

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютеризованих систем управління

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

_____Литвиненко О.Є.

«___» _____ 2021 р.

ДИПЛОМНИЙ ПРОЄКТ
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
"БАКАЛАВР"

Тема: «Аналізатор параметрів повітря та ґрунту»

Виконавець: _____Олексенко Максим Сергійович

Керівник: _____Коба Олена Вікторівна

Нормоконтролер: _____Тупота Євгеній Вікторович

Київ 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютеризованих систем управління

Освітнього ступеня бакалавр

Напрямок (спеціальність) 123 "Комп'ютерна інженерія"

(шифр, найменування)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Литвиненко О.Є.

« _____ » _____ 2021 р.

ЗАВДАННЯ

на виконання дипломної роботи (проєкту)

Олексенка Максима Сергійовича

(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема дипломної роботи (проєкту): «Аналізатор параметрів повітря та ґрунту»

затверджена наказом ректора від «04» лютого 2021 р. №135\ст.

2. Термін виконання роботи (проєкту): з 17.05.2021 по 20.06.2021

3. Вихідні дані до роботи (проєкту): мікроконтролер *Arduino Mega2560*, датчики температури та вологості, дисплей, резистори, світлодіоди, фоторезистори, мова програмування *C++*, середовище програмування *Arduino IDE*

4. Зміст пояснювальної записки:

1) Властивості аналізатора параметрів повітря та ґрунту;

2) Технічні та програмні засоби аналізатора параметрів повітря та ґрунту;

3) Алгоритмічна розробка аналізатора параметрів повітря та ґрунту;

4) Розробка аналізатора параметрів повітря та ґрунту;

5. Перелік обов'язкового графічного (ілюстративного) матеріалу:

1) Нумерація та значення виходів на платі *Arduino Mega2560*;

2) Діаграма послідовності;

3) Діаграма станів;

4) Діаграма прецедентів;

5) Рисунок аналізатору параметрів повітря та ґрунту після підключення датчиків.

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1	Провести аналіз предметної області та програмних продуктів за темою дипломного проекту	17.05.2021 – 19.05.2021	
2	Підготувати перший розділ пояснювальної записки	20.05.2021 – 21.05.2021	
3	Розробити другий розділ пояснювальної записки	22.05.2021 – 24.05.2021	
4	Розробити алгоритми роботи приладу	25.05.2021	
5	Підготувати третій розділ пояснювальної записки	26.05.2021 – 27.05.2021	
6	Розробка аналізатора параметрів повітря та ґрунту	28.05.2021 – 07.06.2021	
7	Підготувати четвертий розділ пояснювальної записки	08.06.2021 – 09.06.2021	
7	Завершити оформлення пояснювальної записки	10.06.2021 – 12.06.2021	
8	Підготувати презентацію та графічні матеріали	13.06.2021 – 14.06.2021	

Дата видачі завдання: 17.05.2021р.

Керівник дипломної роботи (проєкту) Коба О.В.

(підпис керівника)

Завдання прийняв до виконання Олексенко М. С.

(підпис випускника)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Аналізатор параметрів повітря та ґрунту»: 61 с., 56 рис., 15 літературних джерел, 1 додаток.

Ключові слова: *ARDUINO*, МІКРОКОНТРОЛЛЕР, АНАЛІЗ, ДАТЧИКИ, ВОЛОГІСТЬ, ТЕМПЕРАТУРА, ОСВІТЛЕННЯ, ПАРАМЕТРИ ПОВІТРЯ ТА ҐРУНТУ, АНАЛІЗАТОР.

Об'єкт дослідження – параметри повітря та ґрунту.

Предмет дослідження – аналізатор параметрів повітря та ґрунту.

Мета дослідження – створення приладу для аналізу параметрів повітря та ґрунту.

Проведено аналіз даної області, визначено проблематику даної проблеми та створено систему, яка виконує аналіз параметрів повітря та ґрунту, обчислює та виводить на дисплей дані про температуру та вологість повітря, температуру та вологість ґрунту та рівень освітлення.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 ВЛАСТИВОСТІ ТА ЗАСТОСУВАННЯ АНАЛІЗАТОРА ПАРАМЕТРІВ ҐРУНТУ ТА ПОВІТРЯ	9
1.1. Ідея створення.....	9
1.2. Властивості та застосування аналізатора	10
1.3. Висновки до розділу	12
РОЗДІЛ 2 ТЕХНІЧНІ ТА ПРОГРАМНІ ЗАСОБИ АНАЛІЗАТОРА ПАРАМЕТРІВ ПОВІТРЯ ТА ҐРУНТУ	13
2.1. Технічні засоби реалізації проекту.....	13
2.2. Програмні засоби.....	28
2.3. Висновки до розділу	31
РОЗДІЛ 3 АЛГОРИТМІЧНА РОЗРОБКА АНАЛІЗАТОРА ПАРАМЕТРІВ ПОВІТРЯ ТА ҐРУНТУ	32
3.1. Діаграма прецедентів	32
3.2. Діаграма послідовності.....	34
3.3. Діаграма класів	34
3.4. Діаграма станів	35
3.5. Діаграма кооперації.....	36
3.6. Висновки до розділу	37
РОЗДІЛ 4 РОЗРОБКА АНАЛІЗАТОРА ПАРАМЕТРІВ ПОВІТРЯ ТА ҐРУНТУ	38
4.1. Програмні бібліотеки розробки	38
4.2. Апаратне забезпечення для реалізації пристрою:.....	38
4.3. Логіка роботи пристрою	39
4.4. Практична реалізація аналізатора параметрів ґрунту та повітря	40
4.5. Висновки до розділу	59
ВИСНОВКИ.....	60

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61
ДОДАТОК А.....	62

ВСТУП

Рослинництво – одна із головних галузей сільського господарства та харчової промисловості. Вирощуванням рослин займаються як фермери на великих сільськогосподарських підприємствах, так і люди на власних земельних ділянках або навіть на підвіконнях у квартирі. Найбільша складність правильного вирощування рослин – це недотримання відповідних кліматичних норм.

Оптимальна вологість ґрунту є важливою передумовою адекватного росту сільськогосподарських культур та вищих урожаїв. Вологість необхідна не тільки для відновлення водного балансу, але і для регулювання температури. У процесі терморегуляції в рослинах випаровується до 99% отриманої вологи і лише 0,2-0,5% використовується для формування вегетативної маси. У той же час оптимальна вологість ґрунту для посівів змінюється залежно від стадій росту та кліматичних умов.

Температура ґрунту є одним з найважливіших факторів у житті рослин. Схожість насіння, розвиток кореневої системи, життєдіяльність ґрунтової мікрофлори, засвоєння продуктів мінерального живлення корінням тощо сильно залежать від температури ґрунту. Зі збільшенням температури ґрунту активізуються всі ці процеси. Значне зниження температури ґрунту спричиняє загибель озимих злакових, багаторічних та плодкових дерев.

Наприклад насіння більшості сільськогосподарських культур проростає при температурі 3 - 5 °C, а от рису, бавовни тощо, вимагає набагато більш високих температур.

Тому **актуальність** роботи полягає саме у вирішенні проблеми дотримання кліматичних норм, сигналізації людей про відхилення від цієї норми.

Об'єкт дослідження – параметри повітря та ґрунту.

Предмет дослідження – аналізатор параметрів повітря та ґрунту.

Метою роботи є створення приладу для аналізу параметрів повітря та ґрунту.

Задачі для досягнення поставленої мети:

1. Аналіз основних параметрів повітря та ґрунту;
2. Розробка вимог до приладу та інструкції користувача;
3. Визначення технічних та програмних засобів для реалізації пристрою;
4. Розробка *UML*-діаграм відповідно до вимог та роботи системи;
5. Розробка приладу на розроблених попередніх завдань.

Робота складається зі вступу, чотирьох розділів, що описують технічні особливості розробку даного приладу та кожного із його компонентів зокрема, висновку та додатку, який містить код роботи приладу. При виконанні даного проекту було використано різноманітні джерела інформації (веб-сайти, наукові статті та інші).

РОЗДІЛ 1
ВЛАСТИВОСТІ ТА ЗАСТОСУВАННЯ АНАЛІЗАТОРА ПАРАМЕТРІВ
ПОВІТРЯ ТА ҐРУНТУ

1.1. Ідея створення

У наш час досить поширена тенденція до ведення здорового способу життя, а зокрема до вживання здорової та органічної їжі рослинного походження, якість якої не викликає сумнівів. Для того щоб бути впевненим у якості продукції, що вживають, багато людей власноруч хочуть вирощувати продукти споживання. Ідея створення аналізатора параметрів ґрунту та повітря полягає у створенні приладу для полегшення і певної автоматизації процесу вирощування рослин для потреб людини, приладу, який показуватиме рівень основних характеристик навколишнього середовища для забезпечення гарних умов для росту і розвитку рослин. Люди хочуть вирощувати не тільки ті культури, що ростуть на території нашої країни, а і різні екзотичні види, за якими необхідно ще більше догляду та створення комфортних умов для росту і розвитку.

Для вирощування рослин важливими є 5 факторів: освітлення (дуже важливий фактор, адже деякі рослини люблять бути у затінку, деякі – бути на сонці), температура та вологість ґрунту (різні рослини були винайдені у різних кліматичних умовах, деякі у тропічних лісах з великою вологою, деякі у засушених регіонах нашої планети), температура та вологість повітря. Задача приладу - виведення інформації про стан даних факторів, у разі сухого ґрунту або низької температури, повідомляти про це звуковим сигналом та інформацією на дисплеї.

Кафедра КСУ				НАУ 21 12 15 000 ПЗ						
<i>Виконав</i>	Олексенко М.С.			Властивості та застосування аналізатора параметрів повітря та ґрунту	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>			
<i>Керівник</i>	Коба О.В.				Д		9	61		
<i>Консульт.</i>					СП-436Б 123					
<i>Норм. контр.</i>	Тупота Є.В.									
<i>Зав. Каф.</i>	Литвиненко О.Є.									

1.2. Властивості та застосування аналізатора

Назва продукту: Аналізатор параметрів ґрунту та повітря.

Задача пристрою: виведення інформації про рівень освітлення, вологості і температури ґрунту та вологості і температури повітря.

Область застосування приладу: теплиці для промислового вирощування рослин, теплиці для вирощування рослин для власного споживання, парники різних видів, також люди зможуть використовувати його і на вулиці та на підвіконні у квартирах чи приватних будинках.

Цільова аудиторія: аналізатор параметрів ґрунту та повітря – прилад не складний у використанні, тому користуватися можуть всі люди, які хочуть займатися вирощуванням рослин та слідкувати за їх станом. Функціональні вимоги:

- Зчитування інформації з датчиків;
- Вивід температури і вологості повітря на дисплей;
- Вивід температури і вологості ґрунту на дисплей;
- Вивід інформації про рівень освітлення на дисплей;
- Вивід інформації про рівень освітлення за допомогою різнокольорових діодів;
- Звукове сповіщення про відхилення параметрів від норми;
- Сповіщення на дисплеї про відхилення від норми параметрів.

Сценарії використання:

1. Помістити датчик для вимірювання вологості ґрунту в землю

Фото датчика:



Рис. 1.1. Датчик температури *ds18b20*

2. Помістити датчик для вимірювання температури ґрунту в землю.

Фото датчика:



Рис. 1.2. Датчик вологості ґрунту V1.2

3. Увімкнути в джерело живлення прилад.

4. Дочекатися виводу даних про температуру, вологість та освітлення(у разі відхилення параметрів від норми, замість даних про температуру та вологість ґрунту буде звуковий сигнал та повідомлення на дисплеї).

Вимоги до пристрою: надійність у використанні, точність у виконанні функціональних вимог, достовірність надання даних про фактори, які досліджуються.

