOLTAGE (0.282)
```

Після модифікації завантажте зразок коду на свою плату Arduino.

Давайте розглянемо запропонований розробником даного модуля код, який можна поділити на певні структурні частини.

Автор зазначає, що цей фрагмент вихідного коду має використовуватися лише як демонстрація, більш складне калібрування необхідне для промислового застосування.

У першій частині коду наводяться макроси, пов'язані з обладнанням.

```
#define MG_PIN (A0) //визначте, який аналоговий вхідний канал ви збираєтеся
використовувати
#define BOOL_PIN (2)
#define DC_GAIN (8.5) //визначення посилення постійного струму підсилювача
/*****Макроси, пов'язані з програмним забезпеченням*****/
#define READ_SAMPLE_INTERVAL (50) //визначте, скільки зразків ви збираєтеся
взяти під час нормальної роботи
#define READ_SAMPLE_TIMES (5) //визначте часовий інтервал (у мілісекундах)
між кожними зразками в
//нормальна робота
/*****Макроси, пов'язані з програмою*****/
//Ці два значення відрізняються від датчика до датчика. користувач повинен
визначити це значення.
#define ZERO_POINT_VOLTAGE (0,220) //визначення вихідного сигналу датчика
у вольтах, коли концентрація CO2 становить 400 PPM
#define REACTION_VOLTAGE (0,030) //визначення падіння напруги датчика при
переміщенні датчика з повітря в 1000ppm CO2
/*****Основні макроси*****/
float CO2Curve[3] =
{2,602,ZERO_POINT_VOLTAGE,(REACTION_VOLTAGE/(2,602-3))};
//з кривої беруться дві точки
//з цими двома точками утворюється лінія, яка є
//приблизно еквівалентна вихідній кривій
//формат даних (x,y,slope); point: (lg400, 0.324)
//нахил = ( reaction voltage ) / (log400 –log1000)
void setup()
```

```

{
Serial.begin(9600); //Налаштування UART, швидкість передачі = 9600 біт/с
pinMode(BOOL_PIN, INPUT); //встановити пін для введення
digitalWrite(BOOL_PIN, HIGH); // увімкнути підтягувальні резистори
Serial.print("MG-811 Demistration\n");
}
void loop()
{
int percentage;
float volts;
volts = MGRead(MG_PIN);
Serial.print( "SEN0159:" );
Serial.print(volts);
Serial.print("V");
Percentage = MGGetPercentage(volts,CO2Curve);
Serial.print("CO2:");
If (percentage == -1) {
Serial.print( "<400" );
} else {
Serial.print(percentage);
}
Serial.print( "ppm");
Serial.print("\n");
if (digitalRead(BOOL_PIN) ){
Serial.print( "====BOOL is HIGH====" );
} else {
Serial.print( "====BOOL є LOW====" );
}
Serial.print("\n");
delay (500);

```

}

/*****MGRead *****/

Вхід: mg_pin - аналоговий канал

Вихід: вихід SEN-000007

Примітки: ця функція читає вихід SEN-000007

*****/

float MGRead(int mg_pin)

{

int i;

float v=0;

for (i=0;i<READ_SAMPLE_TIMES;i++) {

v += analogRead(mg_pin);

delay (READ_SAMPLE_INTERVAL);

}

v = (v/READ_SAMPLE_TIMES) *5/1024 ;

return v;

}

/*****MQGetPercentage *****/

Вхід: вольти - вихід SEN-000007 вимірюється у вольтах

pcurve - покажчик на криву цільового газу

Вихід: ppm цільового газу

Примітки: за допомогою нахилу та точки прямої. X (логарифмічне значення проміле) рядка може бути отримано, якщо надано у(вихід MG-811). Як це а логарифмічна координата, ступінь 10 використовується для перетворення результату в нелогарифмічний значення.

*****/

int MGGetPercentage(float volts, float *pcurve)

{

if ((volts/DC_GAIN)>=ZERO_POINT_VOLTAGE) {

return -1;

```

} else {
return pow(10, ((volts/DC_GAIN)-pcurve[1])/pcurve[2]+pcurve[0]);

```

2.2.4. Датчик вологості та температури DHT22 та DHT11

Згідно з [17] датчики DHT22 та DHT11 дуже популярні серед любителів електроніки, оскільки вони дуже дешеві, але все ще забезпечують чудову продуктивність. Ось основні характеристики та відмінності між цими двома датчиками:

DHT22 є дорожчою версією, яка, очевидно, має кращі характеристики. Його діапазон вимірювання температури становить від -40 до +125 градусів за Цельсієм з точністю $\pm 0,5$ градуса, тоді як діапазон температур DHT11 становить від 0 до 50 градусів за Цельсієм з точністю ± 2 градуси. Крім того, датчик DHT22 має кращий діапазон вимірювання вологості від 0 до 100% з точністю 2-5%, тоді як діапазон вологості DHT11 становить від 20 до 80% з точністю 5%. Датчики зображені на рисунку 2.13



Рисунок 2.13 – DHT11 зліва, DHT22 справа

Також порівняння характеристик параметрів наведено в таблиці 2.3

Таблиця 2.3. Порівняння характеристик датчиків DHT11 та DHT22

DHT11	Датчик	DHT22
02-50C/+2C	Температурний діапазон	-40-125C/+0,5 C
20-80%/+-5	Діапазон вологості	0-100%/+-2-5%
1Hz (одне читання кожну секунду)	Частота вимірювання	0,5 Hz (одне читання кожні дві секунди)
15.5мм x12мм x5.5мм	Розмір	15.1mm x 25mm x 7.7mm
3-5В	Робоча напруга	3-5В
2,5mA	Максимальний струм під час вимірювання	2,5mA

Є дві специфікації, де DHT11 кращий за DHT22. Це частота дискретизації, яка для DHT11 становить 1 Гц або одне зчитування щосекунди, тоді як частота дискретизації DHT22 становить 0,5 Гц або одне зчитування кожні дві секунди, а також DHT11 має менший розмір корпусу. Робоча напруга обох датчиків становить від 3 до 5 вольт, а максимальний струм, що використовується при вимірюванні, становить 2,5 мА.

Модулі складаються з датчика вологості, та датчика температури NTC (або термістора) та мікросхеми на задній стороні датчика – рисунок 2.14.

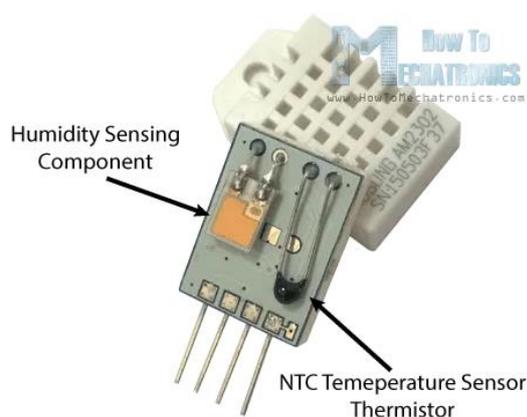


Рисунок 2.14 – Будова модуля (зліва датчик вологості, справа датчик температури)

Для вимірювання вологості вони використовують компонент датчика вологості, який має два електроди з вологостійкою підкладкою між ними. Тому зі зміною вологості змінюється провідність підкладки або змінюється опір між цими електродами. Ця зміна опору вимірюється та обробляється мікросхемою, що робить її готовою до зчитування мікроконтролером. З іншого боку, для вимірювання температури ці датчики використовують датчик температури NTC або термістор.

Термістор – це фактично змінний резистор, який змінює свій опір зі зміною температури. Ці датчики виготовляються шляхом спікання напівпровідникових матеріалів, таких як кераміка або полімери, щоб забезпечити більші зміни опору лише з невеликими змінами температури. Термін «NTC» означає «негативний температурний коефіцієнт», що означає, що опір зменшується зі збільшенням температури (Рисунок 2.15.).

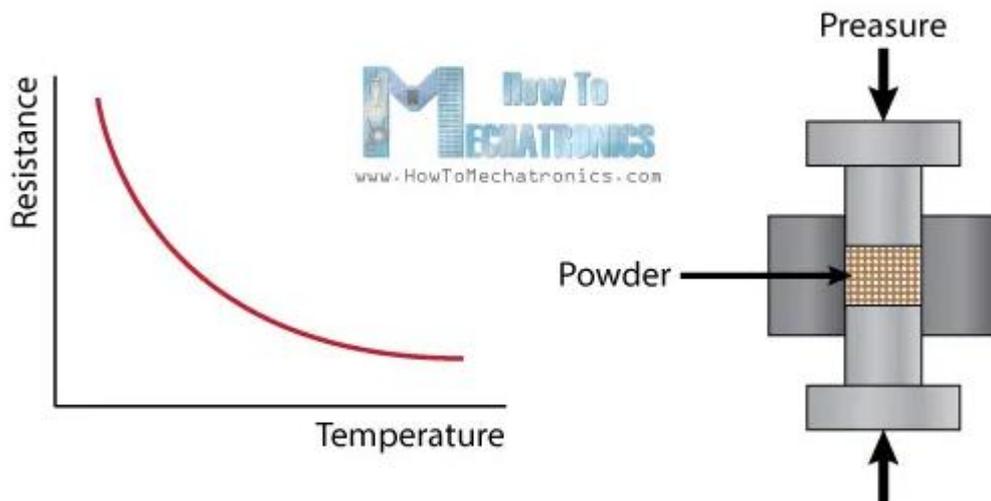


Рисунок 2.15 – Принцип дії датчика

Датчики DHTxx мають чотири контакти, VCC, GND, контакт даних і не підключений контакт, який не використовується. Підтягуючий резистор від 5 до 10 кОм потрібен для підтримки високого рівня лінії передачі даних і забезпечення зв'язку між датчиком і платою Arduino. Є деякі версії цих датчиків, які

постачаються з платами розриву з вбудованим підтягуючим резистором і мають лише 3 контакти.

Датчики DHTXX мають власний однопровідний протокол, який використовується для передачі даних. Для цього протоколу потрібен точний час, а часові діаграми для отримання даних від датчиків можна знайти в таблицях даних датчиків. Однак нам не потрібно особливо турбуватися про ці часові діаграми, тому що ми будемо використовувати бібліотеку DHT, яка подбає про все.

Спочатку нам потрібно встановити бібліотеку DHT, яку можна знайти на офіційному веб-сайті Arduino, потім визначити номер контакту, до якого підключено наш датчик, і створити об'єкт DHT. У розділі налаштування нам потрібно ініціювати послідовний зв'язок, оскільки ми будемо використовувати послідовний монітор для друку результатів. Використовуючи функцію `read22()`, ми будемо читати дані з датчика та помістити значення температури та вологості у змінні `t` і `h`. Якщо ви використовуєте датчик DHT11, вам знадобиться функція `read11()`. Наприкінці ми роздрукуємо значення температури та вологості на серійному моніторі.

Щоб почати використовувати модуль, потрібно написати код та завантажити на нього –

```
#include <dht.h>
#define dataPin 8 // Визначає номер контакту до якого підключається датчик
dht DHT; // Створює об'єкт з назвою DHT
void setup() {
  Serial.begin(9600);
}
void loop() {
  int readData = DHT.read22(dataPin); // Зчитує дані з датчика
  float t = DHT.temperature; // Отримує значення про температуру
  float h = DHT.humidity; // Отримує значення про вологість
  // Друкує результати на монітор
  Serial.print("Temperature = ");
```