Мета розробки пристрою: даний пристрій був розроблений для закріплення теоретичних та практичних навичок з програмування та конструювання приладів, для здобування теоретичних та практичних навиків розробки проектів на базі *Arduino*. Також метою розробки даного проекту було створення пристрою для особистого користування у повсякденному житті.

Умови експлуатації: для ремонту та обслуговування пристрою необхідно звертатися до спеціалістів з електроніки або розробників. Експлуатація повинна відбуватися відповідно до технічних умов експлуатації електронно обчислювальних машин. Користуватися може людина з початковими знаннями експлуатації електронних пристроїв. Для коректної та тривалої роботи аналізатора параметрів ґрунту та повітря необхідно дотримуватися сценарію

експлуатації під час роботи з даним приладом. При експлуатації також необхідно бути обережним зі схемами, які знаходяться в даному приладі.

1.3. Висновки до розділу

В даному розділі було розглянуто основні аспекти використання приладу, його функціональні вимоги, сценарії користування, було описано ідею розробки приладу, було визначено, що прилад буде використовуватись для дослідження рівня різних факторів, навколишнього середовища, а саме, рівня освітлення, температури та вологості ґрунту та повітря. Також під час виконання даного розділу було визначено вплив вимірних факторів на рослини та визначено користь, яку принесе даний прилад людям.

РОЗДІЛ 2

ТЕХНІЧНІ ТА ПРОГРАМНІ ЗАСОБИ АНАЛІЗАТОРА ПАРАМЕТРІВ ПОВІТРЯ ТА ҐРУНТУ

2.1. Технічні засоби реалізації проекту

2.1.1. *Arduino Mega 2560*

Arduino Mega 2560 – прилад на основі мікроконтролера *ATmega2560*, який можна використовувати для багатьох цікавих проектів. Дана плата може підключатися через вихід *USB* або від зовнішнього джерела живлення.

Arduino Mega 2560 – найпотужніша плата на базі *Arduino*, вона має більше аналогових та цифрових виходів (16 та 54 відповідно) за свій популярніший аналог *Arduino Uno* та має більшу флеш-пам'ять (256Кб та 32Кб відповідно). Програмування плати відбувається в середовищі розробки *Arduino IDE*. *ATmega* розроблений таким чином, що перезавантаження виконується самою програмою перед написанням нового коду, а не натисканням кнопки на платформі. Одна з ліній *DTR* (управління потоком даних) *ATmega8U2* підключена до контакту скидання мікроконтролера *ATmega2560* через конденсатор 100 нФ. Активуючи цю лінію, тобто подаючи сигнал з низьким рівнем, мікроконтролер скидається.

Технічні характеристики плати *Arduino Mega2560*:

- Мікроконтролер: *ATmega2560*;
- Робоча напруга: 5V;
- Рекомендована вхідна напруга: 7 – 12V;
- Максимальна вхідна напруга: 6-20V;
- Цифрові входи/виходи: 54;

Кафедра КСУ				НАУ 21 12 15 000 ПЗ				
<i>Виконав</i>	Олексенко М.С.			Технічні та програмні засоби аналізатора параметрів повітря та ґрунту	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>	
<i>Керівник</i>	Коба О.В.				Д	13	61	
<i>Консульт.</i>					СП-436Б 123			
<i>Норм. контр.</i>	Тупота Є.В.							
<i>Зав. Каф.</i>	Литвиненко О.Є.							

- Аналогові входи: 16;
- Постійна сила струму на вході/виході: 40mA;
- Сила струму для виходу 3.3V: 50mA;
- Флеш-пам'ять: 256 КБ;
- Пам'ять пристрою для завантаження: 8 КБ;
- Оперативна пам'ять (SRAM): 8 КБ;
- Енергозалежна пам'ять (EEPROM): 4 КБ;
- Тактова частота: 16 МГц;
- Виходи Arduino підключені до АПЦ: 16;
- Виходи з підтримкою ШИМ: 15;
- Розрядність АЦП плати: 10 біт;
- Максимальна вихідна сила струму на виході 5V: 800mA;
- Розміри плати: 101×53 мм;
- Вага: 45 гр.

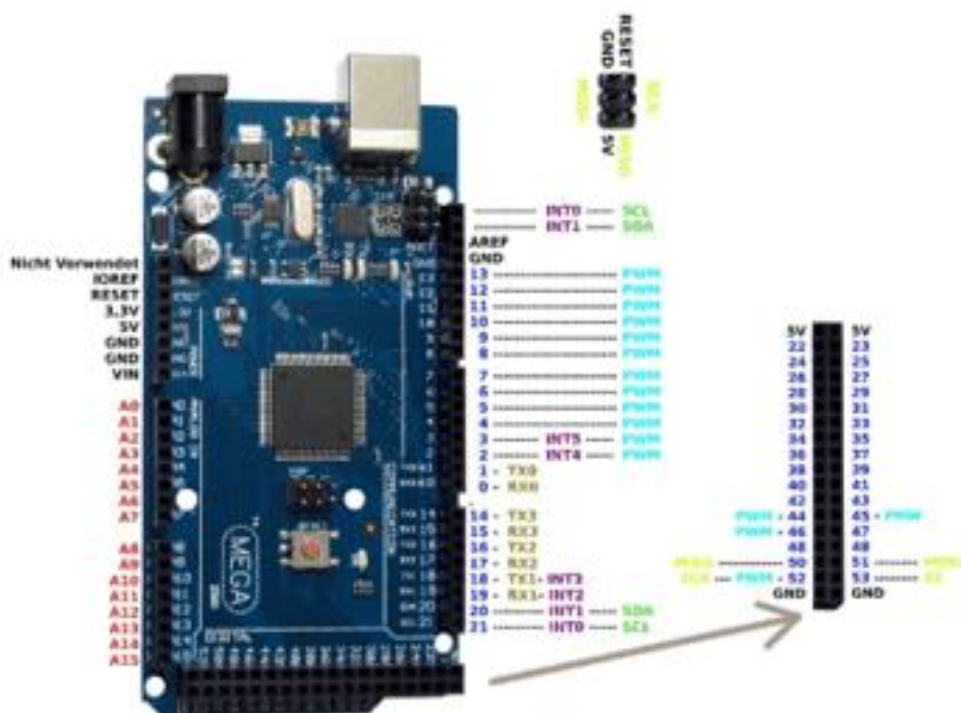


Рис. 2.1. Нумерація та значення виходів на платі *Arduino Mega 2560*

На рисунку 2.1. синім кольором виділено цифрові входи/виходи, тобто ті, що можуть приймати цифровий сигнал. Цифровий сигнал – сигнал, що

передається у вигляді двох станів 0 та 1, тобто є сигнал або немає сигналу. Червоним кольором позначено аналогові входи/виходи, тобто ті, що працюють із аналоговим сигналом. Аналоговий сигнал – це сигнал, що подається безперервно у вигляді хвилі, як і вся інформація у природі, наприклад світлові та звукові хвилі. Чорним кольором виділені входи/виходи для подачі живлення на плату або із плати на датчики чи різні прилади. Жовтим кольором – входи/виходи, які можна підключити по шині *SPI* (*SPI* – послідовний синхронний інтерфейс передачі даних, на відміну від *I2C* має більше проводів для підключення, але має більшу швидкість передачі даних), зеленим по шині *I2C* (*I2C* – послідовна шина із двох проводів для зв'язку інтегральних схем всередині електронних приладів).

2.1.2. *LCD*-дисплей

Рідкокристалічний дисплей – прилад для відображення різного роду символів, інформації. Дисплей – це прилад, що доволі часто використовується для реалізації проектів різного роду та рівня на базі *Arduino*. Цифри 1602 відображають те, що даний дисплей може виводити максимум 16 символів, які відображаються у 2 ряди, тобто у кожному із 2 рядків, що виводяться, може бути максимум 8 символів. Але найголовнішим недоліком цього екрану є той факт, що дисплей має 16 цифрових виходів, з яких обов'язковими є мінімум 6.

Кожен з цих виходів виконує певну свою роль. Нумерація відображена на рисунку 2.2.

Опис кожного виходу *LCD*-дисплею:

1. Заземлення *GND*;
2. Живлення (*VCC*) 5V;
3. Установка контрастності дисплею;
4. Команда, дані;
5. Запис і зчитування даних;
6. Увімкнення (*Enable*);
- 7-14. Лінії даних;

15. Плюс підсвітки;
16. Мінус підсвітки.

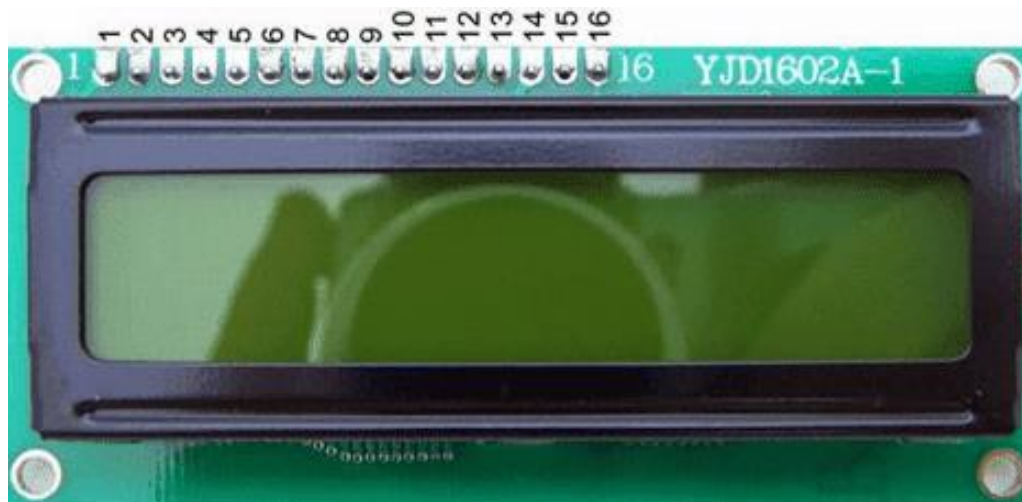


Рис. 2.2. Нумерація виходів дисплею

Характеристики дисплею:

- Світлодіодна підсвітка;
- Контролер *HD44780*;
- Напруга живлення *5V*;
- Формат *16×2* символів;
- Діапазон робочих температур: від *-20 °C* до *+ 70 °C*;
- Кут огляду *180* градусів.

Підключення дисплею напряму до плати відбувається за схемою вказаною на рисунку 2.3. Проте для економії часу та виходів на платі, переважна більшість людей підключають дисплей за допомогою *I2C*-перехідника (рис.2.4.).

Перш ніж підключати дисплей за допомогою *I2C*-перехідника, необхідно з'ясувати, що собою являє *I2C* протокол.

I2C / IIC (Inter-Integrated Circuit) - це протокол, перш за все створений для зв'язку інтегральних схем всередині електронного пристрою. Розробка належить *Philips*. В основі протоколу *I2C* лежить використання 8-бітової шини, яка потрібна для підключення блоків в електроніці управління, та адресної системи, завдяки якій є можливість зв'язку через один і той же провід з декількома пристроями. Найпростіша схема *I2C* може вмістити один головний пристрій (мікроконтролер

Arduino) і декілька підлеглих (наприклад, дисплей). Адреса кожного пристрою знаходиться в діапазоні від 7 до 127, зауважимо, що у кожного пристрою адреса повинна бути унікальною. Використання *I2C*-перехідника зможе скоротити кількість проводів до 4 , а число зайнятих виходів на платі *Arduino* до 2.