```
Serial.print(t);  
Serial.print(" *C ");  
Serial.print("Humidity = ");  
Serial.print(h);  
Serial.println(" % ");  
delay(2000); // Delays 2 secods, as the DHT22 sampling rate is 0.5Hz  
}
```

Після того, як ми завантажимо цей код на плату Arduino, результати температури та вологості від датчика можна буде побачити на моніторі послідовного порту.

2.3. Висновок до розділу 2

Провівши аналіз наявних рішень по аналізу мікроклімату в інтернеті, ми визначили, що є DIY варіанти, та вже готові серійні пристрої. Більшість саморобних варіантів базується на платі Arduino, їх перевага в тому, що можна додавати ті датчики, які необхідні в тій чи другій ситуації. Також такі проекти будуть дешевше ніж готові пристрої представлені в інтернет магазинах, ціна за готовий варіант буде від 2000 до 13000 грн. Перевага готового варіанта в його компактності, та в тому, для його установки та використання не потрібні навички в програмуванні. Для цієї магістерської роботи ми вирішили зробити власний проект для аналізу мікроклімату, аргументуючи це тим, серійні пристрої досить дорогі, та для саморобного пристрою можна використати ті датчики, які більш підходять саме нам.

Розділ 3. Розроблення конфігурації системи моніторингу та керування параметрами мікроклімату в приміщеннях

На перших етапах розроблення системи нами було реалізовано першу версію конфігурації системи для проведення досліджень по зміні параметрів мікроклімату, яку було представлено на VIII Всеукраїнській науково-практичній конференції «Електронні та мехатронні системи: теорія, інновації, практика» [18]. Подальші дослідження показали ряд недоліків даної конфігурації і було прийнято рішення її модифікувати.

Процес побудови системи моніторингу та керування параметрами мікроклімату було вирішено розділити на розробку кількох підсистем, а саме підсистему вимірювання температури та вологості середовища, підсистема індикації, підсистема вимірювання рівня діоксиду вуглецю, підсистема керування виконавчими пристроями. Такий розподіл дозволить реалізувати дискретну систему, і кожна підсистема може бути модифікована або замінена.

3.1. Підсистема моніторингу температури та вологості навколишнього середовища

У процесі проведення аналізу існуючих технічних засобів нами було виділено кілька вимірювальних пристроїв вимірювання температури та вологості, представленим в розділі 2.2.4. Ці пристрої були обрані з урахуванням критеріїв доступності на українському ринку вимірювальних пристроїв, точності та економічної доцільності. Вимірювальні пристрої, які було обрано для можливості імплементації – DHT22 та DHT11.

Перевагами DHT11 є низька ціна (30 – 50 грн.), та менша, у порівнянні з модулем DHT22, періодичність отримання даних (1 зчитування протягом 1 секунди). Ціна DHT22 перевищує ціну модуля DHT11 приблизно в 3 рази, а частота отримання даних в два рази менша (1 зчитування протягом 2 секунд), відповідно до технічної документації. Натомість похибка вимірювань температури

вимірювального пристрою DHT22 приблизно в 4 рази менша за похибку вимірювань DHT11, та похибка вимірювань DHT22 вологості приблизно в 2 рази менша за DHT11. З урахуванням даних фактів нами було обрано використання модуля DHT22. Детальний порівняльний аналіз технічних характеристик DHT11 та DHT22, а також принцип дії даних пристроїв описано в розділі 2.2.4.

Підключення підсистеми вимірювання температури зображено на рисунку 3.1.

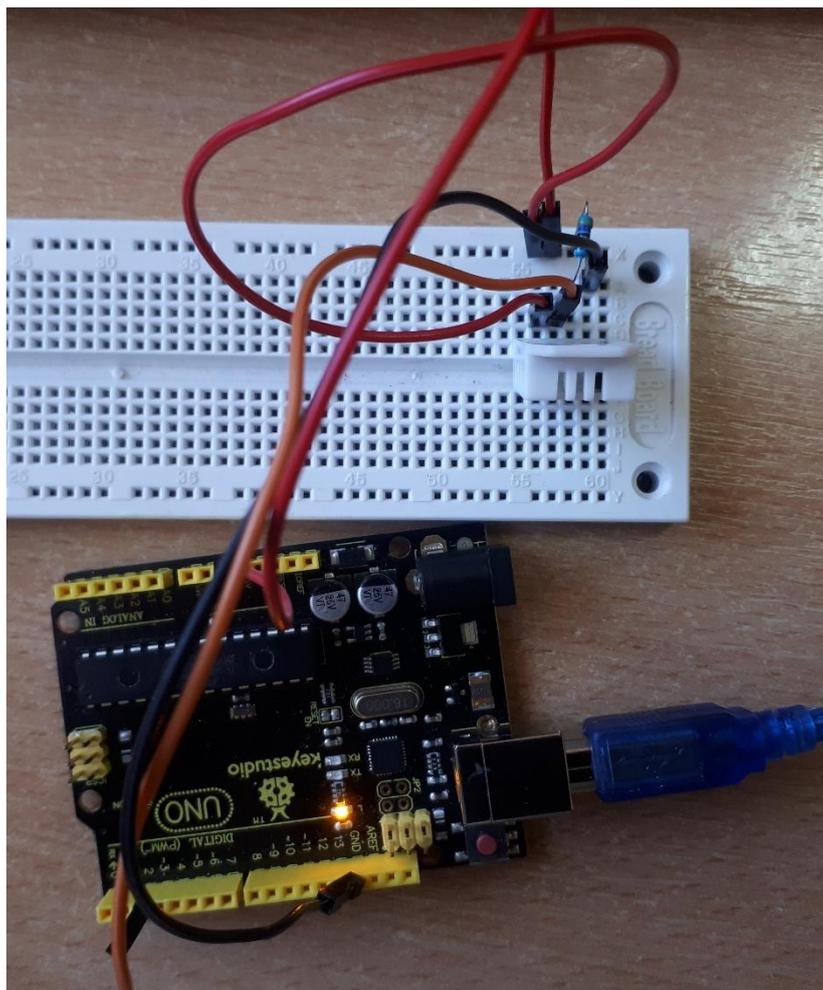


Рисунок 3.1 – Ілюстрація конфігурації підключення елементів підсистеми вимірювання температури та вологості навколишнього середовища

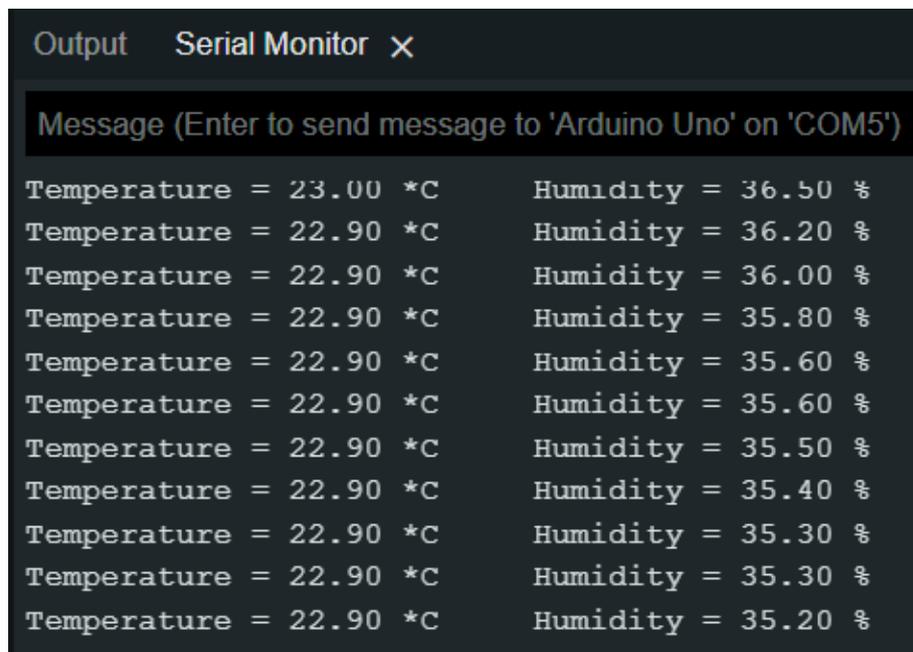
Згідно з [17] при підключенні датчика було використано підтягуючий резистор з опором 4.7 кОм. Загальне підключення датчика до плати керування наступне

(виводи нумеруються зліва на право, якщо дивитись на модуль з фронтальної сторони):

- 1 контакт підключений до живлення, на платі керування – 5V;
- 2 контакт підключений підтягуючий резистор 4.7 кОм, що підключений до виводу живлення 5V, а також підключений до 2-го цифрового виводу на платі керування (провідник з помаранчевою ізоляцією на рисунку 3.1);
- 3 контакт не підключений, не використовується;
- 4 контакт підключений до мінусового контакту джерела живлення, на платі керування – GND.

При розробці програмного забезпечення було здійснено тестування кількох бібліотек, які дозволяють забезпечити коректну роботу вимірювального пристрою. Було здійснено перевірку програм, наведених в [17, 19], обидві бібліотеки дають змогу здійснювати вимірювання параметрів температури та вологості тому нами було використано програмний код наведений в науково-популярній статті [17].

Результатом роботи підсистеми є відображення даних за допомогою монітору послідовного порту, яка продемонстрована на рисунку 3.2.



The image shows a screenshot of the 'Serial Monitor' window in an IDE. The title bar reads 'Output Serial Monitor X'. Below the title bar, there is a header 'Message (Enter to send message to 'Arduino Uno' on 'COM5')'. The main area displays a series of data points for temperature and humidity. The temperature values start at 23.00 °C and then stabilize at 22.90 °C. The humidity values start at 36.50 % and then decrease to 35.20 %.

Temperature	Humidity
23.00 *C	36.50 %
22.90 *C	36.20 %
22.90 *C	36.00 %
22.90 *C	35.80 %
22.90 *C	35.60 %
22.90 *C	35.60 %
22.90 *C	35.50 %
22.90 *C	35.40 %
22.90 *C	35.30 %
22.90 *C	35.30 %
22.90 *C	35.20 %

Рисунок 3.2 – Передача вимірюваних даних за допомогою вимірювального пристрою через протокол послідовного порту

3.2. Підсистема індикації параметрів навколишнього середовища

На перших етапах проєктування варто проаналізувати ті ресурси, які можуть знадобитись при реалізації різних підсистем. Так як однією із підсистем, яка може використовувати значні ресурси з точки зору підключення і використання цифрових виводів плати керування є підсистема індикації. Розглянемо варіанти реалізації даної підсистеми.

Першим етапом є вибір формату та технічних засобів підсистеми індикації. У різних системах моніторингу можуть використовуватись різні технічні засоби, серед яких:

- Світлодіодні системи індикації: використовуються в системах, де немає необхідності відображати значення параметрів, що вимірюються, або відображати значення параметрів за допомогою умовного поділу на рівні або стани, що залежать від значень величини, що вимірюється.

- Звукові системи індикації: використовуються в системах, де необхідно повідомляти про зміну стану за допомогою звукового сигналу, що може означати наявність аварійних режимів роботи або режимів, на які оператор системи повинен зреагувати.

- Системи з візуалізацією даних: подібні системи дозволяють відображати дані у числовому вигляді, часових характеристик або навіть відображати технологічний процес у вигляді інтерактивної моделі. Технічні засоби, що використовуються у подібних системах візуалізації зазвичай коштують значно дорожче, аніж у світлодіодних та звукових системах індикації.

Аналізуючи наявні формати підсистем індикації з перспективи реалізації системи моніторингу та керування параметрами мікроклімату, нами було обрано один з найбільш дешевих рідкокристалічних дисплеїв, а саме: LCD 1602 символний дисплей 16x2. Перевагою даного пристрою полягає в тому, що він дозволяє виводити інформацію у форматі символних повідомлень на 2 рядках, кожен з яких може виводити до 16 символів, а також ціна, що коливається в межах 100 – 120 грн.

Даний дисплей зображений на рисунку 3.3.



Рисунок 3.3 – LCD 1602 символний дисплей

Схема підключення даного дисплею наведена на рисунку 3.4.

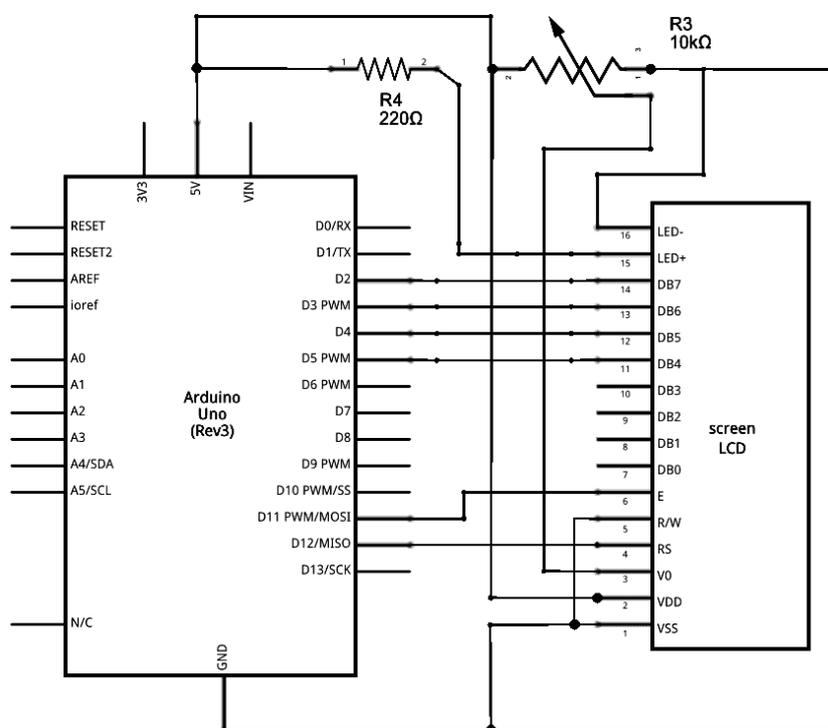


Рисунок 3.4 – Схема підключення

Проблема реалізації даного підключення полягає у тому, що воно вимагає використання 6 цифрових виводів плати керування і це є незадовільними умовами для подальшої реалізації інших підсистем, хоча і досить економічним, зважаючи, що окрім потенціометра R3 на 10 кОм і струмообмежувального резистора дана конфігурація не вимагає інших елементів. Враховуючи необхідність використання вимірювальних пристроїв температури, вологості та рівня діоксиду вуглецю було вирішено мінімізувати кількість цифрових виводів для візуалізації даних. Для цього існує можливість підключення дисплею до плати керування з використанням I2C модуль розширення виводів Arduino для підключення LCD дисплея на PCF 8574T (ціна складає 50-60 грн.), який зображений на рисунку 3.5.

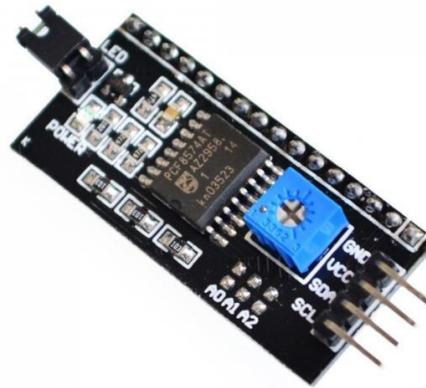


Рисунок 3.5 – I2C модуль розширення виводів Arduino для підключення LCD дисплея на PCF 8574T

Даний модуль розширення дозволяє використовувати всього лише два виводи SDA та SCL (окрім виводів живлення Vcc і GND) для забезпечення передачі даних від мікроконтролера до плати керування. Детальний опис підключення та принципу роботи дисплею в комбінації з платою розширення наведено в науково-популярній статті [20].

Загальне підключення LCD дисплею із використанням плати розширення з мікросхемою PCF 8574T зображено на рисунку 3.6.

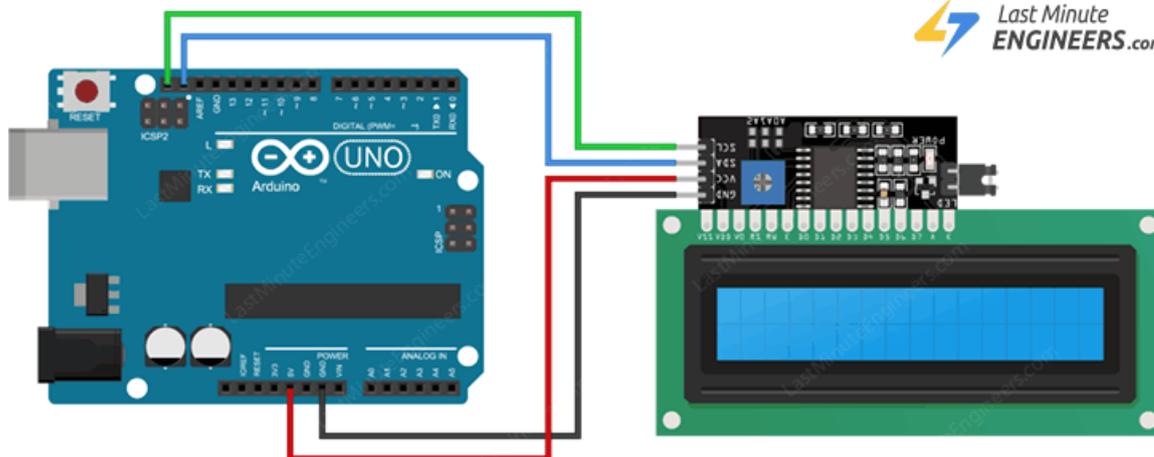


Рисунок 3.6 – Підключення плати керування Arduino до підключення LCD дисплея з використанням плати з мікросхемою PCF 8574T

Програмний код, описаний в розділі 2.2.4, який було взято за основу і який був використаний для реалізації підсистеми вимірювання температури та вологості, був модифікований при додаванні підсистеми індикації.

Для забезпечення взаємодії з дисплеєм в програмному коді було використано бібліотеки `Wire.h` та `LiquidCrystal_I2C.h`. Для керування дисплеєм в програмному коді необхідно оголосити глобальний об'єкт класу шляхом додавання наступного коду `LiquidCrystal_I2C lcd(0x27,16,2)`; де `LiquidCrystal_I2C` – ім'я класу, `lcd` – ім'я об'єкту, `0x27` – це I2C адреса пристрою, `16` – кількість символів, які можливо записати в одному рядку, `2` – кількість рядків дисплея.

До функції `setup` необхідно також додати виклики кількох методів об'єкта `lcd`, а саме `lcd.init()`; `lcd.clear()`; `lcd.backlight()`; кожен з яких виконує наступні функції відповідно: ініціалізація `lcd`, очищення дисплея, увімкнення підсвітки дисплея.

До функції `loop` було додано наступний програмний код:

```
lcd.setCursor(0,0);
lcd.print("Temp. : ");
lcd.print(t);
lcd.print(" C ");
lcd.setCursor(0,1);
```

```
lcd.print("Humi. : ");  
lcd.print(h);  
lcd.print(" % ");
```

Цей код призначений для виведення даних отриманих від модуля DHT22 і записаних у змінні *t* (температура) та *h* (вологість). Метод `setCursor` дозволяє встановити місце на дисплеї з якого буде відбуватись виведення даних. Перший параметр, який дорівнює 0 визначає номер стовпця (всього їх 16, але нумерація починається з 0), а другий параметр, який теж дорівнює 0 – визначає номер рядка (нумерація теж починається з 0). Таким чином виведення даних відбуватиметься з лівого верхнього кутка дисплея. Метод `print` безпосередньо виконує функцію виведення даних. Окрім цього після затримки в 2 секунди, яка необхідна для коректної роботи DHT22, було додано виклик метода `clear()`, для очищення дисплея від попередніх значень.

Результат реалізації підсистеми індикації разом із роботою підсистеми вимірювання температури та вологості зовнішнього середовища продемонстрована на рисунку 3.7.

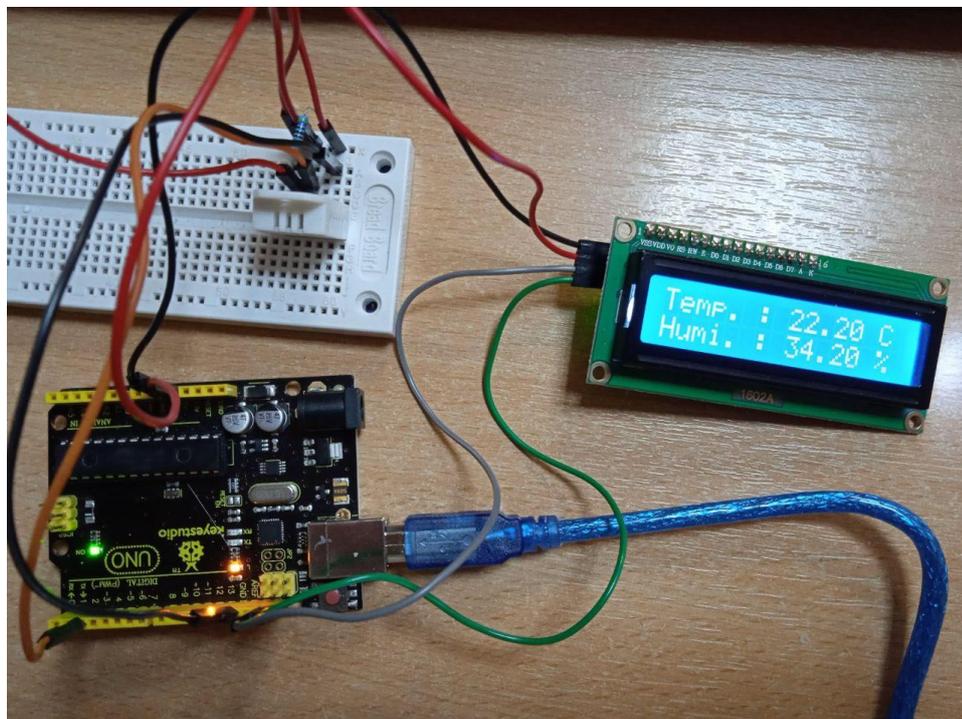


Рисунок 3.7 – Підключення плати керування Arduino до підключення LCD дисплея з використанням плати з мікросхемою PCF 8574T

3.3. Підсистема вимірювання рівня діоксиду вуглецю

Проблематика створення даної підсистеми полягає у складності отримання об'єктивних даних від вимірювальних пристроїв. Аналізуючи існуючі технічні рішення, наведені в розділах 2.2.1 – 2.2.3, варто зазначити, що велике значення у об'єктивності даних залежить від процесу калібрування цих вимірювальних пристроїв. Протягом проведення дослідження було прийнято рішення провести експерименти з різними вимірювальними пристроями.

Першим датчиком, який було використано в підсистемі вимірювання діоксиду вуглецю MQ-135, який описаний в розділі 2.2.1. Слідуючи рекомендаціям, які зазначені в науково-популярній статті [12], що описує принципи роботи даного вимірювального пристрою, було проведено процес калібрування параметра `rzero`, використовуючи запропонований програмний код. Зміни значення параметру `rzero` наведено на рисунку 3.8.



Рисунок 3.8 – Виміряні значення параметру `rzero`

При розробці даної підсистеми у систему було додано тактову кнопку, сигнальний контакт якої підключено до третього цифрового виводу плати керування.

Також було здійснено модифікацію існуючого коду програми системи. Розглянемо зміни, які були внесені.

Було реорганізовано і приведено до одного формату визначення контактів плати керування, які були підключені до зовнішніх пристроїв. Для визначення конфігурації підключення було використано директиву препроцесора `#define`, таким чином було привласнено числові значень, що відповідають номерам цифрових та аналогових виводів, до ключових слів (посилань) таких як `mq135_pin`, `dht_pin` та `button_pin`. Компілятор замінює значення посилань на числові значення протягом компіляції, що дозволяє зменшити кількість пам'яті, що використовується програмою, у порівнянні із визначенням цих значень за допомогою змінних чи констант.

Для функціонування модуля MQ-135 було використано бібліотеку `MQ135.h`, використання якої згадувалось в розділі 2.2.1. Зважаючи на характер зміни параметру (рис. 3.8) було обрано середнє значення 122.42, а також це значення було занесено до файлу бібліотеки `MQ135.h`. У основному коді програми також необхідно було створити об'єкт класу `MQ135`. Ініціалізація об'єкта класу здійснюється за допомогою коду `MQ135 gasSensor = MQ135(mq135_pin)`; аналізуючи який варто зазначити, що метод конструктора потребує значення номера аналогового виводу, у нашому випадку заданого у форматі константи `mq135_pin`. Варто зазначити, що модуль має 4 виводи, один з яких (DOUT) не було використано при створенні даної системи, а аналоговий контакт (AOUT) було підключено до виводу A0 на платі керування.

Для збереження даних величин, що вимірювались в процесі функціонування системи були визначенні змінні `h`, `t`, для вимірювання температури та вологості, а також змінна `ppm`, для вимірювання концентрації діоксиду вуглецю, також відома як мільйонна частка. Всі вищезазначені змінні мають тип даних `float`, що використовується для збереження дробових значень.

Додатковою глобальною змінною, яка була додана до коду основної програми, є змінна `mode` типу даних `int` (цілі числа). Призначення цієї змінної полягає у забезпеченні відображення різного роду даних в залежності від режиму в якому в даний момент працює система. На даному етапі було реалізовано два режими роботи: перший забезпечує відображення даних температури та вологості, другий – відображення концентрації у мільйонних частках (ppm). Логіка зміни режимів наступна. Якщо відбувається натискання на кнопку на сигнальному виводі кнопки з'являється напруга, що приблизно дорівнює значенню 5 вольт. При зчитуванні значення напруги за допомогою команди `digitalRead(button_pin)`; значення напруги, що відповідає 5В буде представлено у вигляді логічної одиниці. І в момент, коли значення `digitalRead(button_pin)` рівне 1 відбувається зміна режиму. Дана логіка реалізована наступною частиною коду:

```
if (digitalRead(button_pin) == 1) {  
    mode++;  
    if (mode > 1) {  
        mode = 0;  
    }  
}
```

Аналізуючи наведений код варто зазначити, що в ньому використовується оператор розгалуження `if` з умовою, що вказана вище. Якщо умова справедлива – виконається тіло конструкції, записане у фігурних дужках, а саме команда `mode++`, яка збільшить значення режиму (`mode`) на 1. Далі слідує ще одна конструкція `if`, яка забезпечує обмеження кількості режимів. Так якщо `mode = 1` і відбувається натискання на кнопку змінна `mode` буде дорівнювати 2, але для цього режиму в програмі немає інструкцій і, відповідно, програма має повернутись до попереднього режиму роботи (`mode = 0`). Для цього і використовується наступна перевірка `if (mode > 1)`. У випадку коли `mode` буде дорівнювати 2 дана умова буде справедлива і виконається тепер вже тіло другої конструкції `if`, яке присвоїть змінній `mode` значення 0. Таким чином забезпечується перемиканням між двома

режимами роботи (дана конструкція також може бути адаптована до більшої кількості режимів), логіка роботи кожного описана далі.

Для відображення різних даних на дисплеї була використана конструкція розгалуження switch-case:

```
switch (mode) {  
    case 0:  
        lcd.setCursor(0, 0);  
        lcd.print("Temp. : ");  
        lcd.print(t);  
        lcd.print(" C ");  
        lcd.setCursor(0, 1);  
        lcd.print("Humi. : ");  
        lcd.print(h);  
        lcd.print(" % ");  
        break;  
    case 1:  
        lcd.setCursor(0, 0);  
        lcd.print("MQ-135");  
        lcd.setCursor(0, 1);  
        lcd.print("CO2: ");  
        lcd.print(ppm);  
        lcd.print(" ppm");  
        break;  
}
```

Наведений код дозволяє порівняти значення змінної mode із 0 або 1, і, в залежності від значення обрати один із двох алгоритмів. У випадку коли змінна mode дорівнює 0 виконуватиметься перший алгоритм відображення даних, описаний у попередньому розділі, інакше на дисплеї будуть відображатись дані про тип датчика, який на даний момент використовується, концентрацію діоксиду вуглецю і одиниці вимірювання.

Результати реалізації даної підсистеми з датчиком MQ-135 наведено на рисунку 3.9.

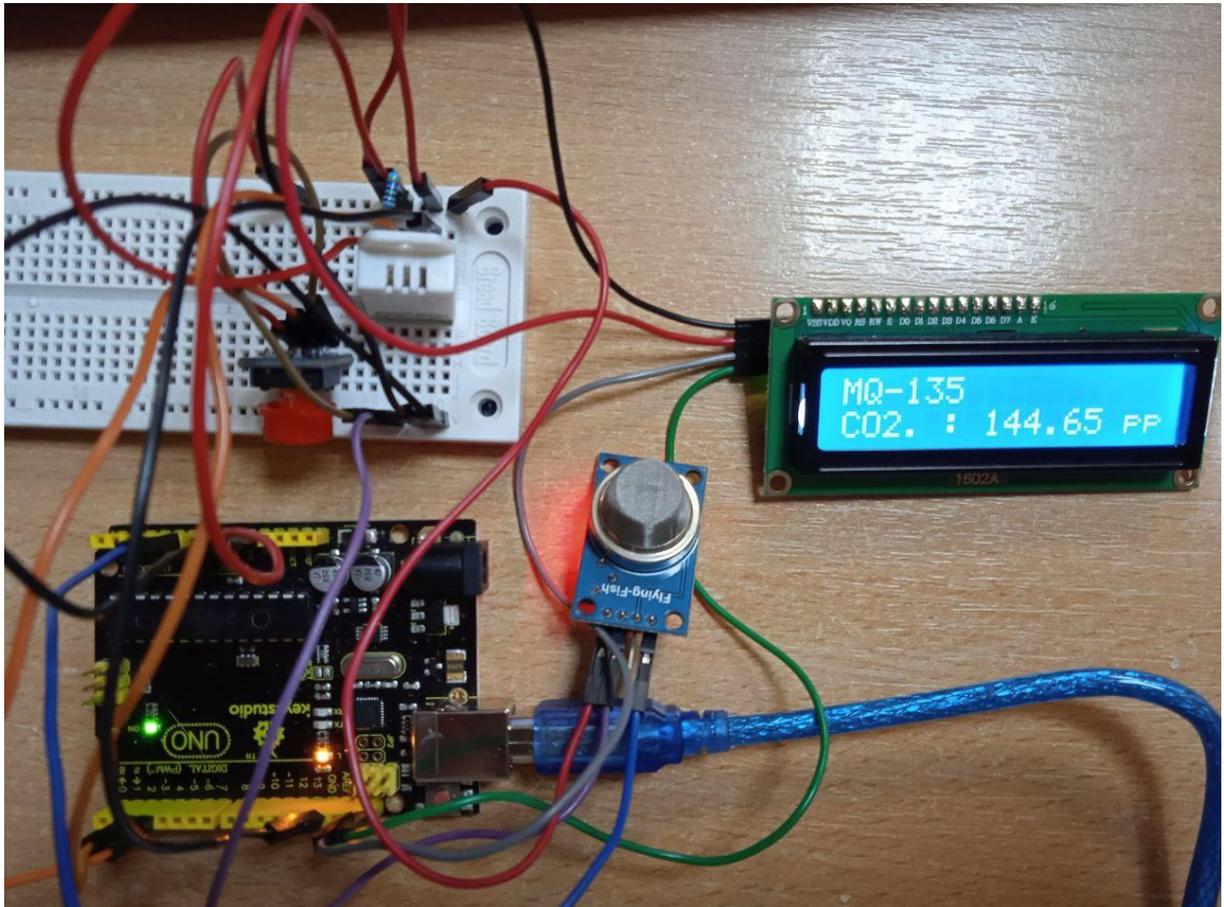


Рисунок 3.9 – Прототип системи моніторингу температури, вологості та рівня діоксиду вуглецю з використанням модулів DHT22 та MQ-135

3.4. Підсистема керування виконавчими пристроями

Основним виконавчим пристроєм для зміни рівня діоксиду вуглецю в приміщенні є електродвигун, як частина системи вентиляції. У даному дослідженні було розглянуто можливість застосування двигунів постійного струму та безколекторних двигунів, а також кілька схем керування даними типами електродвигунів. Варто зазначити, що немає єдиного правильного рішення для вибору двигуна та схеми керування, адже вибір залежить від типу приміщення, його об'єму, економічних ресурсів, а також характеру зміни рівню діоксиду

вуглецю в приміщенні. Тому в даному дослідженні пропонується кілька схем, які, на думку автора, можуть бути використані в системі вентиляції.

Розглянемо використання двигунів постійного струму в якості виконавчого органу підсистеми вентиляції, загальні відомості про керування ДПС наведені в статті [21], де пропонується кілька варіантів керування, одним з яких є використання транзисторної схеми керування, а інший – драйверу двигуна L298N.

Відповідно до [21] ми можемо керувати швидкістю двигуна постійного струму, просто керуючи входною напругою двигуна, і найпоширенішим способом цього є використання сигналу широтно-імпульсної модуляції (ШІМ). ШІМ, або широтно-імпульсна модуляція – це техніка, яка дозволяє нам регулювати середнє значення напруги, що надходить на електронний пристрій, швидко вмикаючи та вимикаючи живлення. Середня напруга залежить від робочого циклу або часу, протягом якого сигнал був увімкненим, порівняно з часом, протягом якого сигнал був вимкненим за один період часу. Таким чином, залежно від розміру двигуна, ми можемо просто підключити вихід Arduino ШІМ до бази транзистора або затвора польового транзистора (MOSFET) і контролювати швидкість двигуна, керуючи виходом ШІМ. ШІМ-сигнал низької потужності Arduino вмикає та вимикає затвор на польового транзистора, через який керується потужний двигун. Схема використання польового транзистора для керування швидкістю ДПС наведена на рисунку 3.9.

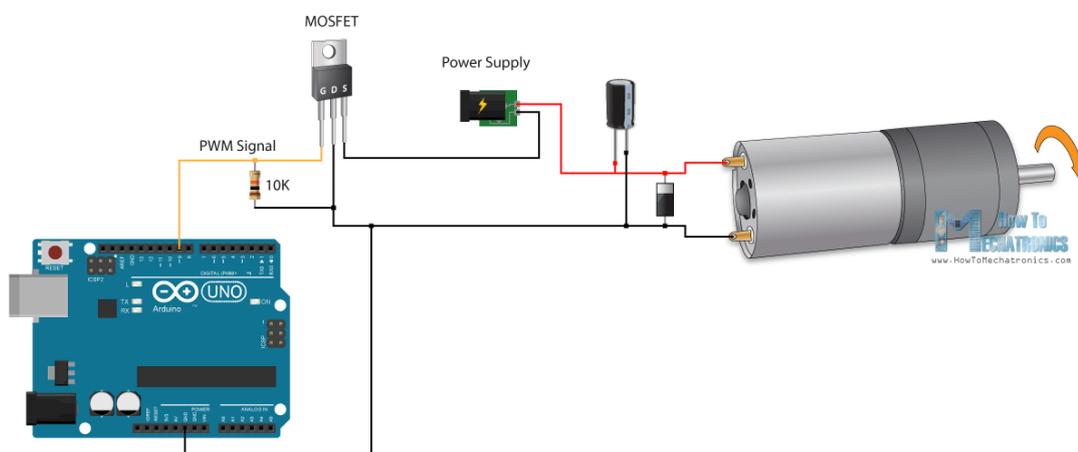


Рисунок 3.9 – Схема керування швидкістю ДПС із використанням польового транзистора

З іншого боку, щоб керувати напрямком обертання, нам просто потрібно змінити напрямок струму, що протікає через двигун, і найпоширенішим способом зробити це є використання Н-подібного мосту. Схема Н-подібного мосту, що наведена на рисунку 3.10, містить чотири перемикаючі елементи, транзистори або MOSFET, з двигуном у центрі, який утворює Н-подібну конфігурацію. Активуємо два певних перемикача одночасно, ми можемо змінити напрямок потоку струму, таким чином змінюючи напрямок обертання двигуна. Отже, якщо ми об'єднаємо ці два методи, ШІМ і Н-міст, ми зможемо мати повний контроль над двигуном постійного струму. Існує багато драйверів двигунів постійного струму, які мають ці функції, і L298N є одним із них.

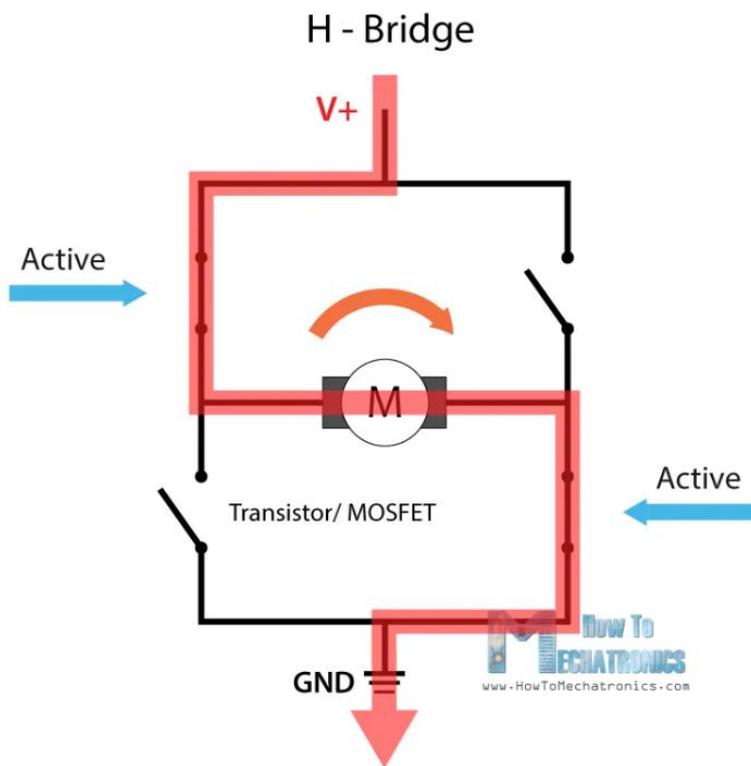


Рисунок 3.10 – Схема Н-подібного мосту

Драйвер L298N - це драйвер двигуна з подвійним Н-подібним мостом, який дозволяє одночасно керувати швидкістю та напрямком двох двигунів постійного струму. Модуль може керувати двигунами постійного струму з напругою від 5 до 35 В з піковим струмом до 2 А зображений на рисунку 3.11.

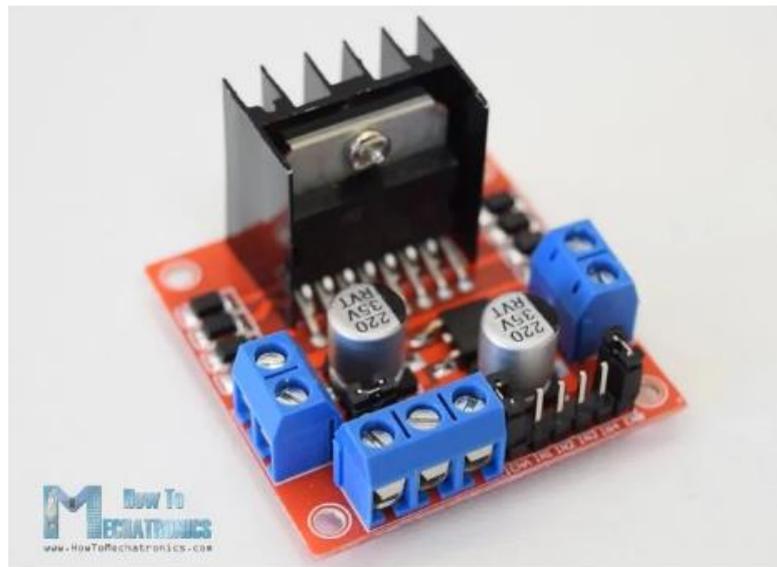


Рисунок 3.11 – Драйвер двигуна L298N

Модуль має дві гвинтові клемні колодки для двигунів А і В, а також іншу гвинтову клемну колодку для контакту заземлення, VCC для двигуна та контакту 5 В, який може бути входом або виходом. Для забезпечення більш потужної вентиляції можна використовувати кілька двигунів постійного струму, приклад такого підключення показано на рисунку 3.12.

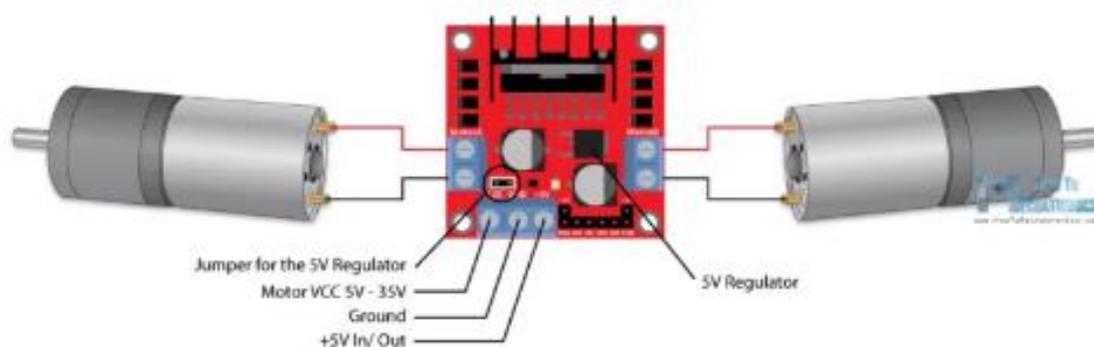


Рисунок 3.12 – Приклад підключення двох двигунів до драйвера

Драйвер двигуна L298 також можна підключити до Arduino, щоб випробувати його нам потрібно такі складові:

- драйвер L298N;
- двигун постійного струму з високим крутним моментом 12В;

- плата керування Arduino;
- макетна плата та перехідні дроти.

Розглянемо яким чином можна модифікувати існуючий код для того щоб забезпечити роботу двигуна при перевищенні допустимого значення концентрації діоксиду вуглецю в повітрі.

Перш за все необхідно додати конфігурацію підключення. Для цього був використаний аналогічний метод, як і у попередньому розділі, використовуючи директиву препроцесора `#define`. Таким чином додаємо три константи `in1`, `in2` та `enA`, що дорівнюють номерам цифрових виводів мікроконтролера до яких підключені ці контакти модуля L298. Важливо зазначити, що вивід `enA` має бути підключеним до цифрового виводу з маркуванням «~» або ж PWM (ШИМ), він буде задавати ширину імпульсів керування `i`, відповідно, швидкість обертання двигуна.

У функції `setup` необхідно задати режими роботи всіх трьох виводів `in1`, `in2` та `enA` за допомогою команди `pinMode()`, та зазначити першим аргументом номер цифрового виводу (в нашому разі посилання: `in1`, `in2`, `enA`), а другим аргументом вказати режим `OUTPUT`. Також бажано задати низькі рівні сигналів на всіх трьох виводах за допомогою наступного коду:

```
digitalWrite(in1, LOW);  
digitalWrite(in2, LOW);  
analogWrite(enA, 0);
```

Наступним кроком є встановлення умови включення двигуна(`iv`) для активації системи вентиляції, а також вимкнення двигунів у разі, коли рівень діоксиду вуглецю знаходиться на допустимому рівні. Реалізація даного алгоритму може і має відбуватись не залежно від режиму роботи, який відповідає за відображення даних, тому цей функціонал буде записаний за межами конструкції `switch-case`, що описана у попередньому розділі.

У даному дослідженні було запропоновано використувувати закону керування за кривою гістерезису. Також цей метод вимагає введення кількох додаткових змінних (`turn_on_value` та `turn_off_value`), а його реалізація за допомогою програмного коду матиме наступний вигляд:

```

if (ppm > turn_on_value) {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    analogWrite(enA, 255);
}
if (ppm < turn_off_value) {
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    analogWrite(enA, 0);
}

```

У даному дослідженні було обрано використання змінних `turn_on_value` та `turn_off_value` з цілочисельним типом даних (`int`), значення яких задаються при їхній ініціалізації і встановлюються відповідно до вимог, описаних в розділі 1.

Логіка, що забезпечує висхідну частину характеристики кривої гістерезису, описана у першій конструкції розгалуження. Якщо виміряне значення діоксиду вуглецю більше за допустиме, що записане у змінній `turn_on_value` на одному з цифрових виводів (`in1` або `in3`) встановлюється високий рівень сигналу, а на іншому – низький, що у свою чергу забезпечує можливість протікання струму через електродвигун у одному напрямку. Рівень напруги на двигуні, що пропорційний до швидкості обертання електродвигуна, встановлюється шляхом подачі ШІМ-сигналу на цифровий вивід `enA`. Цей функціонал забезпечується за допомогою команди `analogWrite()`, яка, в свою чергу, приймає два параметри. Першим параметром вказується номер цифрового виводу (`enA`), а другим параметром середній рівень напруги, який задається у відносній величині в межах від 0 до 255 (де 0 – 0В, а 255 – 5В). У випадку, якщо необхідно змінити напрям обертання двигуна в командах `digitalWrite()` для виводів `in1` та `in2` змінюються рівні напруги з HIGH на LOW і навпаки.

Відповідно до заданої логіки увімкнений стан роботи двигуна буде залишатись до моменту поки значення `ppm` не зменшиться нижче рівня, що задане в змінній `turn_off_value`. У цей момент значення напруги на виводах `in1`, `in2` та `enA`

будуть встановлені на рівні 0В. Таким чином друга конструкція if забезпечує низхідну частину кривої гістерезису.

Варто зазначити, що окрім закону керування може використовуватись і лінійний закон задання швидкості обертання електродвигуна від зміни концентрації діоксиду вуглецю, або ж комбінація цих двох законів, але імплементація даної моделі потребує подальших досліджень.

Також у межах даного дослідження було розглянуто можливість використання безщіткових (BLDC) або ж вентильних двигунів [22]. Для прикладу розглянемо двигун із рейтингом KV = 1000, що означає, що при подачі на двигун напруги у розмірі 7.4 вольта, двигун може досягти максимальних обертів на хвилину значення $7.4 * 1000$, що дорівнює 7400 обертів на хвилину. Подібний двигун зображений на рисунку 3.13.



Рисунок 3.13 – Безщітковий двигун

Безщіткові двигуни потребують зовнішнє джерело електроенергії, і найпоширенішим способом їх живлення є використання літій-полімерних (LiPo) акумуляторів. Число «S» батареї вказує на кількість елементів батареї, кожна з яких має напругу 3.7 В. Для прикладу візьмемо акумулятор 3S LiPo, який має 3 елемента, що забезпечує напругу живлення 11.1 В.

Для керування подібними двигунами використовують драйвери або ж регулятори швидкості обертання ESC. З одного боку ESC має три дроти, які керують трьома фазами двигуна, а з іншого боку він має два дроти, VCC і GND, для живлення зображених на рисунку 3.14.

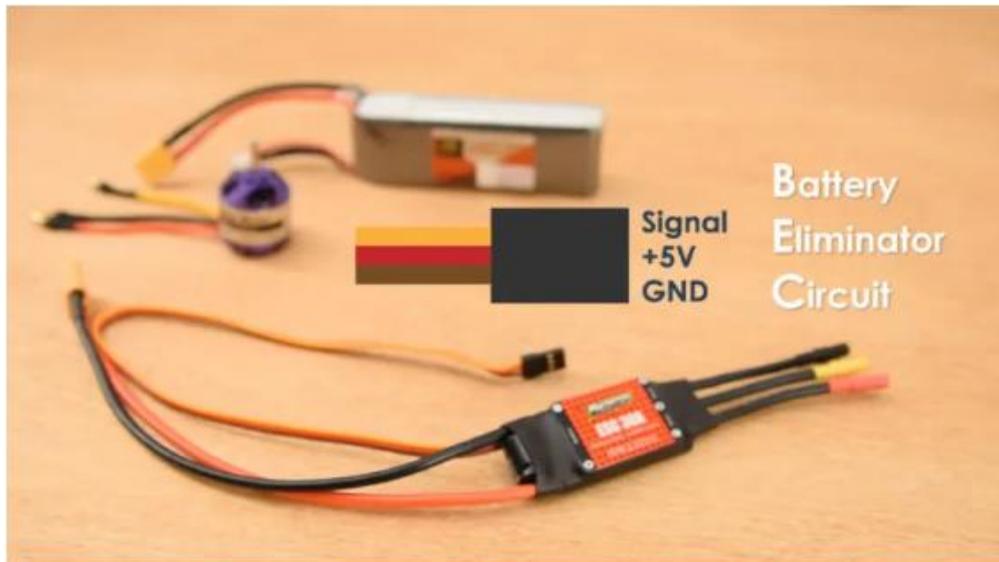


Рисунок 3.14 – Драйвер ESC

Для імплементації цього типу електродвигуна до підсистеми керування виконавчими пристроями забезпечувати генерацію сигнал ШІМ частотою 50 Гц, і залежно від ширини імпульсу або тривалості високого стану, яка повинна варіюватися від 1 мілісекунди до 2 мілісекунд, ESC буде керувати двигуном від мінімальних до максимальних обертів. Для того, щоб протестувати безщітковий двигун, потрібно зібрати схему, подібну до наведеної на рисунку 3.15, з набором пристроїв:

- Безщітковий двигун
- ESC 30A
- Li-Po батарея
- Плата керування Arduino
- Макетна плата та перехідні дроти
- Підсистема вимірювання діоксиду вуглецю
- Підсистема індикації (опціонально)

- Підсистема вимірювання температури та вологості (опціонально)

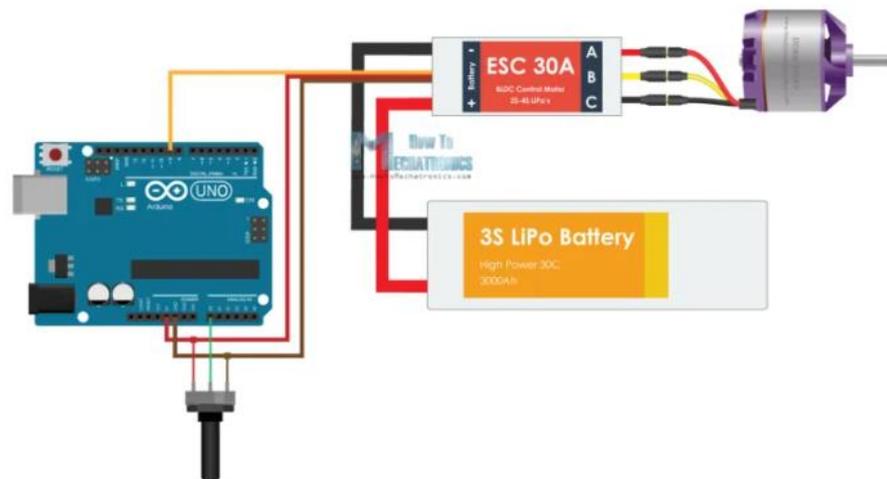


Рисунок 3.15 – Приклад підключення безщіткового двигуна до плати керування Arduino

Варто зазначити, що схема, наведена на рисунку 3.15 не відповідає конфігурації запропонованої системи і використовується для ілюстрації підключення BLDC-двигуна, а також взята з науково-популярної статті [22], проте дана ілюстрація дозволяє зрозуміти конфігурацію підключення.

Для керування BLDC-двигуном із драйвером ESC зручно використовувати ту ж бібліотеку, яка використовується для керування серводвигуном, а саме `Servo.h`. Необхідно створити об'єкт класу `Servo`, поруч із створенням інших об'єктів. Назва об'єкту має відповідати його призначенню, тому назвімо його `ESC`, відповідно код, який забезпечує дану дію:

```
Servo ESC;
```

Також сигнальний контакт ESC необхідно приєднати до ШІМ-виводу на платі керування, а також вказати на нього посилання за допомогою рядку коду `#define esc_pin 6`.

У функції `setup` також необхідно викликати метод `attach`, що потребує обов'язково одного параметру – номера цифрового виводу на платі керування, до якого підключений сигнальний контакт драйверу, та два опціональних параметри,

а саме – мінімальна та максимальна тривалість ШІМ-імпульсу в мікросекундах.

Таким чином виклик методу `attach` матиме вигляд:

```
ESC.attach(9,1000,2000)
```

Модифікуємо частину коду з попереднього прикладу для того щоб тепер мікроконтролер забезпечував керування по закону гістерезисної кривої з використанням BLDC-двигуна та ESC драйвера:

```
if (ppm > turn_on_value) {  
    ESC.write(180);  
}  
if (ppm < turn_off_value) {  
    ESC.write(0);  
}
```

У даному випадку значення 0 і 180 відповідають мінімальній та максимальній тривалості ШІМ-сигналу і будуть встановлювати мінімальну (0) та максимальну швидкість обертання двигуна.

Однак в статті [22] є кілька речей, варто відзначити. Під час початку подачі живлення двигуна значення сигналу має бути таким самим або меншим за мінімальне значення в 1 мілісекунду. Це називається встановленням ESC драйвера на зведення, і двигун видає звуковий сигнал підтвердження, щоб ми знали, що він правильно поставлений на зведення. У випадку, якщо ми маємо більш високе значення під час живлення, що означає, що ми маємо дросельну заслінку, ESC не запустить двигун, доки ми не знизимо дросель до правильного мінімального значення. Це дуже зручно з точки зору безпеки, тому що двигун не заведеться, якщо ми підняли живлення при подачі.

Також потрібно відкалібрувати ESC. Кожен ESC має свої власні верхні та нижні точки, і вони можуть дещо відрізнитися. Наприклад, найнижча точка може становити 1,2 мілісекунди, а найвища — 1,9 мілісекунди. У такому випадку наша дросельна заслінка нічого не робитиме в перші 20%, доки не досягне найнижчого значення 1,2 мілісекунди.

3.5. Висновок до розділу 3

Нами було вибрано використання саморобної системи аналізу та керування параметрами мікроклімату, в її основі була закладена система яка була представлена на VIII Всеукраїнській науково-практичній конференції «Електронні та мехатронні системи: теорія, інновації, практика. Для вимірювання температури та вологості ми обрали модуль DHT 22, вибір був обґрунтований тим, що похибка при вимірюваннях температури в 4 рази менша ніж у DHT11, так і похибка при вимірюванні вологості менша в 2рази. Для підключення модуля використали підтягуючий резистор з опором 4,7 кОм. При розробці програмного забезпечення ми вибрали код з статті [17]. Для візуалізації даних ми вибрали використання дисплея LCD 1602, так як хоча і світлодіодні та звукові системи є дешевшими, але зчитувати данні з дисплея буде інформативніше. Щоб зменшити кількість використаних дисплеєм виводів на платі Arduino, було вирішено використати плату розширення PCF 8574T. Програмний код зазначений в розділі 2.2.4 був використаний та модифікований для додавання підсистем індикації. Для того, щоб мати змогу вимірювати рівень діоксиду вуглецю було додано модуль MQ-135, та додано тактову кнопку задля переключення параметрів на дисплеї з температури та вологості на вміст діоксиду вуглецю в повітрі. Для того щоб була можливість зміни параметрів вуглецю в приміщенні є електродвигун як частина системи вентиляції. Для цього можна вибрати двигун постійного струму, або ж безколекторний, але немає єдиного правильного рішення для вибору двигуна та схеми керування ним, так як вибір залежить від розміру приміщення в якому система буде використовуватись, характеру зміни діоксиду вуглецю, або ж економічним ресурсам. Для нашого проєкту було запропоновано двигун постійного струму, з драйвером двигуна L298N, з цим драйвером ми можемо керувати двигуном через ШІМ. Також щоб керувати напрямком обертання нам потрібно використати H-подібний міст. Як варіант для більш потужної вентиляції можна приєднати до драйвера двигуна L298N кілька двигунів.

Загальні висновки

У ході дослідження було виконано ряд задач, результати яких виокремимо у вигляді висновків:

1. Було проведено аналіз стандартів параметрів мікроклімату України та країн ЄС та впливу зміни параметрів на людину (Розділ 1).

2. Визначено оптимальні параметри мікроклімату для виробничих приміщень, а також значення параметрів, які можуть вказувати на небезпечні умови для здоров'я людини (Розділ 1).

3. Досліджено системи та технічні засоби, що використовуються в системах моніторингу та керування параметрів мікроклімату (Розділ 2).

4. Визначено пристрої для реалізації системи моніторингу та керування параметрами мікроклімату та здійснено дослідження принципів роботи пристроїв керування, вимірювання та виконавчих пристроїв (Розділ 2).

5. Розроблено підсистему вимірювання температури та вологості середовища, підсистема індикації, підсистема вимірювання рівня діоксиду вуглецю, підсистема керування виконавчими пристроями (Розділ 3).

6. Розроблено та реалізовано у вигляді програмного коду алгоритми системи моніторингу та керування параметрами мікроклімату, що забезпечують роботу розробленої системи (Розділ 3).

Література

1. Мікроклімат [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/%D0%9C%D1%96%D0%BA%D1%80%D0%BE%D0%BA%D0%BB%D1%96%D0%BC%D0%B0%D1%82>
2. Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99 [Електронний ресурс] / 1. – 112. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99#Text>.
3. Мікроклімат виробничих приміщень, вплив параметрів мікроклімату на організм людини [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: <https://oppb.com.ua/news/mikroklimat-vyrobnychyh-prymishchen-vplyv-parametriv-mikroklimatu-na-organizm-lyudyny>.
4. Як мікроклімат офісу впливає на працездатність працівників? [Електронний ресурс] / 1. – 2021. – Режим доступу до ресурсу: <https://news.dtkr.ua/labor/social-protection/71225-yak-mikroklimat-ofisu-vplivaje-na-pracездatnist-pracivnikov>.
5. CO₂ у приміщеннях. На що впливає діоксид вуглецю у повітрі та як зменшити його рівень [Електронний ресурс] – Режим доступу до ресурсу: <https://aerostar.ua/ua/news/novosti/co2-u-primischennjah-na-scho-vplivae-dioksid-vuglecju-u-povitri-ta-jak-zmenshiti-jogo-riven.html>.
6. Основні вимоги та заходи з нормалізації мікрокліматичних умов на робочих місцях [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://pro-op.com.ua/article/82-osnovn-vimogi-ta-zahodi-z-normalzats-mkroklimatichnih-umov-na-robochih-mstsyah#:~:text=Нормативне%20регулювання,залишаються%20чинними%20у%202021%20році..>
7. CO₂ Monitor [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://create.arduino.cc/projecthub/ArduinoKoen/co2-monitor-fef97b>.
8. Portable Arduino Temp/Humidity Sensor with LCD [Електронний ресурс]. – 2016. – Режим доступу до ресурсу:

https://create.arduino.cc/projecthub/ThothLoki/portable-arduino-temp-humidity-sensor-with-lcd-a750f4?ref=tag&ref_id=temperature&offset=0.

9. DIY Air Quality Monitor – PM2.5, CO2, VOC, Ozone, Temp & Hum Arduino Meter [Електронний ресурс] – Режим доступу до ресурсу: <https://howtomechatronics.com/projects/diy-air-quality-monitor-pm2-5-co2-voc-ozone-temp-hum-arduino-meter/>.

10. Термогігрометр з датчиком вимірювання CO2 SBTR РТН-9с [Електронний ресурс] / 1 – Режим доступу до ресурсу: https://sbtr.com.ua/ua/p1536046093-termogigrometr-datchikom-izmereniya.html?gclid=EAIAIQobChMIpvau1MSR-wIVGrayCh3-BwMHEAQYBCABEgKZtfD_BwE.

11. Вимірювач рівня CO2 TFA «AIRCO2NTROL 5000», логер-реєстратор [Електронний ресурс] – Режим доступу до ресурсу: <https://comfortshop.com.ua/product/izmeritel-urovnya-co2-tfa-airco2ntrol-5000-logger-registrator-120h33h66-mm-31500802/>.

12. [Arduino basics tutorials] use MQ135 air quality detecting module [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://steemit.com/utopian-io/@cha0s0000/arduino-basics-tutorials-use-mq135-air-quality-detecting-module>.

13. Модуль датчика якості повітря MQ-135 [Електронний ресурс] – Режим доступу до ресурсу: <https://arduino.ua/prod1201-modyl-datchika-kachestva-vozdyha-mq135>.

14. МН-Z19В NDIR CO2 Sensor with Arduino, MHZ19В [Електронний ресурс] – Режим доступу до ресурсу: <https://www.electronicclinic.com/mh-z19b-ndir-co2-sensor-with-arduino-mhz19b/>.

15. Датчик CO2 МН-Z19В [Електронний ресурс] – Режим доступу до ресурсу: <https://arduino.ua/prod1532-datchik-co2-mh-z19>.

16. SKU:SEN0159 [Електронний ресурс] – Режим доступу до ресурсу: https://wiki.dfrobot.com/CO2_Sensor_SKU_SEN0159.

17. DHT11 & DHT22 Sensors Temperature and Humidity Tutorial using Arduino [Електронний ресурс] – Режим доступу до ресурсу:

<https://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/>.

18. Боряк Б. Р. Модель системи керування рівнем вуглекислого газу в приміщеннях / Б. Р. Боряк, А. О. Косинков. // Збірник наукових праць за матеріалами VIII Всеукраїнської науково-практичної конференції «Електронні та мехатронні системи: теорія, інновації, практика». – 2022. – С. 13–15.

19. Temperature Monitoring With DHT22 & Arduino [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: <https://create.arduino.cc/projecthub/mafzal/temperature-monitoring-with-dht22-arduino-15b013>.

20. Interface an I2C LCD with Arduino [Електронний ресурс] – Режим доступу до ресурсу: <https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/>.

21. L298N Motor Driver – Arduino Interface, How It Works, Codes, Schematics [Електронний ресурс] / 1. – 2020. – Режим доступу до ресурсу: <https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>.

22. Arduino Brushless Motor Control Tutorial | ESC | BLDC [Електронний ресурс] – Режим доступу до ресурсу: <https://howtomechatronics.com/tutorials/arduino/arduino-brushless-motor-control-tutorial-esc-blcd/>.

Додатки

Додаток А. Section 3. development of the configuration of the indoor microclimate monitoring and management system

At the first stages of system development, we implemented the first version of the system configuration for conducting research on changes in microclimate parameters, which was presented at the VIII All-Ukrainian scientific and practical conference "Electronic and Mechatronic Systems: Theory, Innovation, Practice" [18]. Further research revealed a number of disadvantages of this configuration and it was decided to modify it. It was decided to divide the process of building a system for monitoring and controlling microclimate parameters into the development of several subsystems, namely, the Subsystem for measuring ambient temperature and humidity, the display subsystem, the Subsystem for measuring the level of carbon dioxide, and the control subsystem for actuators. This distribution will allow you to implement a discrete system, and each subsystem can be modified or replaced.

3.1. Subsystem for monitoring ambient temperature and humidity

In the process of analyzing the existing technical means, we identified several measuring devices for measuring temperature and humidity, presented in Section 2.2.4. these devices were selected taking into account the criteria of availability of measuring devices on the Ukrainian market, accuracy and economic feasibility. The measuring devices that were selected for implementation are DHT22 and DHT11. The advantages of DHT11 are a low price (30 – 50 UAH), and a lower frequency of data acquisition compared to the DHT22 module (1 read within 1 second). The price of DHT22 exceeds the price of the DHT11 module by about 3 times, and the frequency of data acquisition is half as low (1 read Within 2 seconds), according to the technical documentation. But the temperature measurement error of the DHT22 measuring device is approximately 4 times less than the measurement error of DHT11, and the measurement error of DHT22

humidity is approximately 2 times less than that of DHT11. taking into account these facts, we chose to use the DHT22 module. A detailed comparative analysis of the technical characteristics of DHT11 and DHT22, as well as the principle of operation of these devices, is described in Section 2.2.4. The connection of the temperature measurement subsystem is shown in figure 3.1.

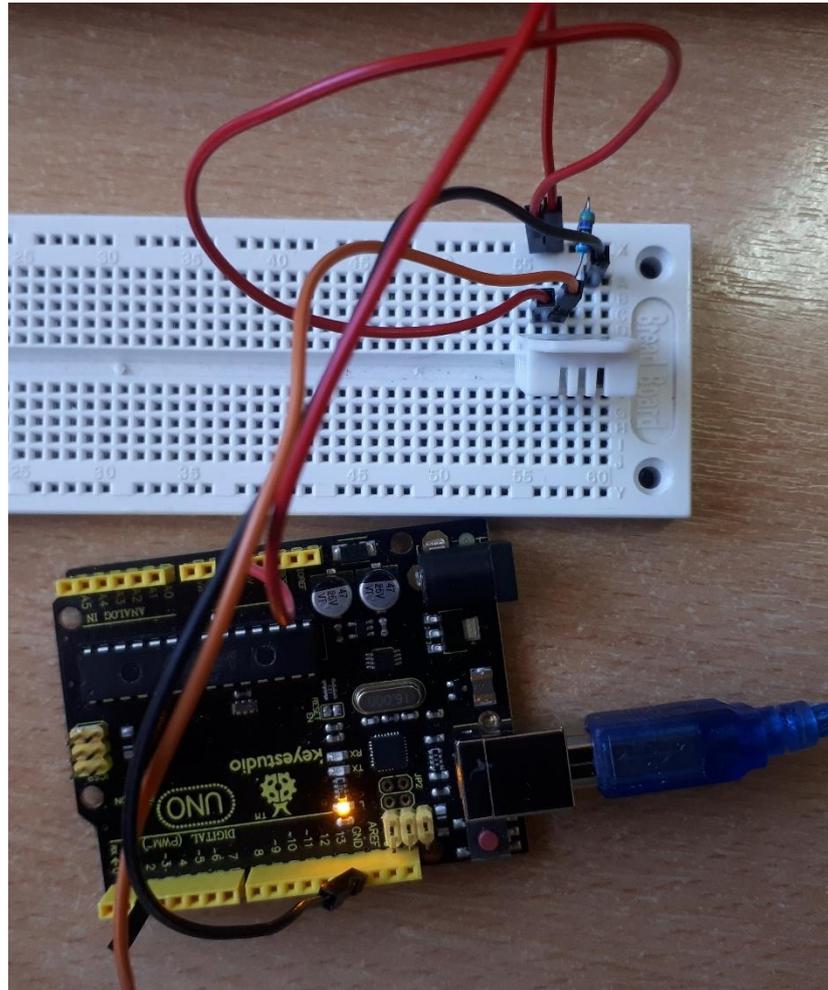


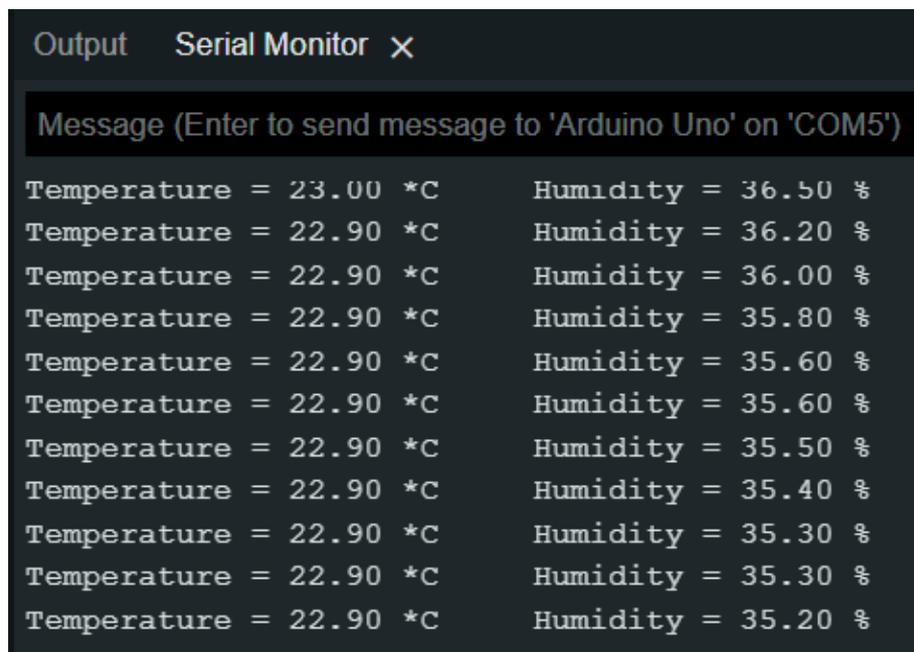
Figure 3.1-illustration of the connection configuration of elements of the ambient temperature and humidity measurement subsystem

According to [17], a pull-up resistor with a resistance of 4.7 Kohm was used when connecting the sensor. The general connection of the sensor to the Control Board is as follows (the pins are numbered from left to right when viewed from the front of the module):

- 1 pin is connected to the power supply, on the Control Board-5V;
- 2 pin is connected to a 4.7 Kohm pull-up resistor connected to the 5V power output, and also connected to the 2nd digital output on the Control Board (Orange insulated conductor in figure 3.1);
- 3 pin is not connected, not used;
- Pin 4 is connected to the negative pin of the power supply, on the Control Board – GND.

During software development, several libraries were tested to ensure the correct operation of the measuring device. We checked the programs given in [17, 19], both libraries allow us to measure temperature and humidity parameters, so we used the program code given in the Popular Science article [17].

The result of the subsystem is Data Display using the serial port monitor, which is shown in Figure 3.2.



The screenshot shows a 'Serial Monitor' window with a title bar 'Output Serial Monitor X'. Below the title bar is a message: 'Message (Enter to send message to 'Arduino Uno' on 'COM5')'. The main area of the window displays a series of data points for temperature and humidity. The temperature values are 23.00 *C, followed by ten instances of 22.90 *C. The humidity values are 36.50 %, followed by ten instances of 36.20 %, 36.00 %, 35.80 %, 35.60 %, 35.60 %, 35.50 %, 35.40 %, 35.30 %, 35.30 %, and 35.20 %.

Temperature (*C)	Humidity (%)
23.00	36.50
22.90	36.20
22.90	36.00
22.90	35.80
22.90	35.60
22.90	35.60
22.90	35.50
22.90	35.40
22.90	35.30
22.90	35.30
22.90	35.20

Figure 3.2-transmission of measured data using a measuring device via the serial port protocol

3.2. Subsystem for displaying environmental parameters

At the first stages of design, it is necessary to analyze the resources that may be needed in the implementation of various subsystems. Since one of the subsystems that can use significant resources in terms of connecting and using the digital pins of the Control Board is the display subsystem. Let's consider the implementation options for this subsystem. The first step is to select the format and technical means of the display subsystem. Different monitoring systems may use different technical tools, including:

- - LED display systems: used in systems where there is no need to display the values of the measured parameters, or display the values of parameters by conditional division into levels or states that depend on the values of the measured value.

- - Sound indication systems: used in systems where it is necessary to report a state change by means of an audible signal, which may indicate the presence of emergency operating modes or modes to which the system operator must respond.

- Data visualization systems: such systems allow you to display data in numerical form, time characteristics, or even display the technological process as an interactive model. The technical tools used in such visualization systems are usually significantly more expensive than in LED and sound display systems. Analyzing the available formats of display subsystems from the perspective of implementing a system for monitoring and controlling microclimate parameters, we chose one of the cheapest liquid crystal displays, namely: LCD 1602 16x2 character display. The advantage of this device is that it allows you to display information in the format of symbolic messages on 2 lines, each of which can display up to 16 characters, as well as the price, which ranges from 100 – 120 UAH.

This display is shown in Figure 3.3.



Figure 3.3 – LCD 1602 character display

The connection diagram for this display is shown in Figure 3.4.

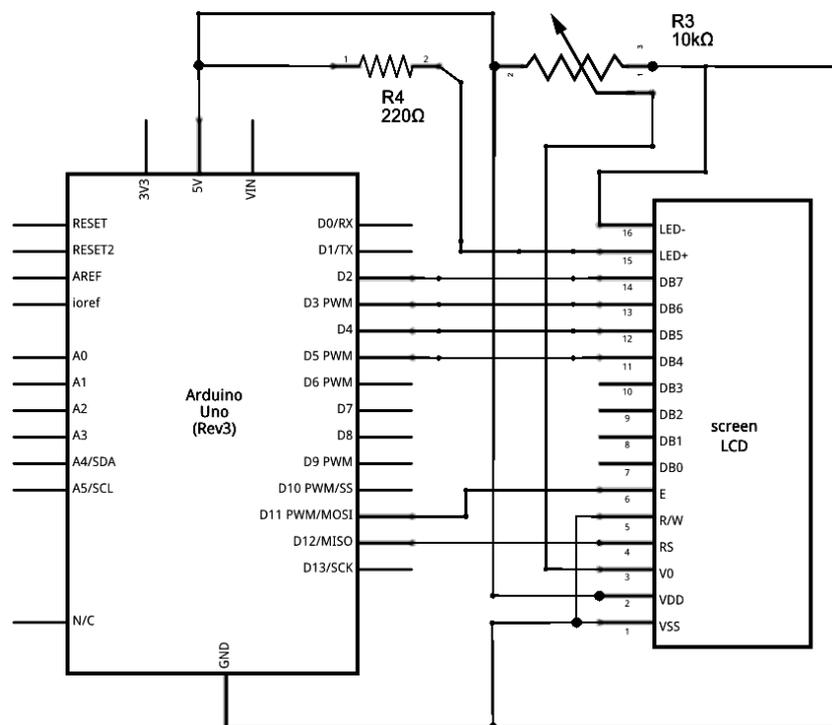


Figure 3.4-connection diagram

The problem with implementing this connection is that it requires the use of 6 digital pins of the Control Board and this is unsatisfactory conditions for further implementation

of other subsystems, although it is quite economical, given that in addition to the 10kohm potentiometer R3 and the current-limiting resistor, this configuration does not require other elements. Given the need to use temperature, humidity, and carbon dioxide measurement devices, it was decided to minimize the number of digital outputs for data visualization. To do this, it is possible to connect the display to the control board using the I2C Arduino pin expansion module for connecting the LCD display to the PCF 8574T (the price is 50-60 UAH), which is shown in Figure 3.5

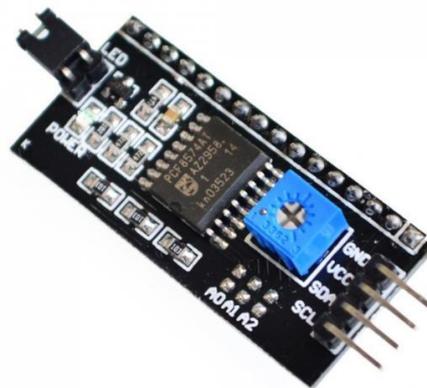


Figure 3.5-I2C Arduino pin expansion module for connecting an LCD display to a PCF 8574T

This expansion module allows you to use only two SDA and SCL pins (except for the VCC and GND power pins) to transfer data from the microcontroller to the control board. A detailed description of the connection and operation of the display in combination with an expansion card is given in the Popular Science article [20].

The general connection of an LCD display using an expansion card with a PCF 8574T chip is shown in Figure 3.6.

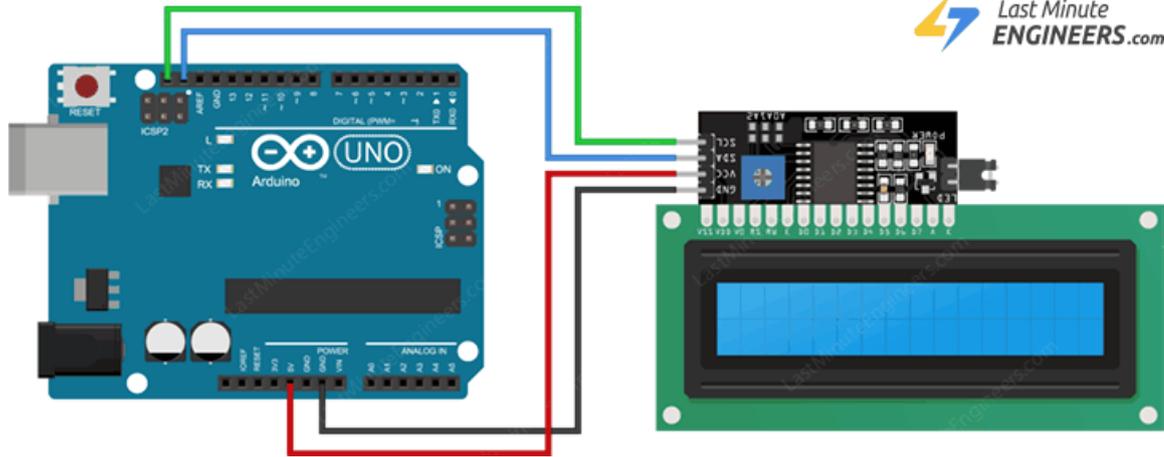


Figure 3.6 - connecting an Arduino control board to an LCD display using a board with a PCF 8574t chip

The program code written in Section 2.2.4, which was taken as a basis and which was used to implement the temperature and humidity measurement subsystem, was modified with the addition of the display subsystem.

Wire libraries were used to interact with the display in the program code.h and LiquidCrystal_I2C.h. To control the display in the program code, you must declare a global class object by adding the following code `LiquidCrystal_I2C lcd (0x27,16,2);` where `LiquidCrystal_I2C` is the class name, `lcd` is the object name, `0x27` is the I2C address of the device, `16` is the number of characters that can be written in one line, `2` is the number of lines of the display.

You must also add calls to several methods of the `lcd` object to the setup function, namely `lcd.init();` `lcd.clear();` `lcd.backlight ();` each of which performs the following functions, respectively: initializing the `lcd`, cleaning the display, turning on the display backlight.