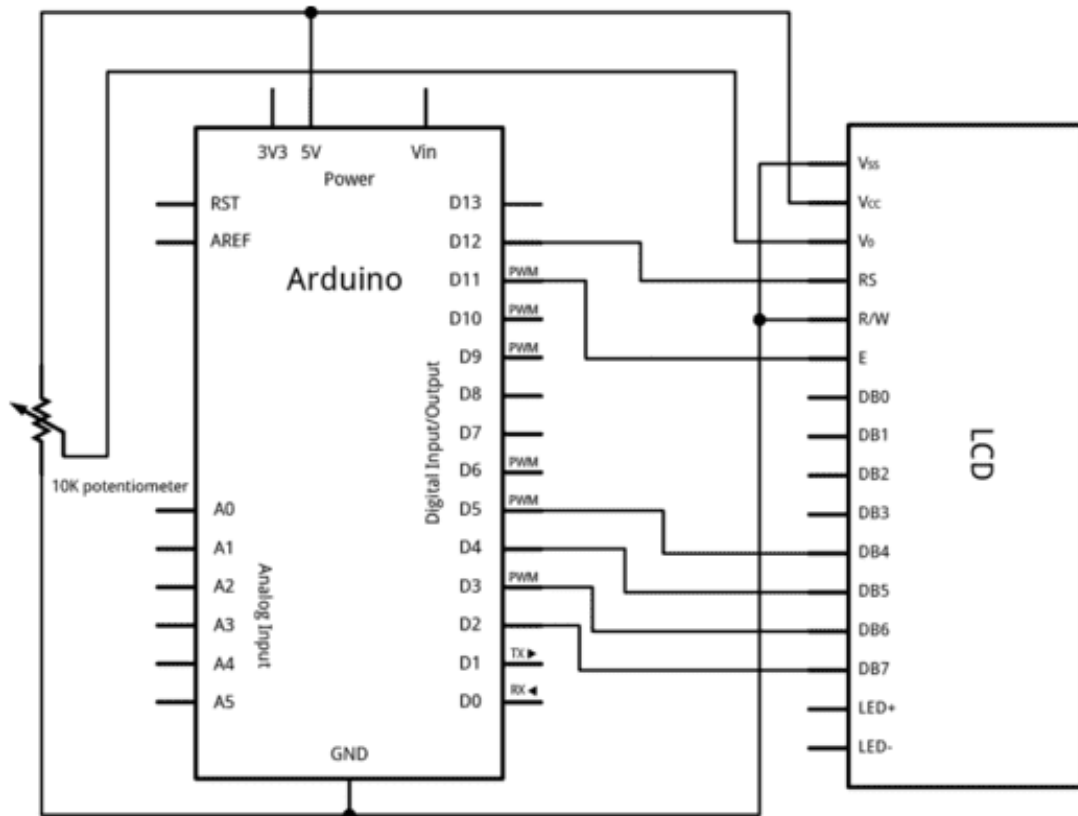


Рис. 2.3. Підключення дисплею напряму, без *I2C*-перехідника



Рис. 2.4. *I2C*-перехідника

З одного боку модуля ми бачимо виходи *i2c*:

- Вивід *GND*, підключається до виходу *GND* на платі;
- Живлення (*VCC*), підключається до виходу *5V*;
- *SCL* підключається до виходу *A5*;
- *SDA* підключається до виходу *A4*.

З іншого боку перехідника ми бачимо зовнішні роз'єми живлення. Також на платі багато ніжок, за допомогою яких модуль приєднується безпосередньо до дисплея. Схему підключення адаптера до дисплея можемо побачити на рисунку 2.5.

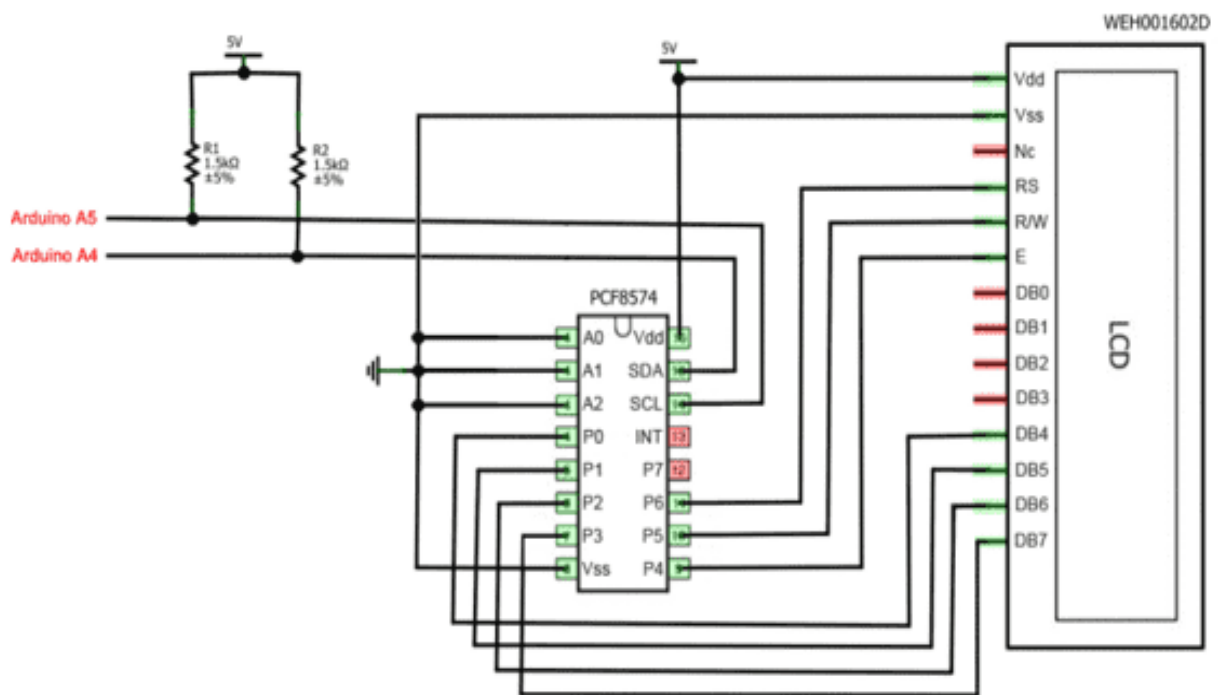


Рис. 2.5. Схеми підключення адаптера *I2C* до дисплея

Після підключення адаптера *I2C* до дисплея, можемо виконувати підключення, цієї системи безпосередньо до плати *Arduino Mega*.

2.1.3. Датчик температури та вологості повітря *DHT22*

Датчик вологості і температури *DHT22* доволі простий у використанні та підключенні до плати. Досить проста також і будова датчика: для вимірювання

вологості в середині датчика вбудовано гігрометр та термістор для вимірювання температури (рис.2.6).

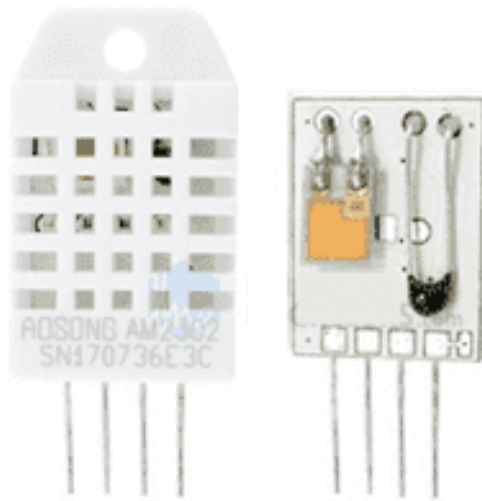


Рис. 2.6. Будова датчика *DHT22*

Будова гігрометра являє собою верхній електрод, вологостримуюча підкладка, нижній електрод та склянна підкладка (рис.2.7). По мірі поглинання водяної пари підкладка звільняє йони, що в свою чергу змінюють провідність між електродами. Зміна опору між двома електронами пропорційно рівню вологості.

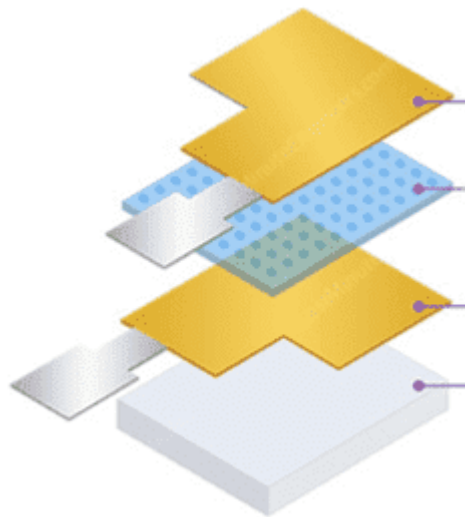


Рис. 2.7. Будова гігрометра на датчику *DHT22*

Як було вже сказано, температура вимірюється за допомогою термістора, який знаходиться на датчику *DHT22* (рис.2.6). Термістор (*NTC*) – це резистор, який змінює опір в залежності від температури. *NTC*, «*Negative Temperature*

Coefficient» (від'ємний температурний коефіцієнт) – означає, що зі збільшенням температури опір термістора зменшується.



Рис. 2.8. Зображення *NTC* термістора

Характеристики датчика *DHT22*:

- Робоча напруга: від 3 до 5 *V*;
- Максимальна робоча напруга: 2,5 *mA*;
- Діапазон вимірювання вологості: 0-100% (похибка 2-5%);
- Діапазон вимірювання температури: -40 – 80 °C (похибка 0,5 °C);
- Частота вибірки: 0,5 *Гц*;
- Розміри корпусу: 15,1 *мм* × 25 *мм* × 7,7 *мм*.
- Підключення датчика:
- Датчик *DHT22* має на виході 4 піни:
- Вихід *VCC* – підключається до виходу *Arduino 5V*;
- Вихід *Data* – використовується для передачі даних з датчика на плату, вихід підключається до цифрового виходу на платі;
- Вихід *NC* – не використовується;
- Вихід *GND* – заземлення, підключається до виходу *GND* на платі.

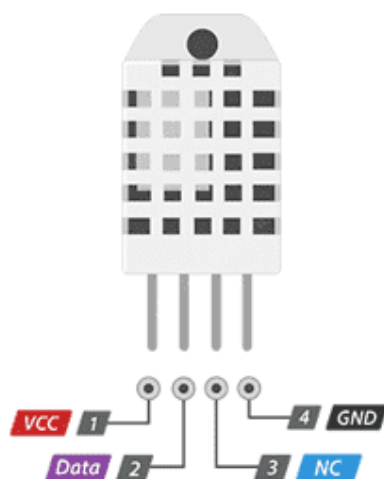


Рис. 2.9. Виходи датчика *DHT22*

2.1.4. Ємнісний датчик вологості ґрунту *V1.2*

Даний датчик, вимірює рівень вологості повітря ємнісним методом, а не резисторним, як інші датчики, це дозволяє продовжити термін роботи датчики, так як він менше піддається корозії.



Рис. 2.10. Датчик вимірювання вологості ґрунту *V1.2*

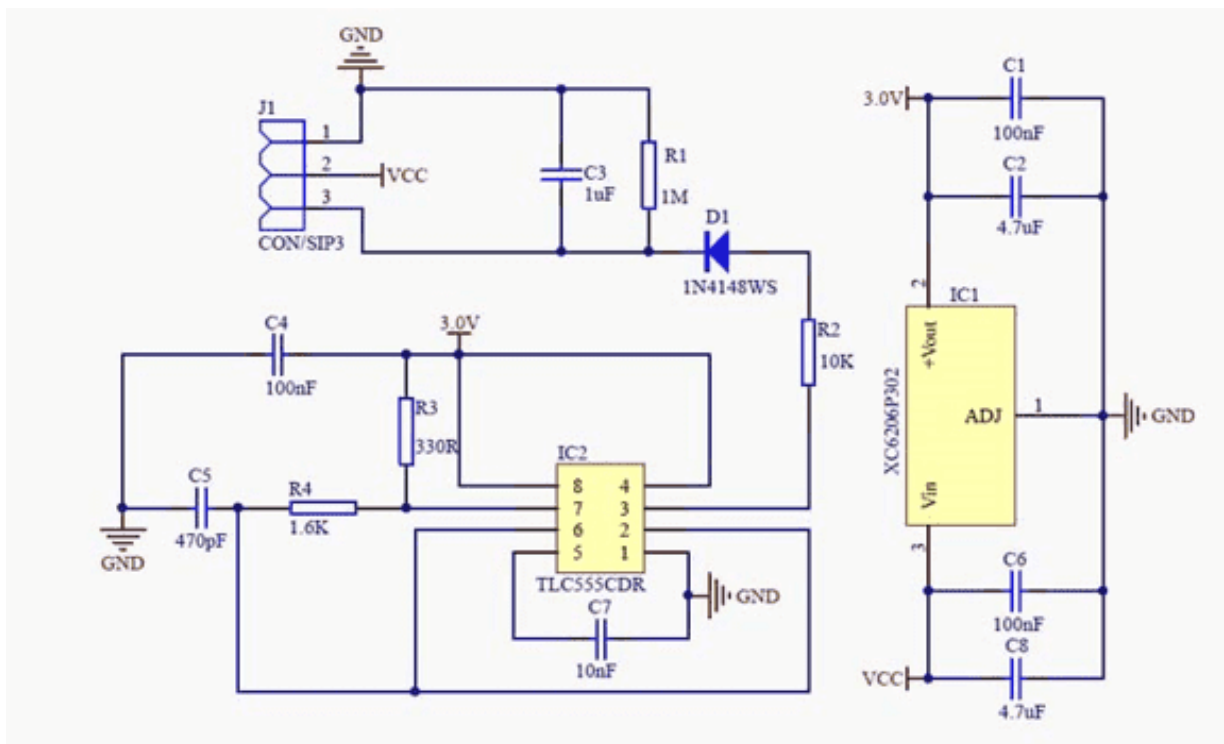


Рис. 2.11. Принципова схема датчика

На даній принциповій схемі можна побачити генератор з фіксованою частотою, побудований на мікросхемі таймера *NE555*. Генератор подає прямокутний сигнал на датчик, який є по суті конденсатором. Для прямокутного сигналу даний конденсатор має певний реактивний опір, чим більша вологість ґрунту, тим більша ємність датчика. Отже існує менший реактивний опір для прямокутного сигналу, що знижує напругу на сигнальній лінії.

Технічні параметри датчика:

- Робоча напруга: 5mA
- Напруга живлення: $3.3 - 5.5\text{V}$
- Розмір: $99 \times 16 \times 10\text{ мм}$
- Вага: 1г
- Вихідна напруга: $0 - 3\text{V}$

Схема підключення даного датчика не дуже складна. Датчик має три виходи:

- *VCC* (живлення) – підключається до виходу 5V на платі;

- *AOUT* (канал передачі даних) – підключається до аналогового виходу на платі;
- *GND* (заземлення) – підключається до виходу *GND* на платі.

2.1.5. Датчик температури *DS18B20*

DS18B20 - цифровий датчик температури з багатьма корисними функціями. Насправді, *DS18B20* - це цілий мікроконтролер, який може зберігати значення вимірювання, може змінювати точність вимірювання, як взаємодіяти з контролером та багато іншого більше.

Особливості даного датчика:

- Похибка датчика складає 0.5 °C у діапазоні температур від -10 до 85 °C;
- Температура вимірювання: від -55 до 125 °C;
- Напруга живлення від 3.3 до 5 V;
- Інформація передається по протоколу *1-Wire*;
- Кожен прилад має свій унікальний номер.

Пам'ять датчика складається із двох частин: *SRAM* (оперативної) та *EEPROM* (енергозалежної). Під час запуску датчик перебуває в стані спокою.

Для початку вимірювання контролер *Arduino* виконує команду «перетворення температури». Отриманий результат буде збережений у 2 байтах реєстру температур, після чого датчик повернеться до початкового стану простою. Якщо ланцюг підключений у зовнішньому режимі живлення, мікроконтролер регулює стан перетворення. Під час виконання команди рядок знаходиться в низькому стані, після закінчення програми рядок переходить у високий стан. Цей метод неприпустимий, коли живиться від паразитної ємності, оскільки високий рівень сигналу повинен бути постійним на шині.

Обмін інформацією по шині *1-Wire* здійснюється для наступних операцій:
Ініціалізація: Визначення послідовності сигналів, з якої починаються вимірювання та інші операції. Ведучий посилає імпульс скидання, після чого

датчик повинен подавати імпульс присутності, щоб вказати, що він готовий виконати операцію.

Запис даних: байт даних передається на датчик.

Зчитування даних: датчик отримує один байт.

Вихід *VCC* підключається до цифрового виходу на платі, вихід *Data* підключається до цифрового виходу на платі та вихід *GND* до відповідного виходу на платі.

Алгоритм отримання інформації про температуру з датчика складається з наступних етапів:

- Визначення адреси датчика та перевірка його з'єднання.
- На датчик надсилається команда з проханням зчитати температуру і ввести вимірне значення в реєстр.
- Видається команда зчитування інформації з реєстру та відправлення отриманого значення на "монітор порту".
- За необхідності дані перетворюються на Цельсій / Фаренгейт.

2.1.6. Фоторезистор

Фоторезистор – це напівпровідниковий прилад (датчик), який змінює свій внутрішній опір під впливом світла.



Рис. 2.12. Зображення фоторезистора

Кожен фоторезистор складається з дуже товстої керамічної підкладки, покритій шаром напівпровідникового матеріалу. На цей напівпровідник змійкою наноситься тонкий шар золота, платини або іншого антикорозійного металу. Шари наносять напиленням.

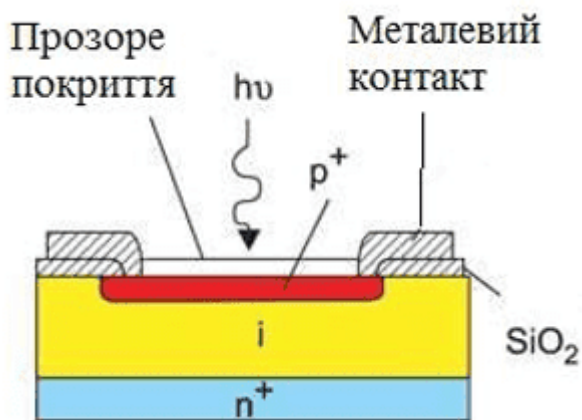


Рис. 2.13. Структурна будова фоторезистора

Фоторезистори використовуються в робототехніці як датчики світла. Вбудований у пристрій датчик дозволяє йому визначати ступінь освітленості, визначати білі або чорні ділянки на поверхні. Одиниці вимірювання ступеня освітлення – люкс, а фоторезистор відображає дані в одиницях вимірювання опору – омах. Для переведення опору в одиниці вимірювання ступеня освітлення використовують залежність (Рис.2.14).

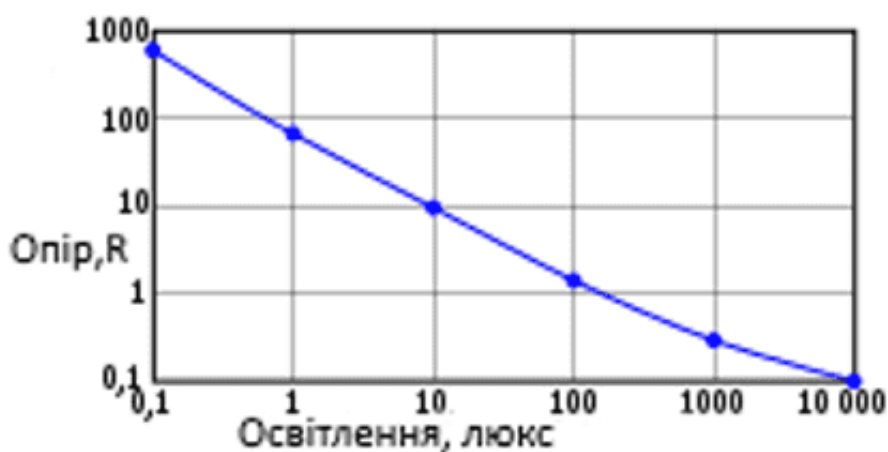


Рис. 2.14. Залежність опору від ступеня освітлення

2.1.7. П'єзоелемент

П'єзокерамічні передавачі (п'єзоелектричні випромінювачі) – це електроакустичні пристрої відтворення звуку, що використовують зворотній п'єзоелектричний ефект, тобто появу механічних деформацій під дією електричного поля.

Існує два типи п'єзоелектричних передавачів: із вбудованим генератором та без нього. П'єзоелектричні передавачі з вбудованим генератором видають постійний тон відразу після подачі номінальної напруги. Зазвичай вони використовуються для простих звукових сповіщень.

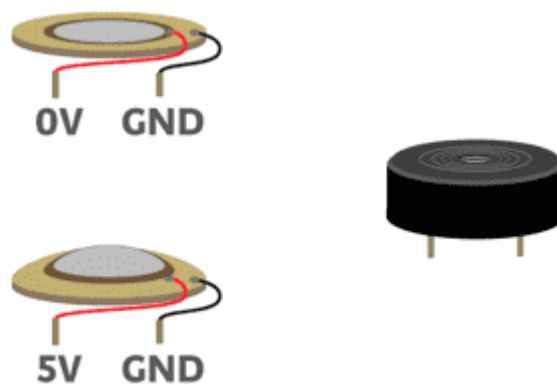


Рис. 2.15. Зображення п'єзоелемента та принцип його роботи

Порівняно з традиційними електромагнітними перетворювачами звуку, п'єзоелектричні перетворювачі мають просту конструкцію. П'єзокерамічний випромінювач складається з металевої пластини, на яку нанесена п'єзоелектрична кераміка, яка має електропровідне розпилення. Пластина і напилення є контактами п'єзоелектричного випромінювача, тоді як пристрій має полярність: плюс і мінус.

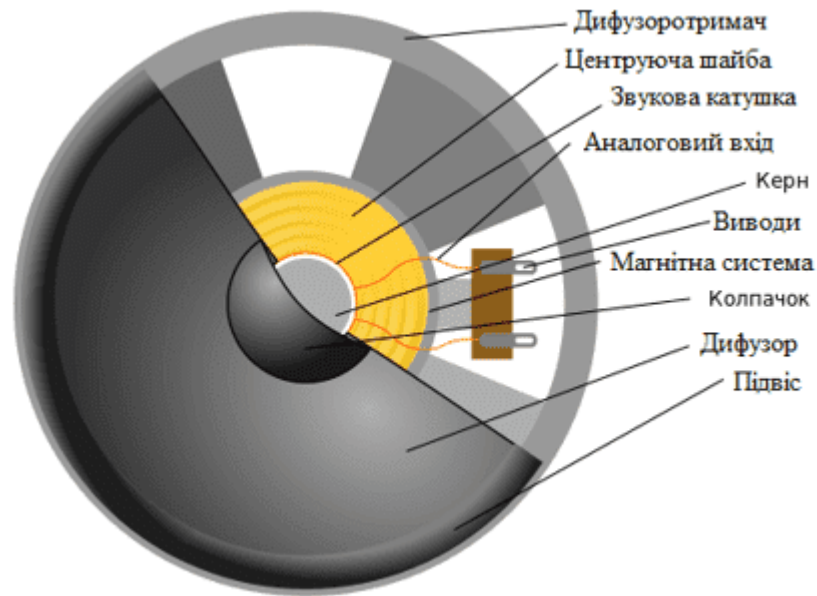


Рис. 2.16. Будова п'єзоелемента

2.1.8. Світлодіод

Світлодіоди - це напівпровідникові елементи, які використовуються для відображення та освітлення. Вони мають полярність (+ і -) і відчують напрямок постійного струму. Якщо світлодіод підключено неправильно, постійний струм не буде протікати, і пристрій не ввімкнеться. Анод (довга ніжка діода) підключений до плюса, відповідно катод (коротка ніжка діода) підключена до мінуса.