```
The following program code has been added to the loop function:
lcd.setCursor(0,0);
lcd.print("Temp. : ");
lcd.print(t);
lcd.print(" C ");
lcd.setCursor(0,1);
```

```
lcd.print("Humi. : ");  
lcd.print(h);  
lcd.print(" % ");
```

This code is used to output data received from the DHT22 module and written to the variables *t* (temperature) and *H* (humidity). The `setCursor` method allows you to set the location on the display from which data will be output. The first parameter, which is equal to 0, defines the column number (there are 16 of them in total, but numbering starts with 0), and the second parameter, which is also equal to 0, defines the row number (numbering also starts with 0). This way, data will be displayed from the upper-left corner of the display. The `print` method directly performs the function of data output. In addition, after a delay of 2 seconds, which is necessary for the correct operation of DHT22, a call to the `clear ()` method was added to clear the display of previous values.

The result of implementing the display subsystem together with the operation of the ambient temperature and humidity measurement subsystem is shown in Figure 3.7.

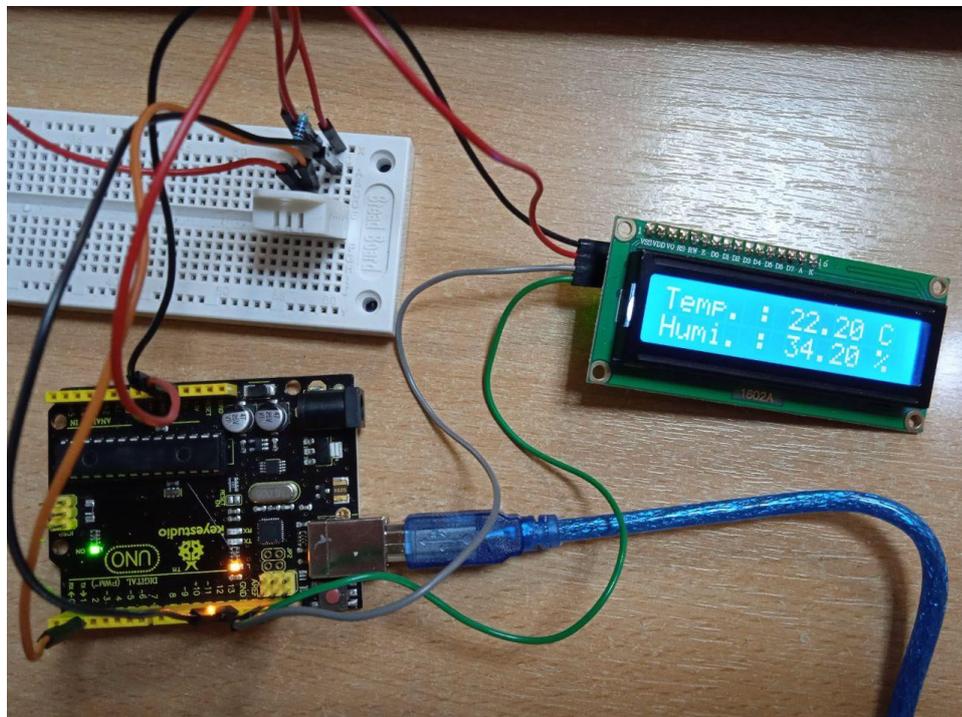


Figure 3.7 - connecting an Arduino control board to an LCD display using a board with a PCF 8574t chip

3.3. Subsystem for measuring the level of carbon dioxide

The problem of creating this subsystem is the complexity of obtaining objective data from measuring devices. Analyzing the existing technical solutions given in sections 2.2.1 – 2.2.3, it is worth noting that a great importance in the objectivity of data depends on the calibration process of these measuring devices. During the study, it was decided to conduct experiments with various measuring devices.

The first sensor that was used in the carbon dioxide measurement subsystem MQ-135, which is described in Section 2.2.1. following the recommendations specified in the popular scientific article [12], describing the principles of operation of this measuring device, the rzero parameter calibration process was carried out using the proposed program code. Changes in the zero parameter value are shown in Figure 3.8.



Figure 3.8-measured values of the rzero parameter

When developing this subsystem, a clock button was added to the system, the signal contact of which is connected to the third digital output of the Control Board.

The existing system program code was also modified. Let's look at the changes that were made.

It was reorganized and brought to a single format for determining the contacts of the control board that were connected to external devices. To determine the connection configuration, the preprocessor directive `#define` was used, so numeric values corresponding to the numbers of digital and analog pins were assigned to keywords (links) such as `mq135_pin`, `dht_pin`, and `button_pin`. The compiler replaces reference values with numeric values during compilation, which reduces the amount of memory used by the program compared to defining these values using variables or constants.

For the operation of the MQ-135 module, the `mq135.h` library was used, the use of which was mentioned in Section 2.2.1. due to the nature of the parameter change (fig. 3.8) the average value of 122.42 was selected, and this value was also entered in the `mq135.h` library file. In the main code of the program, it was also necessary to create an object of the class `MQ135`. Initialization of the class object is performed using the code `MQ135 gasSensor = MQ135(mq135_pin);` analyzing which it is worth noting that the constructor method requires the value of the analog output number, in our case given in the format of the constant `mq135_pin`. It is worth noting that the module has 4 pins, one of which (DOUT) was not used when creating this system, and the analog pin (aout) was connected to pin A0 on the control board.

To preserve these values measured during the operation of the system, the variables `h`, `t`, for measuring temperature and humidity, and the variable `ppm`, for measuring the concentration of carbon dioxide, also known as the millionth, were defined. All of the above variables have the float data type, which is used to store fractional values.

An additional global variable that has been added to the main program code is the mode variable of the `INT` (integers) data type. The purpose of this variable is to provide a display of various types of data, depending on the mode in which the system is currently operating. At this stage, two operating modes were implemented: the first one displays temperature and humidity data, and the second one displays concentration in millionths (ppm). The logic for changing modes is as follows. If the button is pressed, a voltage appears on the signal output of the button, which is approximately equal to the value of 5

volts. When reading The Voltage value using the `digitalRead(button_pin)` command; the voltage value corresponding to 5V will be represented as a logical unit. And at the moment when the `digitalRead(button_pin)` value is equal to 1, a mode change occurs. this logic is implemented by the following part of the code:

```
if (digitalRead(button_pin) == 1) {
    mode++;
    if (mode > 1) {
        mode = 0;
    }
}
```

Analyzing the above code, it is worth noting that it uses the `if` Branch operator with the condition specified above. If the condition is valid, the construct body written in curly brackets is executed, namely the `mode++` command, which increases the mode value by 1. this is followed by another `if` construct, which provides a limit on the number of modes. So if `mode = 1` and the button is pressed, the mode variable will be equal to 2, but the program does not have instructions for this mode and, accordingly, the program must return to the previous operating mode (`mode = 0`). To do this, use the following `If` check (`mode > 1`). If mode is equal to 2, this condition will be valid and the body of the second `if` construction will now be fulfilled, which will assign the mode variable the value 0. Thus, it is provided by switching between two operating modes (this design can also be adapted to a larger number of modes), the `log switch-case`:

```
switch (mode) {
    case 0:
        lcd.setCursor(0, 0);
        lcd.print("Temp. : ");
        lcd.print(t);
        lcd.print(" C ");
        lcd.setCursor(0, 1);
        lcd.print("Humi. : ");
        lcd.print(h);
```