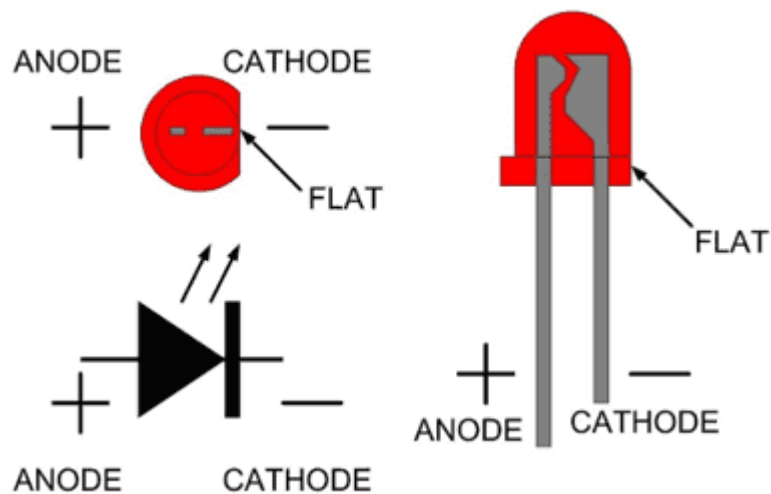


Рис. 2.17. Полярність діода

2.2. Програмні засоби

2.2.1. *Arduino IDE*

Arduino IDE – інтегральне середовище розробки, для розробки, тестування, компіляції коду для плат на базі *Arduino* та завантаження коду на плату. Дана програма має дуже просту структуру, що дозволяє спростити процес розроблення коду та дозволяє займатися цим людям без попередніх навичок.

Після завантаження програми на головній панелі знаходяться 4 функціональні області програми:

1. Меню;
2. Панель швидкого доступу до найважливіших функцій;
3. Редактор тексту;
4. Панель повідомлень;

Меню «Файл» та «Правка» мають стандартний набір функцій;

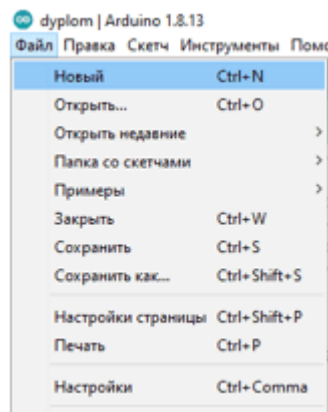


Рис. 2.18. Меню «Файл»

Меню «Скетч» містить функції для компіляції коду та підключення додаткових бібліотек.

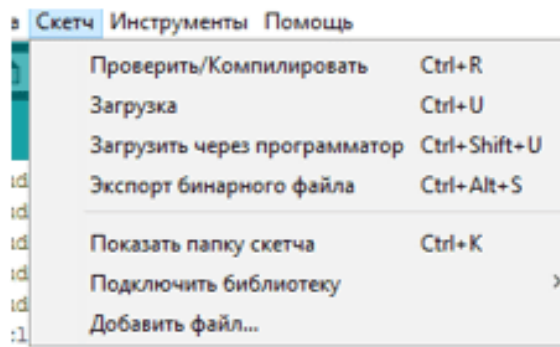


Рис. 2.19. Меню «Скетч»

Меню «Инструменты» містить функції автоматичного форматування коду, відкривання монітору порта, вибору плати, яку програмують.

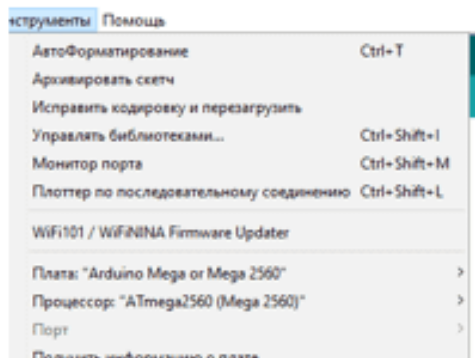


Рис. 2.20. Меню «Инструменты»

Панель швидкого доступу містить кнопки для таких функцій:

1. Компіляція програми;
2. Завантаження коду на плату;
3. Новий проект;
4. Відкрити проект;
5. Записати проект на диск;
6. Відкрити монітор порта;



Рис. 2.21. Панель швидкого доступу

2.2.2. Мова програмування C++

C++ - об'єктно-орієнтована мова програмування загального програмування. Він має імперативні, об'єктно-орієнтовані та загальні функції програмування. C++ працює на багатьох операційних системах: *Windows*, *Linux*, *Mac* та інші.

C++ - це ефективна та потужна мова, яка широко використовується на різноманітних графічних платформах, платформах 3D-моделювання та моделювання в реальному часі. Завдяки великій кількості бібліотек з багатьма функціями, робота на C++ стає простішою та зручнішою, ніж на C. Оскільки об'єктно-орієнтоване програмування забезпечує підтримку успадкування, поліморфізму, інкапсуляції тощо. На відміну від C, C++ дозволяє обробляти винятки та виконувати перезавантаження функцій. Мова програмування C++ використовується для розробки ігор, системних додатків, браузерів та інших проектів. В *Arduino* C++ використовується для написання бібліотек для полегшення роботи з датчиками та іншими пристроями, що підключається до плати та скетчів.

Особливості C++:

- Краще керування пам'яттю – можливість динамічного керування пам'яттю за допомогою операторів *new* і *delete*;
- Об'єктно-орієнтований - забезпечує підтримку абстракція, успадкування, поліморфізму, інкапсуляції, ці функції спрощують написання коду;
- Переносимість – більшість компіляторів підтримують стандарти *ANSI*, це дає можливість використовувати однаковий код на різних платформах: *Windows*, *Linux*, *Mac* та інші;
- Обробка виключень – функція спрощує обробку та ідентифікацію виключень.

2.3. Висновки до розділу

В даному розділі було розглянуто програмні та технічні засоби, які використовувались для створення аналізатора параметрів повітря та ґрунту. До технічних засобів належать: дисплей, датчик вологості та температури *DHT22*, датчик температури *ds18b20*, ємнісний датчик вологості ґрунту, фоторезистор, п'єзоелемент, світлодіод та безпосередньо мікроконтролер *Arduino Mega2560*, до програмних засобів належать середа розроблення *Arduino IDE* та мова програмування *C++*. Під час написання даного розділу було вивчено будову перерахованих елементів та їх основні функції.

РОЗДІЛ 3

АЛГОРИТМІЧНА РОЗРОБКА АНАЛІЗАТОРА ПАРАМЕТРІВ ПОВІТРЯ ТА ҐРУНТУ

Діаграма *UML* - це діаграма, заснована на *UML* (уніфікованому мовою моделювання) для візуального представлення системи разом з її основними учасниками, ролями, діями, артефактами або класами, щоб краще розуміти, змінювати, підтримувати або додавати інформацію через систему.

UML-діаграми в основному використовувалися як універсальна мова моделювання в області розробки програмного забезпечення. Однак з тих пір вони знайшли своє застосування в документуванні різних бізнес-процесів або робочих процесів. Наприклад, діаграми активності, можна використовувати для заміни блок-схем. Вони надають стандартизований метод моделювання робочого процесу і більш широкий набір функцій для підвищення зручності читання та ефективності.

3.1. Діаграма прецедентів

В *UML* діаграма прецедентів може підсумовувати інформацію про користувачів вашої системи (також відомих як теми) і їх взаємодіях з системою. Використовуйте різні спеціальні символи і з'єднувачі для їх створення. Графік успішного використання може допомогти команді обговорити і просувати:

- Сценарії, в яких ваша система або додаток взаємодіє з людьми, організаціями або зовнішніми системами;
- Цілі, які ваша система або програма допомагає їм (гравцям) досягти;
- Обсяг вашої системи.

Кафедра КСУ				НАУ 21 12 15 000 ПЗ			
<i>Виконав</i>	Олексенко М.С.			Алгоритмічна розробка аналізатора параметрів повітря та ґрунту.	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	Коба О.В.				Д		32 61
<i>Консульт.</i>					СП-436Б 123		
<i>Норм. контр.</i>	Тупота Є.В.						
<i>Зав. Каф.</i>	Литвиненко О.Є.						

Діаграма прецедентів не зможе змоделювати порядок виконання кроків як діаграма прецедентів. Натомість правильна діаграма прецедентів показує загальний огляд відносин між варіантами використання, учасниками і системами. Експерти рекомендують використовувати діаграми варіантів використання в якості доповнення до більш значущих варіантів використання в текстовій формі. Діаграми варіантів використання *UML* ідеально підходять для: відображення цілей взаємодії системи з користувачем, визначити і систематизувати функціональні вимоги в системі, Вказування контексту і вимог до системи.

У даній системі є три сутності, які виконують певні функції: комплекс датчиків (вимірювання температури повітря, вимірювання вологості повітря, вимірювання температури ґрунту, вимірювання вологості ґрунту, вимірювання освітлення), система (зчитування даних з датчиків, обробка зчитаних даних, вивід текстової інформації на дисплей, звуковий сигнал), користувач (увімкнення приладу в джерело живлення, аналіз отриманої інформації, встановлення датчиків у відповідне положення).

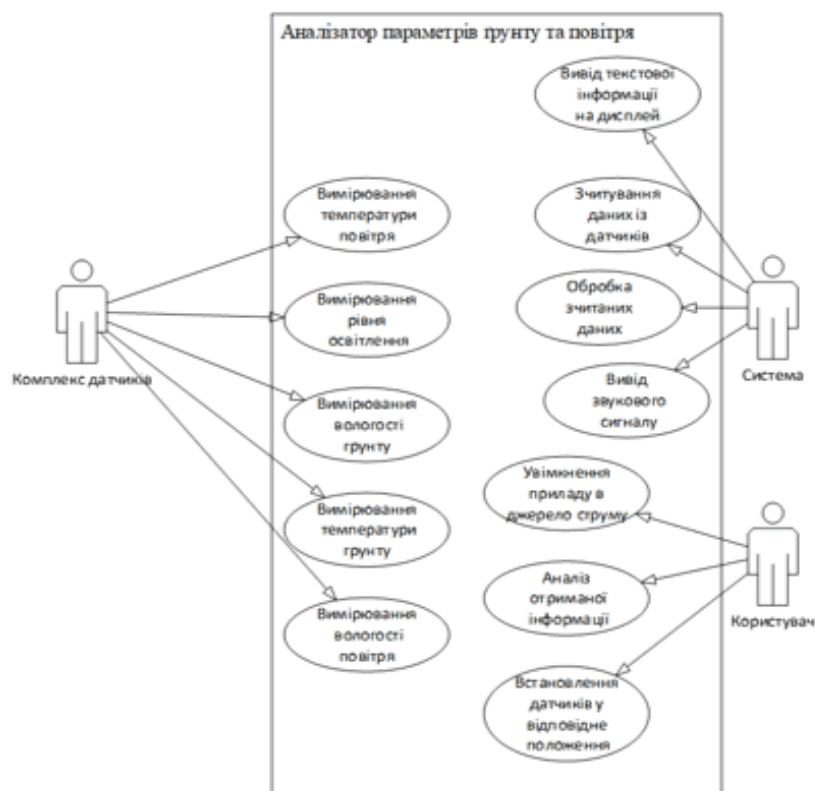


Рис. 3.1. Діаграма прецедентів

3.2. Діаграма послідовності

Діаграма послідовності - це тип діаграми взаємодії, оскільки вона описує, як і в якому порядку група об'єктів працює разом. Розробники програмного забезпечення та бізнес-професіонали використовують ці діаграми для розуміння вимог нової системи або для документування існуючого процесу. Діаграми послідовностей іноді називають діаграмами подій або сценаріями подій.

У даній системі діаграма послідовності являє собою користувача, який виконує два кроки: підготовка датчиків, увімкнення пристрою та систему, яка виконує теж два кроки: зчитування інформації з датчиків, вивід інформації.

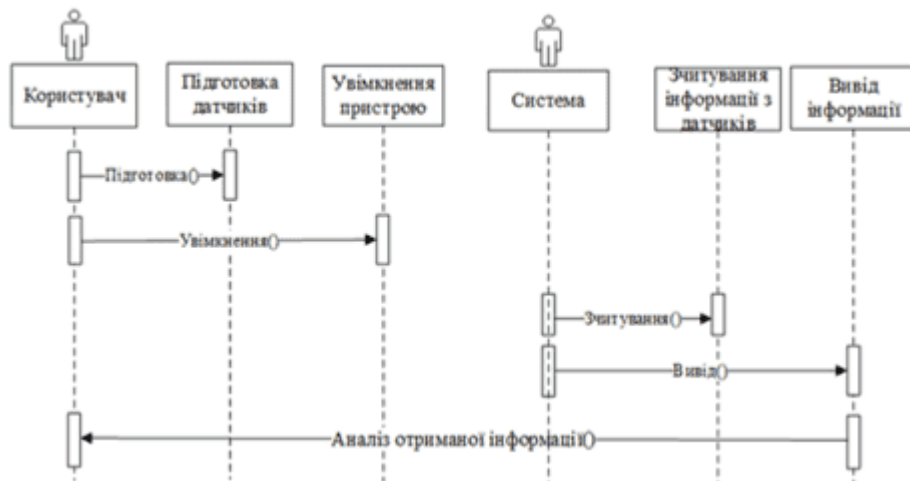


Рис. 3.2. Діаграма послідовності

3.3. Діаграма класів

При розробці програмного забезпечення діаграма класів в *UML* це відображення безперервної структури, яка описує структуру системи, показуючи категорії системи, їх властивості, функції (або методи) і взаємозв'язок об'єктів.

У даній системі діаграма класів відображає не типові класи, які використовуються в ООП, а класи як частини системи, які відповідають за певні частини процесу і виконують їх.



Рис. 3.3. Діаграма класів

3.4. Діаграма станів

Діаграма стану - це діаграма *UML*, які використовуються для формування динаміки системи. Вони визначають різний стан об'єкта протягом його життя, і ці стани змінюються подіями, станами, в яких існує об'єкт і які змінюються при зміні подій. Основна мета діаграми станів - визначити тривалість життя об'єктів від створення до кінця. Дана система описується 5 станами: налаштування датчиків, увімкнення приладів, зчитування даних, перевірка даних, аналіз отриманих даних.

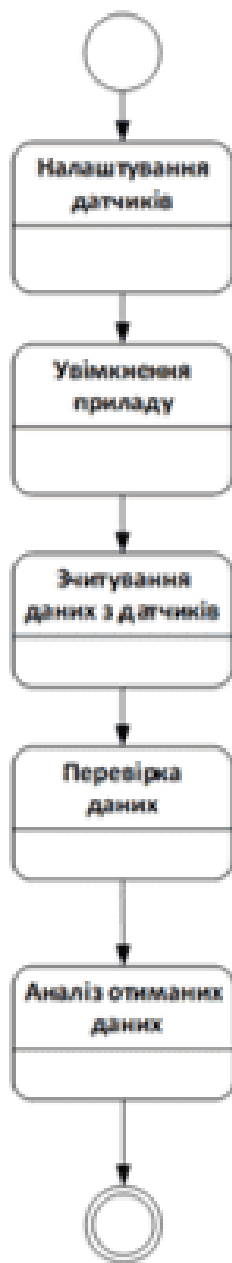


Рис. 3.4. Діаграма станів

3.5. Діаграма кооперації

Діаграми кооперації використовуються, щоб показати, як об'єкти взаємодіють для виконання поведінки конкретного варіанту використання або частини варіанту використання. На відміну від діаграми послідовності, діаграми кооперації показує відносини між об'єктами.

У даній системі в діаграмі кооперації зображено три сутності: користувач, набір датчиків, система. Ці сутності виконують функції одна відносно одної. Користувач виконує функцію налаштування відносно набору датчиків, також користувач виконує функцію увімкнення відносно системи, в свою чергу система виконує функцію виводу даних відносно користувача та функцію зчитування даних відносно набору датчиків, а набір датчиків виконує функцію перевірки відносно даних. Отже на цій діаграмі ми можемо бачити, що для коректної роботи системи необхідні і важливі всі три сутності.



Рис. 3.5. Діаграма кооперації

3.6. Висновки до розділу

В даному розділі було спроектовано систему та описано основні функції системи за допомогою *UML*-діаграм. Діаграма *UML* - це діаграма, заснована на *UML* (уніфікованому мовою моделювання) для візуального представлення системи разом з її основними учасниками, ролями, функціями та класами. При написанні даного розділу було розроблено такі діаграми: діаграма прецедентів, діаграма станів, діаграма кооперації, діаграма послідовності та діаграма класів, також було описано значення кожної діаграми.

РОЗДІЛ 4

РОЗРОБКА АНАЛІЗАТОРА ПАРАМЕТРІВ ПОВІТРЯ ТА ҐРУНТУ

4.1. Програмні бібліотеки розробки

При розробці були використані такі бібліотеки:

- *Wire.h* – бібліотека для можливості роботи по шині *I2C*;
- *DallasTemperature.h* – бібліотека для роботи з датчиком температури *ds18b20*;
- *DHT.h* – бібліотека для роботи з датчиком вологості і температури температури *DHT22*;
- *LiquidCrystal_I2C.h* – бібліотека для роботи з дисплеєм;
- *SPI.h* – бібліотека для роботи по шині *SPI*;
- *OneWire.h* – бібліотека для роботи з протоколом *1-Wire*.

4.2. Апаратне забезпечення для реалізації пристрою:

При розробці були використані такі пристрої:

- 1x плата *Arduino Mega*;
- 1x датчик вологості і температури *DHT22*;
- 1x ємнісний датчик вологості ґрунту *V1.2*
- 1x *LCD*-дисплей 1602 з інтерфейсною шиною *I2C*;
- 1x датчик температури *ds18b20*;
- 1x динамік *buzzer 5V, 2300 Гц*;
- 1x макетна плата на базі *Arduino*;
- 3x фоторезистори;

Кафедра КСУ				НАУ 21 12 15 000 ПЗ			
Виконав	Олексенко М.С.			Розробка аналізатора параметрів повітря та ґрунту	Літера	Аркуш	Аркушів
Керівник	Коба О.В.				Д	38	61
Консульт.					СП-436Б 123		
Норм. контр.	Тупота Є.В.						
Зав. Каф.	Литвиненко О.Є.						

- 1x корпус для *Arduino Mega*;
- 3x резистори на 220 Ом;
- 4x резистори на 10 КОм;
- 1x резистор на 4,7 КОм;
- 1x корпус для пристрою;
- 1x джерело живлення (12В);
- 1x світлодіод червоного кольору;
- 1x світлодіод жовтого кольору;
- 1x світлодіод зеленого кольору;
- 1x кабель для *Arduino Mega*;
- 20x проводів типу «тато-тато»;
- 30x проводів типу «мама-тато».

4.3. Логіка роботи пристрою

Логіка роботи аналізатора параметрів повітря та ґрунту доволі проста, користувач налаштовує датчики температури і вологості ґрунту в правильне положення, вмикає пристрій у джерело живлення, датчик *DHT22* вимірює температуру і вологість повітря за допомогою функцій *.readTemperature()* та *.readHumidity()*, освітлення вимірюється за допомогою фоторезисторів, температура ґрунту вимірюється за допомогою датчика *ds18b20* та вологість ґрунту вимірюється за допомогою датчика вологості ґрунту, у разі відмінності параметрів вологості ґрунту виводиться повідомлення на екран та звук у динаміку, інші дані виводяться на екран, рівень світла можна також побачити за допомогою світлодіодів, які показують у якому проміжку (від 0% до 100%) знаходиться рівень освітленості.

4.4. Практична реалізація аналізатора параметрів ґрунту та повітря

Перед початком роботи необхідно було помістити плату *Arduino Mega* в корпус, для подальшого уникнення різного роду фізичних пошкоджень. Але перед тим, як помістити плату в корпус, його спочатку необхідно було зібрати, тому що продається він у розібраному стані (рис.4.1.).



Рис. 4.1. Корпус для плати у розібраному стані



Рис. 4.2. *Arduino Mega 2560* в захисному корпусі

4.4.1. Підключення дисплея

Після виконання всіх робіт пов'язаних із забезпечення безпечного користування платою *Arduino Mega 2560* ми можемо приступати до підключення необхідних елементів.

Спочатку було підключено *LCD*-дисплей 1602, для того, щоб виводити результати не на монітор порту, як можна зробити в середовищі *Arduino IDE*.

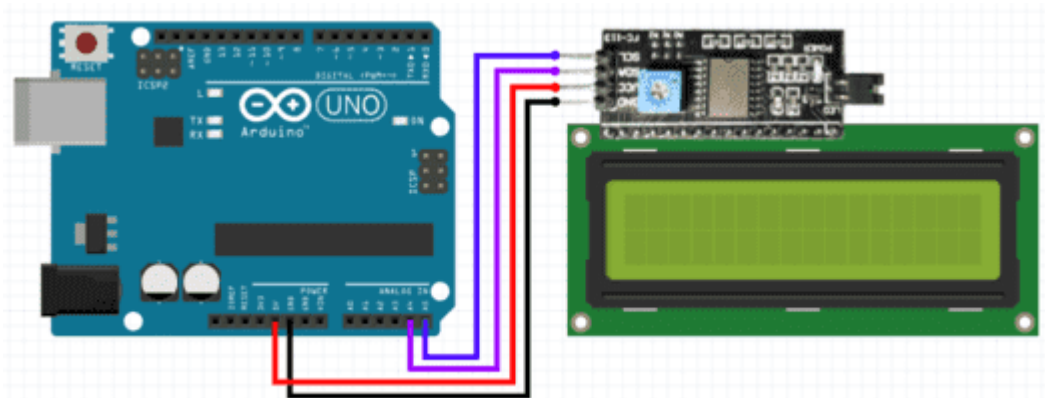


Рис. 4.3. Схема підключення дисплею

Після підключення дисплею, приступаємо до програмного опису даної системи. Для роботи з *LCD*-дисплеєм нам необхідні будуть 2 бібліотеки: *Wire.h* – бібліотека для роботи з шиною *I2C*, та бібліотека безпосередньо для роботи з дисплеєм *LiquidCrystal_I2C.h*.

Після підключення бібліотеки необхідно вказати адресу *I2C*:

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

Після цих дій, можна повноцінно використовувати дисплей, для створення аналізатору.

```
lcd.init();           // функція ввімкнення LCD  
lcd.backlight();     // функція ввімкнення підсвітки  
lcd.clear();         // функція очистки екрану  
lcd.setCursor(1, 0); // функція встановлення курсору в відповідне місце  
lcd.print("Oleksenko!"); // функція виводу даних на екран дисплею
```

Після написання коду завантажуюємо його безпосередньо на плату, перед цим обравши порт і плату в середовищі *Arduino IDE*.