```
    lcd.print(" % ");  
    break;  
case 1:  
    lcd.setCursor(0, 0);  
    lcd.print("MQ-135");  
    lcd.setCursor(0, 1);  
    lcd.print("CO2: ");  
    lcd.print(ppm);  
    lcd.print(" ppm");  
    break;  
}
```

This code allows you to compare the value of the mode variable with 0 or 1, and, depending on the value, choose one of two algorithms. If the mode variable is 0, The First Data Display algorithm described in the previous section will be executed, otherwise the display will display data about the type of sensor currently in use, the concentration of carbon dioxide, and the unit of measurement.

The results of implementing this subsystem with the MQ-135 sensor are shown in Figure 3.9. MQ-135 are shown in Figure 3.9.

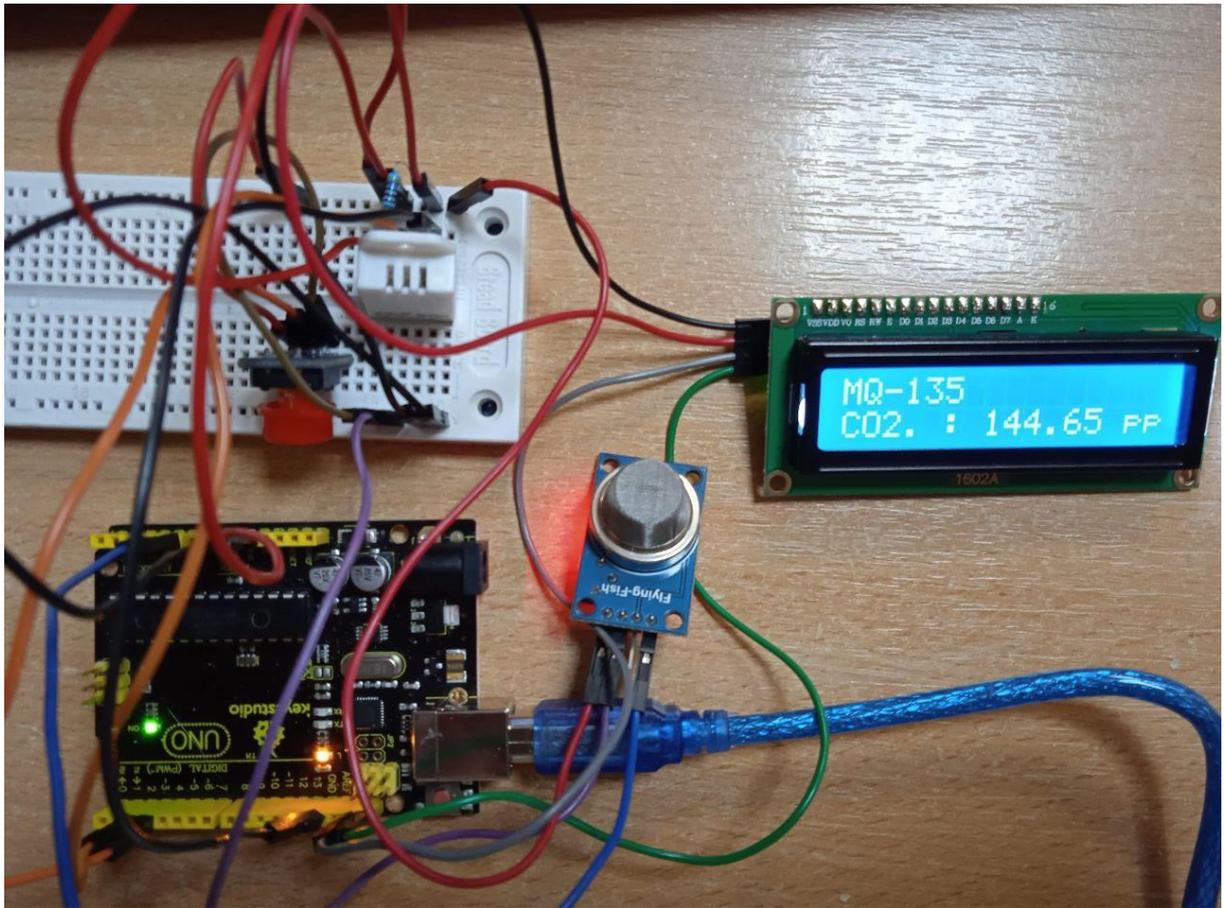


Figure 3.9-prototype of a temperature, humidity, and carbon dioxide monitoring system using DHT22 and MQ-135 modules

3.4. control subsystem for actuators

The main actuator for changing the level of carbon dioxide in a room is an electric motor, as part of the ventilation system. In this study, the possibility of using DC motors and brushless motors was considered, as well as several control schemes for these types of electric motors. It is worth noting that there is no single correct solution for choosing an engine and control scheme, because the choice depends on the type of room, its volume, economic resources, as well as the nature of changes in the level of carbon dioxide in the room. Therefore, this study suggests several schemes that, according to the author, can be used in the ventilation system.

Consider the use of DC motors as the executive body of the ventilation subsystem, general information about the control of DPS is given in the Article [21], which offers

several control options, one of which is the use of a transistor control circuit, and the other is the driver of the L298N motor.

According to [21], we can control the speed of a DC motor by simply controlling the input voltage of the motor, and the most common way to do this is to use a Pulse Width Modulation (PWM) signal. PWM, or Pulse Width Modulation, is a method that allows us to adjust the average voltage applied to an electronic device by quickly turning the power on and off. The average voltage depends on the duty cycle or time during which the signal was turned on, compared to the time during which the signal was turned off in a single time period. So, depending on the size of the motor, we can simply connect the Arduino PWM output to the base of a transistor or gate field effect transistor (MOSFET) and control the motor speed by controlling the PWM output. The Arduino low-power PWM signal turns on and off the gate on the field-effect transistor through which the powerful motor is controlled. A diagram of using a field-effect transistor to control the DPS speed is shown in Figure 3.9.

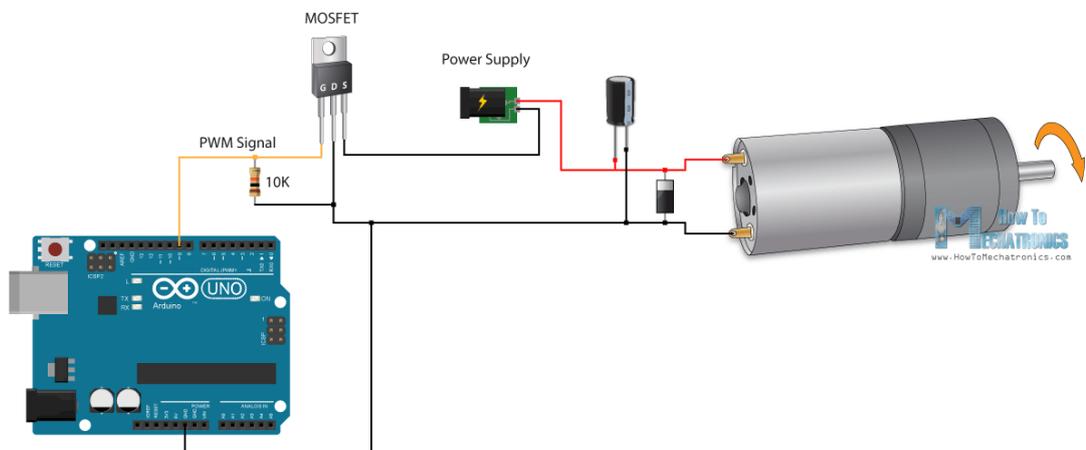


Figure 3.9-DPS speed control circuit using a field-effect transistor

On the other hand, to control the direction of rotation, we just need to change the direction of current flowing through the motor, and the most common way to do this is to use an H-shaped bridge. The H-shaped bridge circuit shown in Figure 3.10 contains four switching elements, transistors or MOSFETs, with a motor in the center that forms an H-shaped configuration. By activating two specific switches at the same time, we can change

the direction of current flow, thereby changing the direction of rotation of the motor. So if we combine these two methods, PWM and H-bridge, we can have full control over the DC motor. There are many DC motor drivers that have these features, and the L298N is one of them.

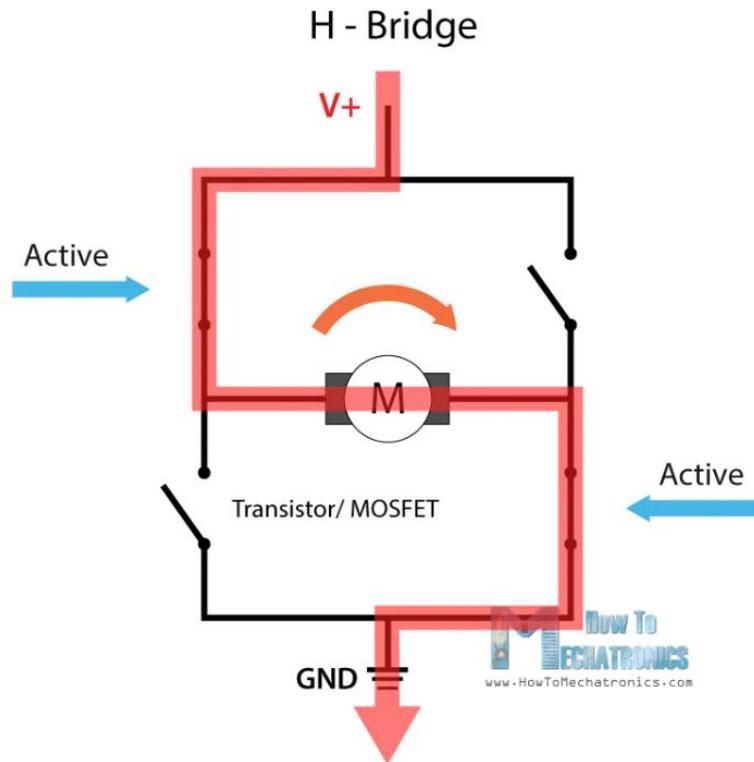


Figure 3.10-diagram of an H-shaped bridge

The L298N driver is a dual H-bridge motor driver that allows you to simultaneously control the speed and direction of two DC motors. The module can control DC motors with a voltage of 5 to 35 V with a peak current of up to 2 A is shown in Figure 3.11.

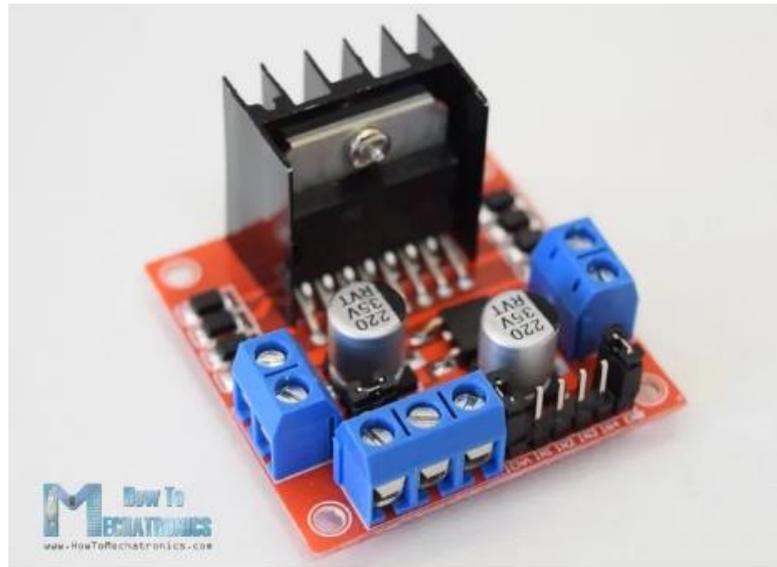


Figure 3.11-L298N engine driver

The module has two screw terminal blocks for Motors A and B, as well as another screw terminal block for the ground contact, VCC for the motor, and a 5V contact that can be input or output. Several DC motors can be used to provide more powerful ventilation. an example of such a connection is shown in Figure 3.12.

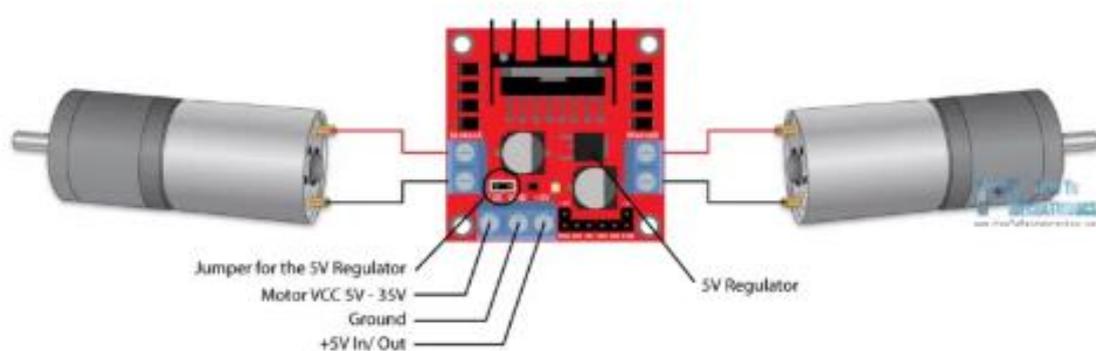


Figure 3.12-example of connecting two motors to the driver

The L298 motor driver can also be connected to an Arduino, to test it, we need the following components::

- L298N driver;
- 12V high torque DC motor;
- Arduino control board;

- breadboard and adapter wires.

Let's look at how you can modify the existing code in order to ensure engine operation when the permissible concentration of carbon dioxide in the air is exceeded.

First of all, you need to add the connection configuration. To do this, we used a similar method as in the previous section, using the `#define` preprocessor directive. Thus, we add three constants `in1`, `in2` and `enA`, which are equal to the numbers of the digital pins of the microcontroller to which these pins of the L298 module are connected. It is important to note that the `enA` PIN must be connected to the digital pin marked "~" or PWM (PWM), it will set the width of the control pulses and, accordingly, the speed of rotation of the motor.

In the `setup` function, you need to set the operating modes of all three pins `in1`, `in2`, and `enA` using the `pinMode ()` command, and specify the digital output number with the first argument (in our case, reference: `in1`, `in2`, `enA`), and specify the OUTPUT mode with the second argument. It is also advisable to set low signal levels on all three pins using the following code:

```
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
analogWrite(enA, 0);
```

The next step is to set the engine start-up condition(`sd`) to activate the ventilation system, as well as the occurrence of engines when the carbon dioxide level is at an acceptable level. The implementation of this algorithm can and should occur regardless of the mode of operation that is responsible for displaying data, so this functionality will be written outside the switch-case design described in the previous section.

In this study, it was proposed to use the hysteresis curve control law. This method also requires entering several additional variables (`turn_on_value` and `turn_off_value`), and its implementation using program code will look like this:

```
if (ppm > turn_on_value) {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    analogWrite(enA, 255);
```

```

}
if (ppm < turn_off_value) {
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    analogWrite(enA, 0);
}

```

In this study, we chose to use the `turn_on_value` and `turn_off_value` variables with an integer data type (`int`), the values of which are set during their initialization and set in accordance with the requirements described in Section 1.

The logic that provides the ascending part of the hysteresis curve characteristic is described in the first branch design. If the measured value of carbon dioxide is greater than the permissible value recorded in the `turn_on_value` variable, a high signal level is set on one of the digital pins (`in1` or `in3`), and a low signal level is set on the other, which in turn allows current to flow through the electric motor in one direction. The motor voltage level, which is proportional to the motor speed, is set by applying a PWM signal to the ENA digital pin. This functionality is provided using the `analogWrite ()` Command, which, in turn, accepts two parameters. The first parameter indicates the digital output number (`enA`), and the second parameter indicates the average voltage level, which is set in a relative value in the range from 0 to 255 (where 0 is 0V, and 255 is 5v). If you need to change the direction of rotation of the motor in the `digitalWrite()` commands for pins `in1` and `in2`, the voltage levels change from HIGH to LOW and vice versa.

According to the specified logic, The enabled state of the engine will remain until the ppm value decreases below the level set in the `turn_off_value` variable. At this point, the voltage values at pins `in1`, `in2`, and `enA` will be set to 0V. thus, the second if construct provides the descending part of the hysteresis curve.

It is worth noting that in addition to the control law, a linear law of setting the speed of rotation of an electric motor from changes in the concentration of carbon dioxide, or a combination of these two laws, can also be used, but the implementation of this model requires further research.

Also, in the framework of this study, the possibility of using brushless (BLDC) or Valve Motors was considered [22]. For example, consider a motor rated KV = 1000, which means that when a voltage of 7.4 volts is applied to the motor, the motor can reach a maximum RPM of $7.4 * 1000$, which is 7400 rpm. A similar engine is shown in Figure 3.13.



Figure 3.13-brushless motor

Brushless motors require an external source of electricity, and the most common way to power them is to use lithium polymer (LiPo) batteries. The number "s" of the battery indicates the number of battery cells, each of which has a voltage of 3.7 V. For example, take the 3s LiPo battery, which has 3 cells that provide a supply voltage of 11.1 v.

To control such motors, drivers or ESC speed controllers are used. On the one hand, the ESC has three wires that control the three phases of the motor, and on the other hand, it has two wires, VCC and GND, for power, shown in Figure 3.14.

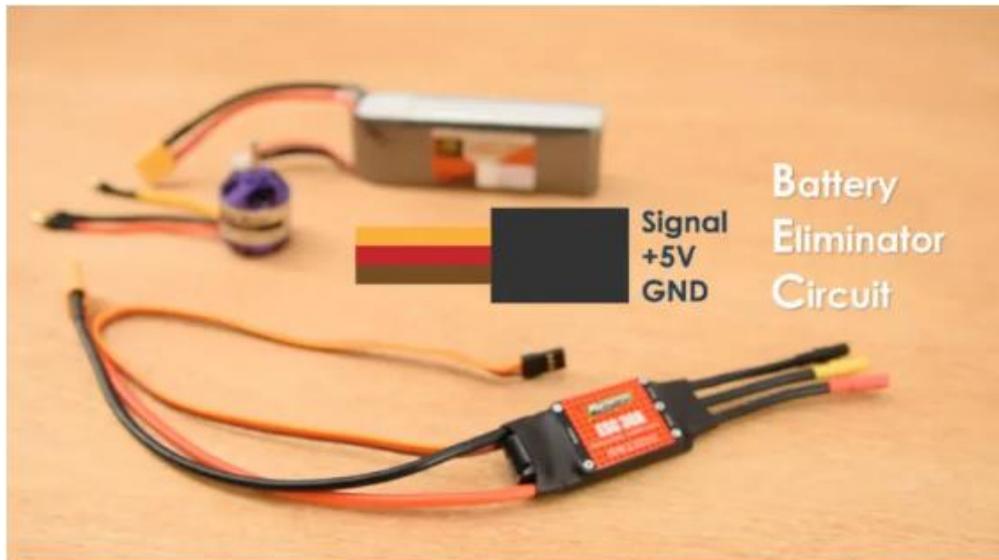


Figure 3.14 – ESC Driver

When implementing this type of electric motor in the actuator control subsystem, it generates a PWM signal with a frequency of 50 Hz, and depending on the pulse width or duration of the high state, which should vary from 1 millisecond to 2 milliseconds, the ESC will control the motor from minimum to maximum revolutions. To test a brushless motor, you need to assemble a circuit similar to the one shown in Figure 3.15 with a set of devices:

- Brushless motor ESC 30A
- Li-Po battery
- Arduino control board
- Breadboard and adapter wires
- Carbon dioxide measurement subsystem
- Display subsystem (optional)
- Temperature and humidity measurement subsystem (optional)

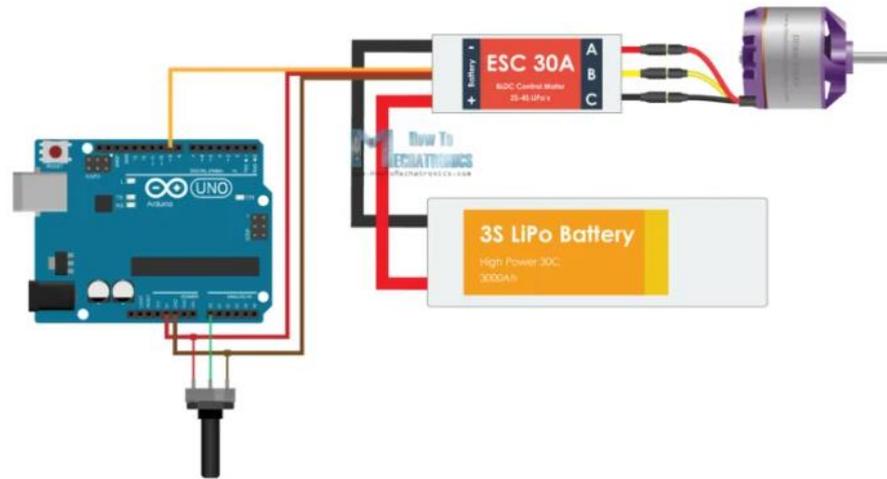


Figure 3.15-example of connecting a brushless motor to an Arduino control board

It is worth noting that the diagram shown in Figure 3.15 does not correspond to the configuration of the proposed system and is used to illustrate the connection of the BLDC motor, and is also taken from a popular science Article [22], but this illustration allows you to understand the connection configuration.

To control a BLDC motor with an ESC driver, it is convenient to use the same library that is used to control the servo motor, namely Servo.h. You must create an object of the Servo class next to the creation of other objects. The name of the object must correspond to its purpose, so let's call it ESC, respectively, the code that provides this action:

```
Servo ESC;
```

Also, the ESC signal pin must be connected to the PWM pin on the control board, as well as specify a link to it using a line of code. `#define esc_pin 6`.

In the setup function, you also need to call the attach method, which requires one parameter – the number of the digital output on the Control Board to which the driver signal contact is connected, and two optional parameters, namely, the minimum and maximum duration of the PWM pulse in microseconds. Thus, calling the attach method will look like:

```
ESC.attach(9,1000,2000)
```

We modify part of the code from the previous example so that the microcontroller now provides control according to the law of the hysteresis curve using the BLDC motor and ESC driver:

```
if (ppm > turn_on_value) {  
    ESC.write(180);  
}  
if (ppm < turn_off_value) {  
    ESC.write(0);  
}
```

In this case, the values 0 and 180 correspond to the minimum and maximum duration of the PWM signal and will set the minimum (0) and maximum motor speed.

However, there are a few things worth noting in the Article [22]. At the start of motor power supply, the signal value must be the same or less than the minimum value of 1 millisecond. This is called setting the ESC driver to the cocking, and the engine Beeps a confirmation signal to let us know that it is correctly set to the cocking. In the event that we have a higher power value, which means that we have a throttle, the ESC will not start the engine until we lower the throttle to the correct minimum value. This is very convenient from the point of view of safety, because the engine will not start if we have raised the power when feeding.

You also need to calibrate the ESC. Each ESC has its own upper and lower points, and they may differ slightly. For example, the lowest point may be 1.2 milliseconds, and the highest point may be 1.9 milliseconds. In this case, our throttle will do nothing in the first 20% until it reaches its lowest value of 1.2 milliseconds.

3.5. Conclusion to Section 3

We chose the use of a self-made system of analysis and control of microclimate parameters, it was based on the system that was presented at the VIII All-Ukrainian scientific and practical conference "Electronic and Mechatronic Systems: Theory, innovation, practice. To measure temperature and humidity, we chose the DHT 22

module, the choice was justified by the fact that the error in temperature measurements is 4 times less than that of DHT11, and the error in humidity measurement is 2 times less. To connect the module, a pull-up resistor with a resistance of 4.7 Kohm was used. When developing the software, we selected the code from the Article [17]. For data visualization, we chose to use the LCD 1602 display, because although led and sound systems are cheaper, it will be more informative to read data from the display. To reduce the number of pins used by the display on the Arduino board, it was decided to use the PCF 8574t expansion board. The program code specified in Section 2.2.4 has been used and modified to add display subsystems. In order to be able to measure the level of carbon dioxide, the MQ-135 module was added, and a clock button was added to switch the parameters on the display from temperature and humidity to the content of carbon dioxide in the air. In order to be able to change the carbon parameters in the room, there is an electric motor as part of the ventilation system. To do this, you can choose a DC motor, or a brushless one, but there is no single correct solution for choosing a motor and its control scheme, since the choice depends on the size of the room in which the system will be used, the nature of changes in carbon dioxide, or economic resources. For our project, a DC motor with an L298N motor driver was proposed, with this driver we can control the motor via PWM. Also, to control the direction of rotation, we need to use an H-shaped bridge. Alternatively, for more powerful ventilation, you can connect multiple motors to the L298N engine driver.

Додаток Б. Програмний код системи моніторингу та керування параметрами мікроклімату

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <dht.h>
#include <MQ135.h>

#define button_pin 3
#define mq135_pin A0
#define dht_pin 2

#define enA 9
#define in1 8
#define in2 7

const int turn_on_value = 400;
const int turn_off_value = 360;

MQ135 gasSensor = MQ135(mq135_pin);
dht DHT;

float t;
float h;
float ppm;

int mode = 0;

LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
void setup() {  
  Serial.begin(9600);  
  
  pinMode(button_pin, INPUT);  
  
  pinMode(enA, OUTPUT);  
  pinMode(in1, OUTPUT);  
  pinMode(in2, OUTPUT);  
  
  digitalWrite(in1, LOW);  
  digitalWrite(in2, LOW);  
  analogWrite(enA, 0);  
  
  lcd.init();  
  lcd.clear();  
  lcd.backlight();  
}  
  
void loop() {  
  
  ppm = gasSensor.getPPM();  
  Serial.println(ppm);  
  
  int readData = DHT.read22(dht_pin); // Reads the data from the sensor  
  t = DHT.temperature;                // Gets the values of the temperature  
  h = DHT.humidity;                   // Gets the values of the humidity  
  
  if (digitalRead(button_pin) == 1) {  
    mode++;  
    if (mode > 1) {
```

```
    mode = 0;
  }
  delay(500);
}

switch (mode) {
  case 0:
    lcd.setCursor(0, 0);
    lcd.print("Temp. : ");
    lcd.print(t);
    lcd.print(" C ");
    lcd.setCursor(0, 1);
    lcd.print("Humi. : ");
    lcd.print(h);
    lcd.print(" % ");
    Serial.print("Temp. : ");
    Serial.print(t);
    Serial.println(" C ");
    Serial.print("Humi. : ");
    Serial.print(h);
    Serial.println(" % ");
    break;
  case 1:
    lcd.setCursor(0, 0);
    lcd.print("MQ-135");
    lcd.setCursor(0, 1);
    lcd.print("CO2: ");
    lcd.print(ppm);
    lcd.print(" ppm");
    break;
```

```
}

if (ppm > turn_on_value) {
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  analogWrite(enA, 255);
}

if (ppm < turn_off_value) {
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  analogWrite(enA, 0);
}

delay(2000); // Delays 2 seconds, as the DHT22 sampling rate is 0.5Hz
lcd.clear();
}
```

Додаток В. Презентація кваліфікаційної роботи



Національний університет «Полтавська політехніка імені Юрія Кондратюка»
National University "Yuri Kondratyuk Poltava Polytechnic"

Косинков Антон

Система моніторингу та керування параметрами мікроклімату в приміщеннях на основі стандартів ЄС

Науковий керівник:
Боряк Б.Р.
кандидат технічних наук

Полтава 2022

Формалізована постановка задач дослідження

- **Мета:** Розробленняга дослідження методів моніторингу та керування параметрами мікроклімату у виробничих приміщеннях
- **Задачі дослідження**
 1. Дослідження стандартів параметрів мікроклімату України та ЄС
 2. Аналіз впливу зміни параметрів мікроклімату на стан людини.
 3. Дослідження технічних засобів для забезпечення моніторингу параметрів мікроклімату
 4. Дослідження технічного забезпечення систем контролю мікроклімату
 5. Підсистема моніторингу температури та вологості навколишнього середовища
 6. Підсистема індикації параметрів навколишнього середовища
 7. Підсистема вимірювання рівня діоксиду вуглецю.
 8. Підсистема керування виконавчими пристроями
- **Об'єкт дослідження** процес моніторингу та керування параметрами мікроклімату у виробничих приміщеннях .
- **Предмет дослідження** Чисельні показники параметрів мікроклімату

Дослідження стандартів параметрів мікроклімату України та ЄС

Період року	Категорія робіт	Температура повітря, °С	Відносна вологість, %	Швидкість руху, м/с
Холодний період року	Легка 1а	22-24	60-40	0,1
	Легка 1б	21-23	60-40	0,1
	Середньої важкості Іа	19-21	60-40	0,2
	Середньої важкості Іб	17-19	60-40	0,2
	Важка ІІІ	16-18	60-40	0,3
Теплий період року	Легка 1а	23-25	60-40	0,1
	Легка 1б	22-24	60-40	0,2
	Середньої важкості Іа	21-23	60-40	0,3
	Середньої важкості Іб	20-22	60-40	0,3
	Важка ІІІ	18-20	60-40	0,4

Таблиця 1. Оптимальні величини температури, відносної вологості та швидкості руху повітря в робочій зоні виробничих приміщень

3

Дослідження впливу CO₂ на людину

Рівень CO ₂ (ppm)	Вплив на людину
300-400 ppm	Оптимальне значення
400-600 ppm	Норма для роботи в офісі без зниження продуктивності праці. На організм ніяк не впливає.
До 600 ppm	Норма для спальні, дитячого садка і школи.
600-1000 ppm	Падає продуктивність праці, складніше концентрувати увагу і сприймати інформацію. З'являється головний біль.
1000-1500 ppm	Людина стає млявою, падає концентрація уваги, складніше приймати рішення, збільшується кількість помилок в роботі. Виникає ризик розвитку хронічної втоми.
Від 1500	Сильний головний біль. Зникає бажання працювати через втому.

Таблиця 2. Вираженість симптомів залежно від концентрації рівня CO₂

4

Дослідження технічних засобів для забезпечення моніторингу параметрів мікроклімату



Рисунок 1. Вимірювач рівня CO2 TFA «AIRCO2NTROL 5000»



Рисунок 2. DIY Монітор якості повітря – вимірювач

Недоліки:

- Висока ціна
- Не можна підключити виконавчі пристрої

Переваги:

- Компактність
- Точність вимірювання

Недоліки:

- Складність реалізації
- Необхідність калібрування датчиків

Переваги:

- Можливість інтеграції зовнішніх виконавчих пристроїв

5

Вимірювальні, виконавчі та керуючі пристрої систем моніторингу та керування параметрами мікроклімату



Рисунок 3. Плата керування Arduino Uno

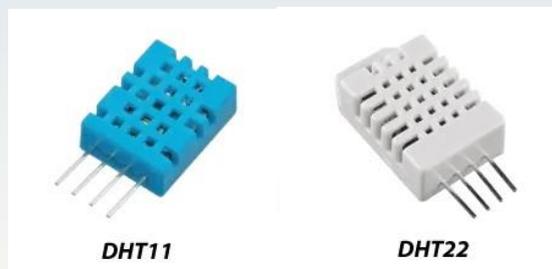


Рисунок 4. Модулі вимірювання температури та вологості



Рисунок 5. Вимірювальні пристрої рівня діоксиду вуглецю

6

Підсистема моніторингу температури та вологості навколишнього середовища

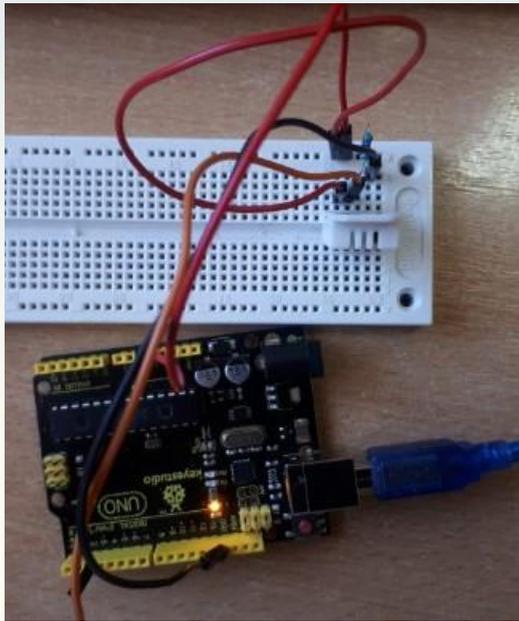


Рисунок 6. Ілюстрація конфігурації підключення елементів підсистеми вимірювання температури та вологості навколишнього середовища

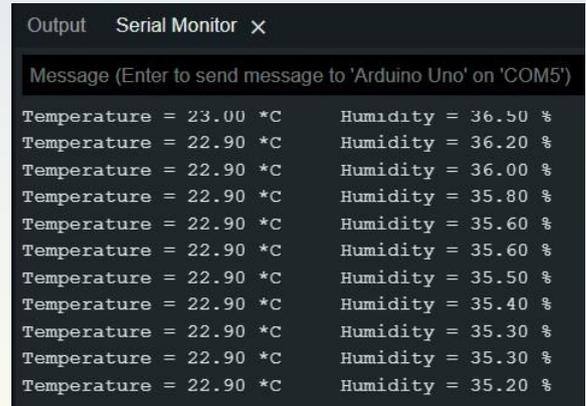


Рисунок 7. Передача вимірних даних за допомогою вимірювального пристрою через протокол послідовного порту

Підсистема індикації параметрів навколишнього середовища

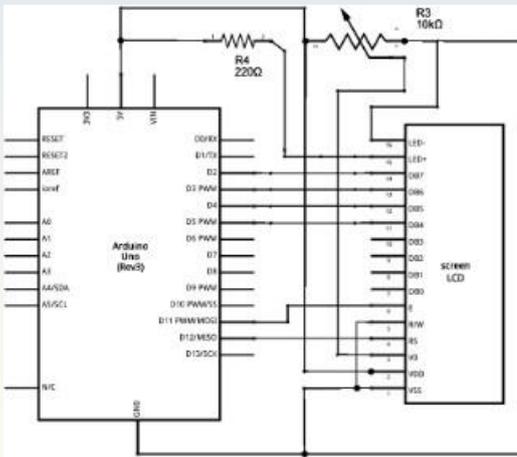


Рисунок 8. Схема підключення дисплею до Arduino Uno

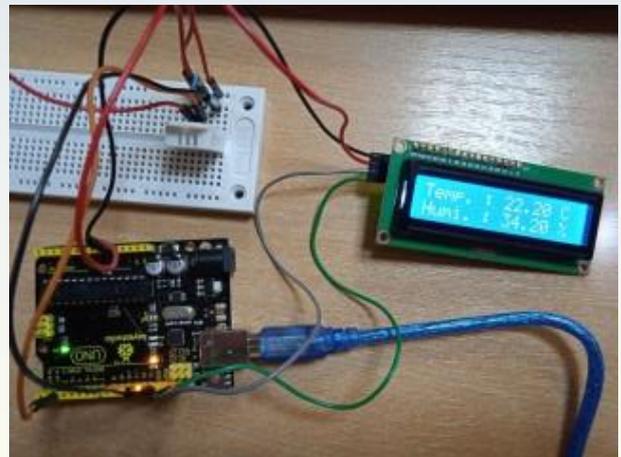


Рисунок 9. Ілюстрація конфігурації підключення дисплею до Arduino Uno

Підсистема вимірювання рівня діоксиду вуглецю

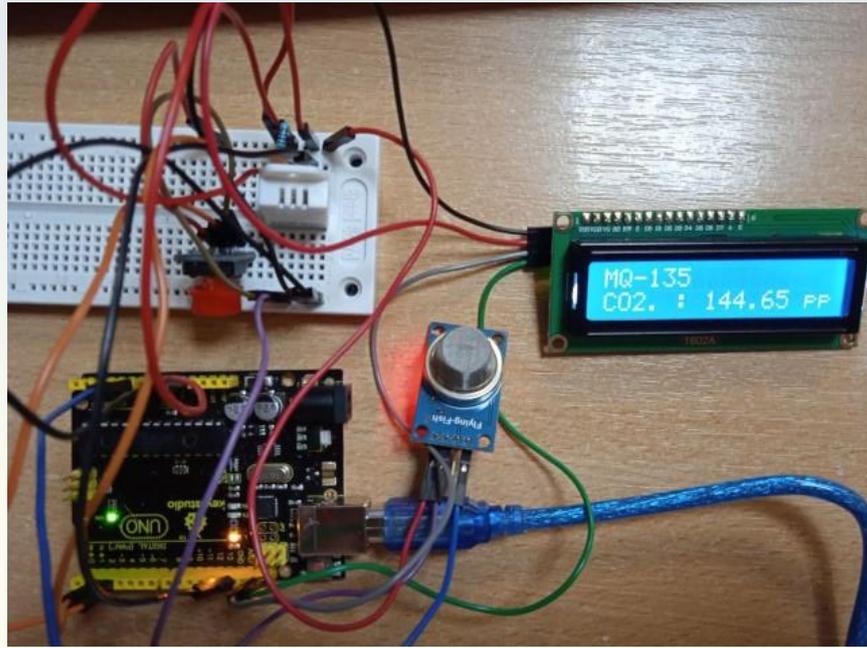


Рисунок 10. Ілюстрація системи з підключеним датчиком рівня діоксиду вуглецю

9

Підсистема керування виконавчими пристроями

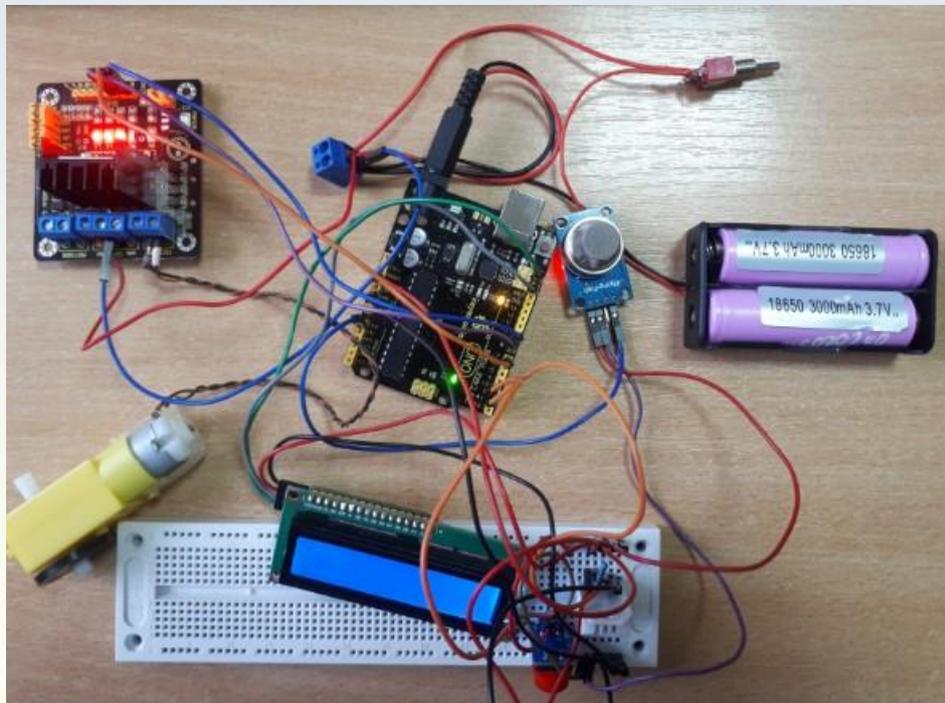


Рисунок 11. Конфігурація підключення підсистеми керування виконавчими пристроями

10

Висновки

1. Було проведено аналіз стандартів параметрів мікроклімату України та країн ЄС та впливу зміни параметрів на людину (Розділ 1).
2. Визначено оптимальні параметри мікроклімату для виробничих приміщень, а також значення параметрів, які можуть вказувати на небезпечні умови для здоров'я людини (Розділ 1).
3. Досліджено системи та технічні засоби, що використовуються в системах моніторингу та керування параметрів мікроклімату (Розділ 2).
4. Визначено пристрої для реалізації системи моніторингу та керування параметрами мікроклімату та здійснено дослідження принципів роботи пристроїв керування, вимірювання та виконавчих пристрої (Розділ 2).
5. Розроблено підсистему вимірювання температури та вологості середовища, підсистема індикації, підсистема вимірювання рівня діоксиду вуглецю, підсистема керування виконавчими пристроями (Розділ 3).
6. Розроблено та реалізовано у вигляді програмного коду алгоритми системи моніторингу та керування параметрами мікроклімату, що забезпечують роботу розробленої системи (Розділ 3).