Рис. 4.4. Результат підключення дисплею і завантаження написаного коду на плату

4.4.2. Підключення датчика *DHT22*

Для підключення датчика до плати необхідно використати резистор опором 10 *КОм*. Резистор – це елемент, який чинить опір напруги, яку подає джерело живлення, у нашому випадку напругу подає плата *Arduino*. Резистор необхідно встановити між виходами *VCC* та *Data* на датчику. Підключення датчика здійснюється за даною схемою:

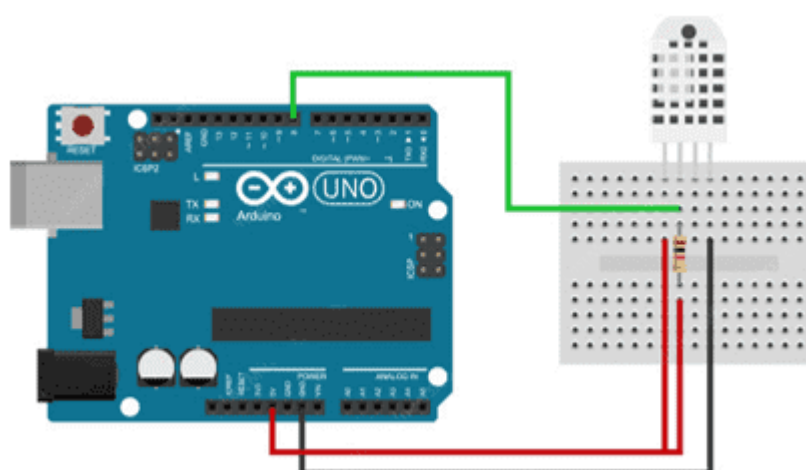


Рис. 4.5. Схема підключення датчика *DHT22*

Практична реалізація підключення даного датчика. Спочатку необхідно підключити проводи до виходів на датчику *DHT22*.

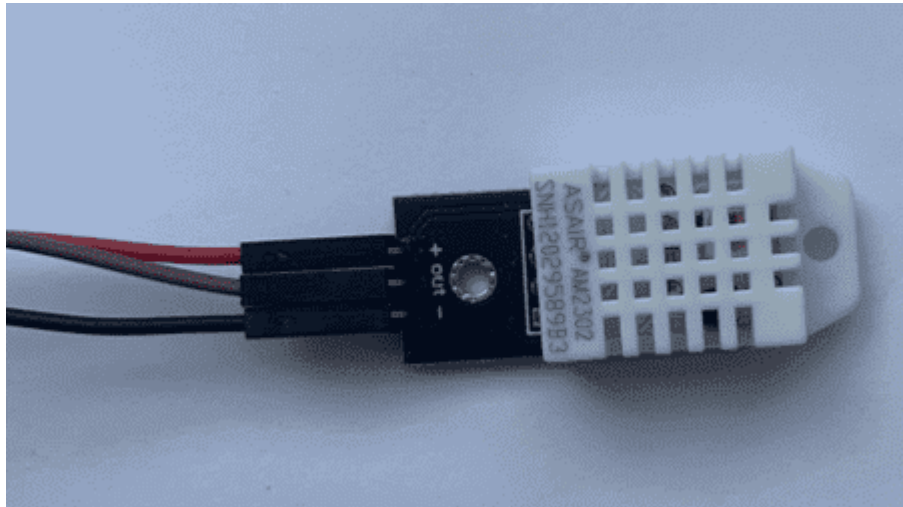


Рис. 4.6. Підключення проводів до датчика

Підключаємо датчики до виходів на безпаячній платі, після підключення необхідно підключити резистор опором 10 КОм між виходами «+» та *out*.

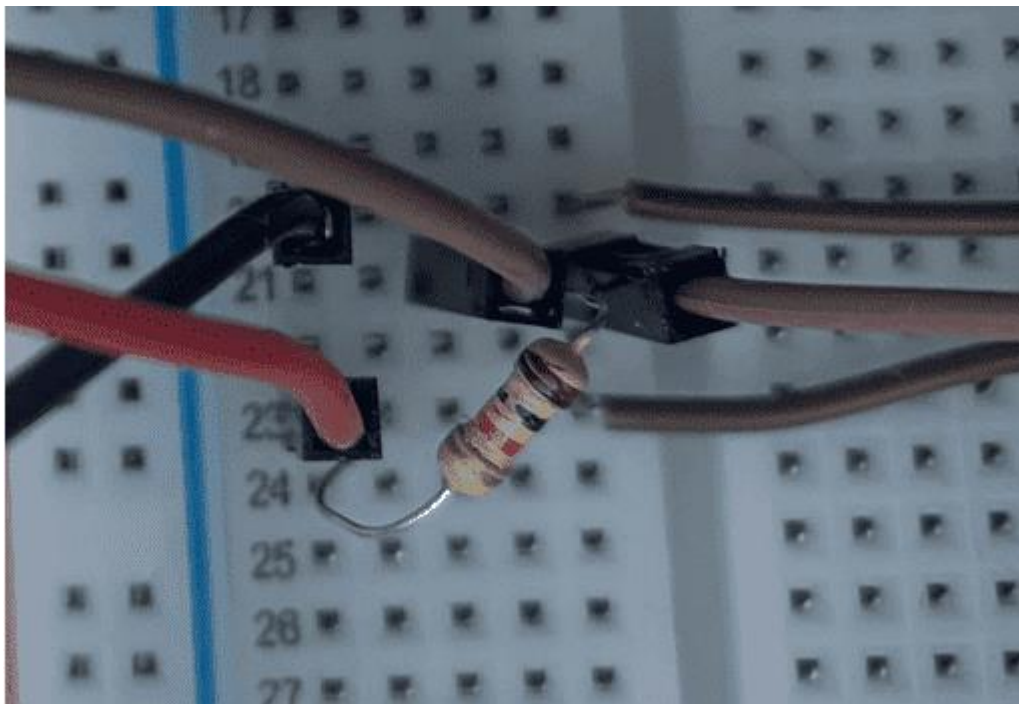


Рис. 4.7. Підключення резистора

Після підключення резистора, необхідно підключити дріт коричневого кольору до цифрового виходу на платі.

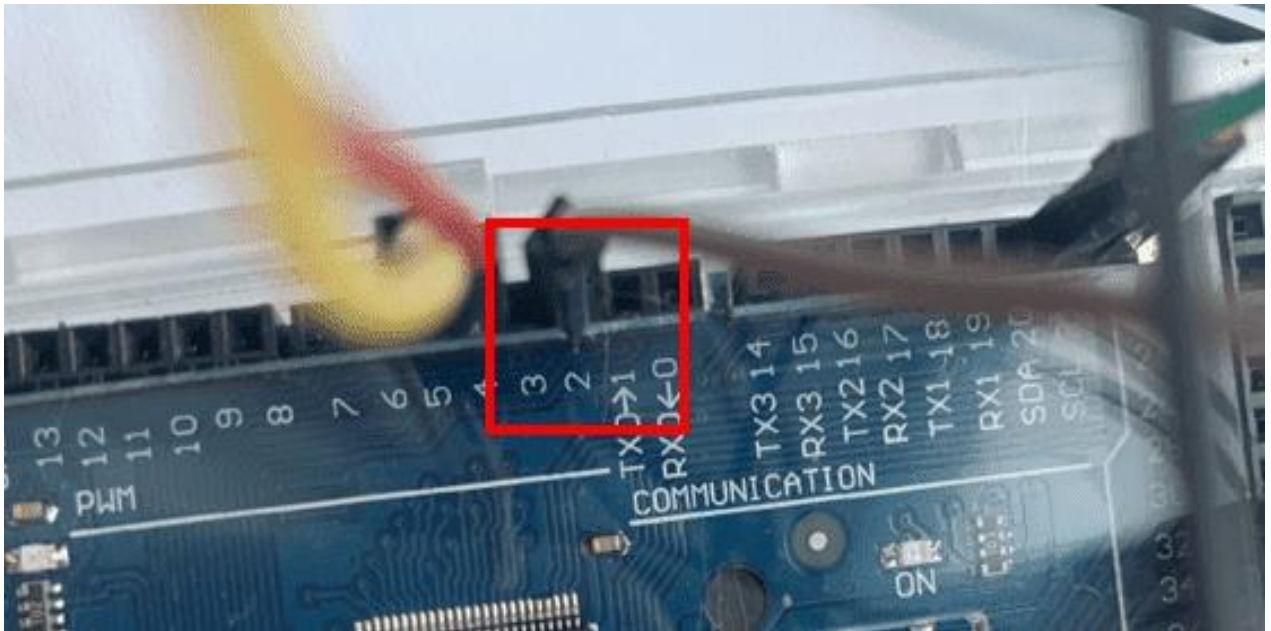


Рис. 4.8. Підключення датчика до цифрового виходу

Після підключення датчика, було написано код для роботи з датчиком. Код для датчика температури вологості і температури повітря *DHT22* досить простий.

Перш за все для роботи з датчиком необхідно буде підключити спеціальну бібліотеку для роботи з даним датчиком.

```
#include "DHT.h"
```

Після підключення бібліотеки необхідно вказати вихід на платі до якого буде підключений вихід *Data* на датчику.

```
#define DHTPIN 2
```

Далі можна писати код для зчитування даних. Зчитування виконується за допомогою функцій *readHumidity()* та *readTemperature()*.

```
float dht_hum = dht.readHumidity();
```

```
float dht_tem = dht.readTemperature();
```

Перевірка чи датчик працює коректно:

```
if (isnan(dht_hum) || isnan(dht_tem)) {
```

```
Serial.println("Ошибка считывания");
```

```
return;}
```

Вивід інформації на дисплей:

```
lcd.setCursor(2, 0);  
lcd.print("hum: ");  
lcd.print(dht_hum);  
lcd.print(" %");  
lcd.setCursor(2, 1);  
lcd.print("tem: ");  
lcd.print(dht_tem);  
lcd.print(" C");
```

Отже, можемо бачити, що робота з даним датчиком не викликає великих труднощів. Результат коректної роботи датчика та правильно написаного коду ми можемо бачити на рисунку.



Рис. 4.9. Результат роботи датчика *DHT22*

4.4.3. Підключення ємнісного датчика вимірювання вологості повітря *VI.2*

Схема підключення датчика виглядає наступним чином:

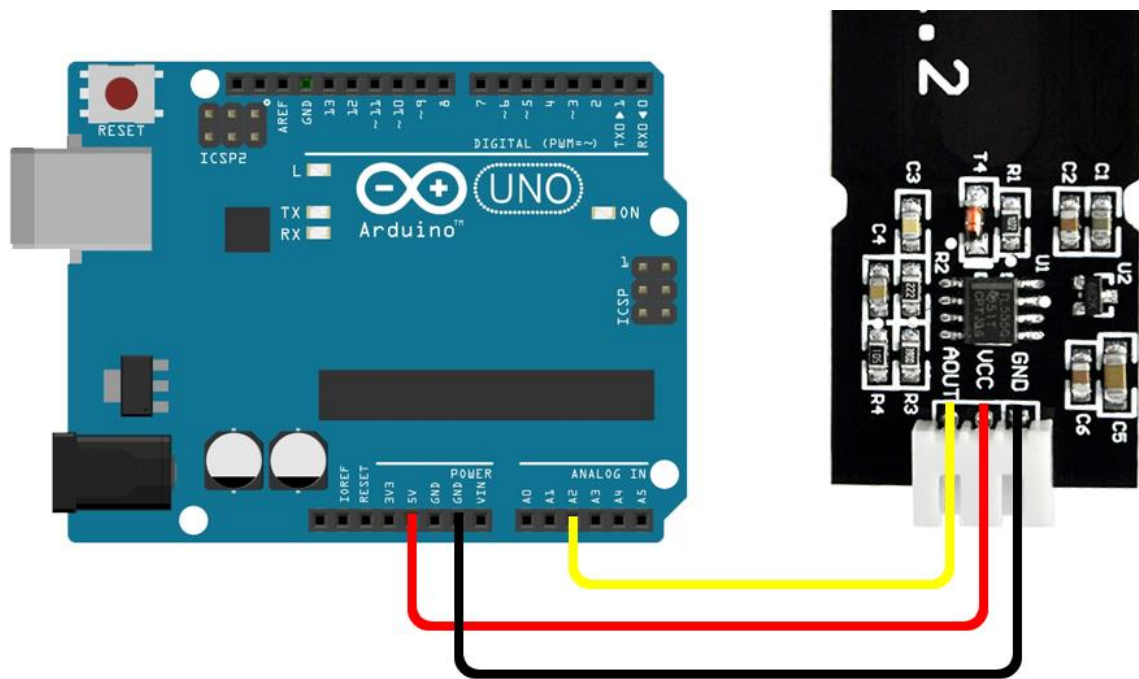


Рис. 4.10. Схема підключення ємнісного датчика вологості ґрунту

Практична реалізація даного датчика виглядає наступним чином. Спочатку необхідна підключити датчик до проводів через спеціальний роз'єм.

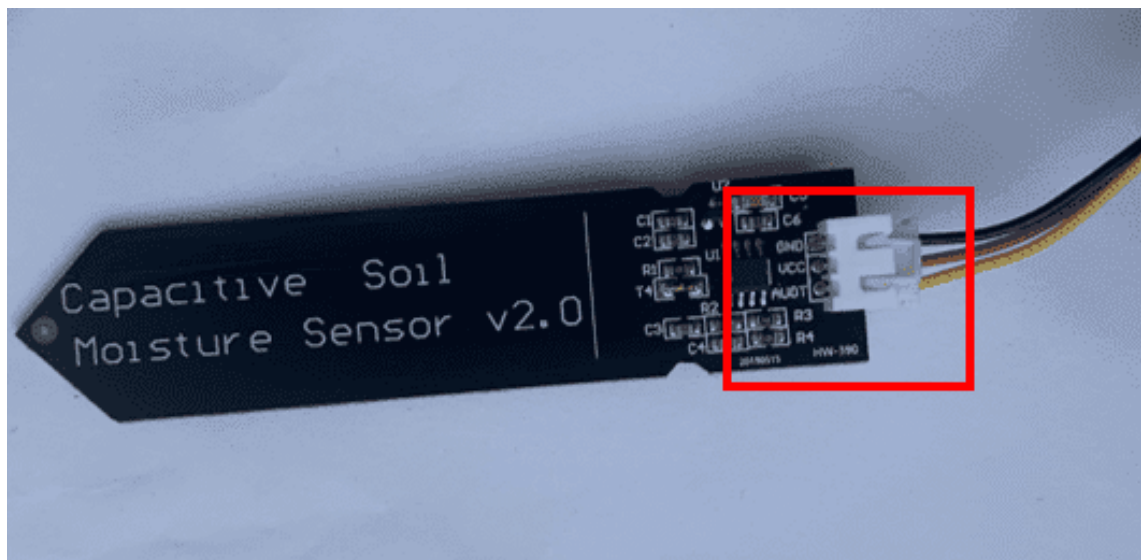


Рис. 4.11. Підключення датчика в роз'єм

Жовтий провід підключаємо на аналоговий вихід на платі, червоний до виходу 5V, чорний до виходу GND.

Після підключення даного датчика необхідно реалізувати його роботу за допомогою програмного коду:

Перш за все необхідно вказати вихід на платі до якого підключений аналоговий вихід:

```
#define SENSOR_GR A0
```

Після чого можна прописати код для зчитування даних з датчика.

```
sensor_gr=analogRead(SENSOR_GR);
```

```
Serial.println("sensor_gr");
```

```
Serial.println(float(sensor_gr),1);
```

Зчитування аналогових даних з різних датчиків виконують за допомогою функції *analogRead()*, і для перевірки виводимо дані на екран монітору, а не на дисплей. Бачимо, що дані виводяться не у відсотках, аналоговий вихід показує напругу, яку пропускає через себе датчик.

Необхідно перевести дані у відсотки за допомогою функції *map()*:

```
sensor_gr_per=float(map(sensor_gr,SENSOR_GR_MIN,SENSOR_GR_MAX,0.0,100.0));
```

Дана функція приймає аргументами 5 значення:

sensor_gr – вимірне значення напруги, яке проходить через датчик;

SENSOR_GR_MIN – максимальне значення напруги, що може пропускати датчик;

SENSOR_GR_MAX – мінімальне значення напруги, що може пропускати датчик;

0 та 100 – значення мінімального та максимального значення вологості ґрунту.

Для обчислення мінімального значення напруги, яке може пропускати датчик необхідно було занурити датчик у склянку з водою, оскільки вологості води 100%. Для вимірювання максимального значення напруги, яке може пропускати датчик необхідно було виміряти значення датчика, коли той був сухий і не занурений у ґрунт чи воду. Після цього необхідно було задати ці значення константами:

```
#define SENSOR_GR_MIN 612.0
```

```
#define SENSOR_GR_MAX 263.0
```



Рис. 4.12. Результат роботи датчика вологості ґрунту

4.4.4. Підключення датчика температури *ds18b20*

Підключення датчика до мікроконтролера *Arduino* відбувається за допомогою трьох дротів, які необхідно буде припаяти до датчика. Також для підключення даного датчика необхідний буде резистор на $4,7\text{ КОм}$, який необхідно розмістити між виходом *VCC* та *Data*.

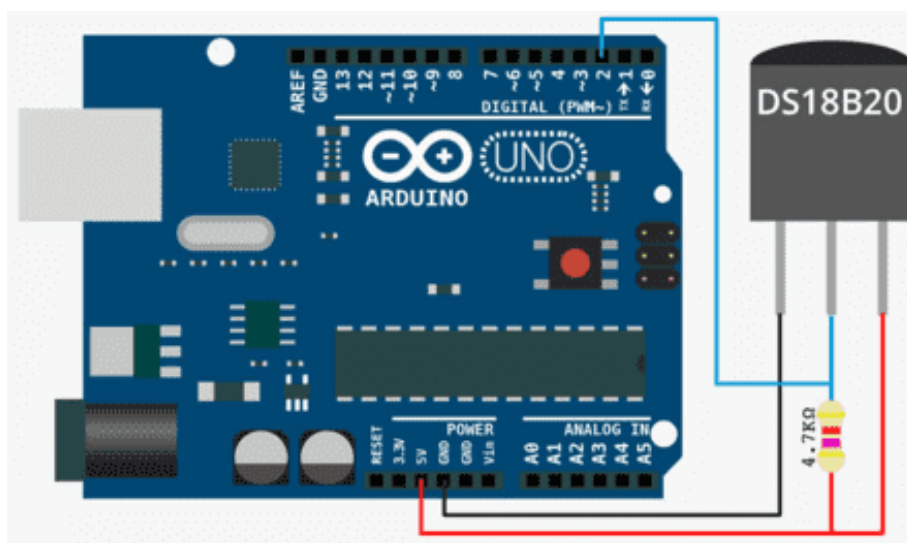


Рис. 4.13. Схема підключення датчика *DS18B20*

Для простішого підключення датчика та економії вхідних портів на безпасчній платі було виконання впаювання резистора між дротами, які ведуть від виходів *VCC* (провід червоного кольору на фото) та *Data* (провід жовтого кольору на фото) на датчику.



Рис. 4.14. Результат впаювання резистора

Після чого необхідно підключити проводи до виходів на мікроконтролері *Arduino*.

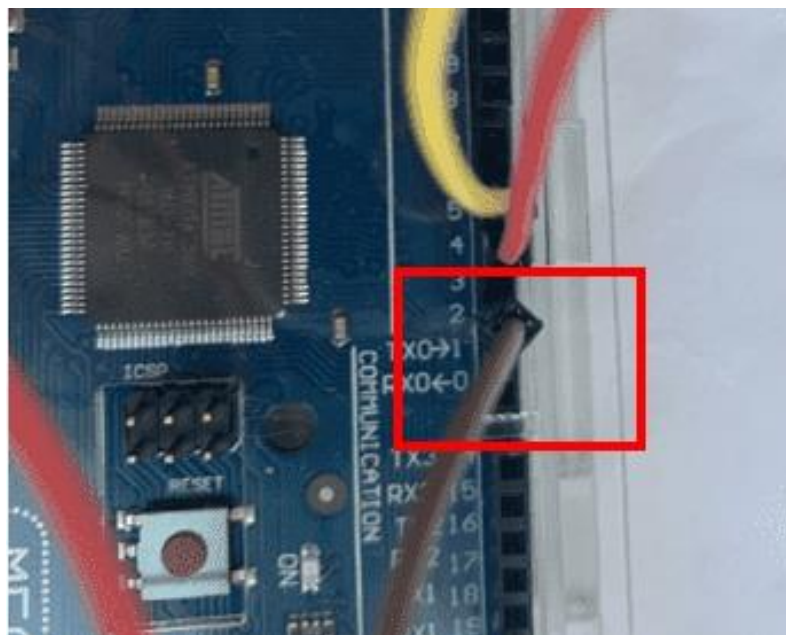


Рис. 4.15. Підключення виходу *Data* на датчику до цифрового виходу на платі

Після підключення датчика необхідно описати код для роботи зчитування даних з датчика. Для початку необхідно підключити дві бібліотеки для полегшення роботи з даним датчиком:

- *OneWire.h* – бібліотека для можливості роботи по шині I2C;
- *DallasTemperature.h* – бібліотека для роботи з датчиком температури *ds18b20*.

Вказати номери виходів на платі *Arduino* до яких підключається датчик. Підключення виходу *VCC* (живлення) датчика *DS18B20* до цифрового виходу зумовлено тим, що якщо подавати живлення на його постійно, то датчик швидко нагріється і буде повертати невірне значення.

```
#define ONE_WIRE_BUS 5 //вихід data на термістори
```

```
#define temp_power 4//вихід vcc на термістори
```

Створити об'єкт типу *OneWire* і вказати те, що об'єкт підключається по шині, в наступній строчці зв'язати об'єкт із об'єктом типу *DallasTemperature*:

```
OneWire oneWire(ONE_WIRE_BUS);
```

```
DallasTemperature t_gr_sensor(&oneWire);
```

Функція для зчитування даних з датчика:

```
float gr_temperature(){
```

```
digitalWrite(temp_power,HIGH); // вмикаємо живлення датчика
```

```
delay(100);//затримка
```

```
t_gr_sensor.requestTemperatures();//зчитування 1 значення
```

```
delay(500);//затримка півсекунди перед повторним визначенням
```

```
t_gr_sensor.requestTemperatures();//зчитування повторного значення
```

```
float t=float(t_gr_sensor.getTempCByIndex(0));//записуємо значення з датчика у змінну
```

```
digitalWrite(temp_power,LOW);//вимикаємо живлення датчика
```

```
delay(5000);
```

```
return (t);//повернення значення
```

```
}
```

Функція зчитує значення 2 рази через те, що після підключення датчика він повертає перше значення хибне, хибне значення буває лише першим після підключення. Після опису функції необхідно увімкнути датчик:

```
t_gr_sensor.begin();
```

Після увімкнення можемо виконати зчитування температури ґрунту та вивести на дисплей зчитані дані:

```
temp_ground=gr_temperature();
```

```
lcd.setCursor(0, 1);
```

```
lcd.print("tem_gr: ");
```

```
lcd.print(temp_ground);
```

```
lcd.print(" C");
```

Після підключення та написання коду можемо виконати завантаження коду на плату та перевірити правильність роботи датчика:



Рис. 4.16. Результат роботи датчика

4.4.5. Підключення фоторезисторів для визначення рівня світла

Підключення фоторезистора відбувається за простою схемою.

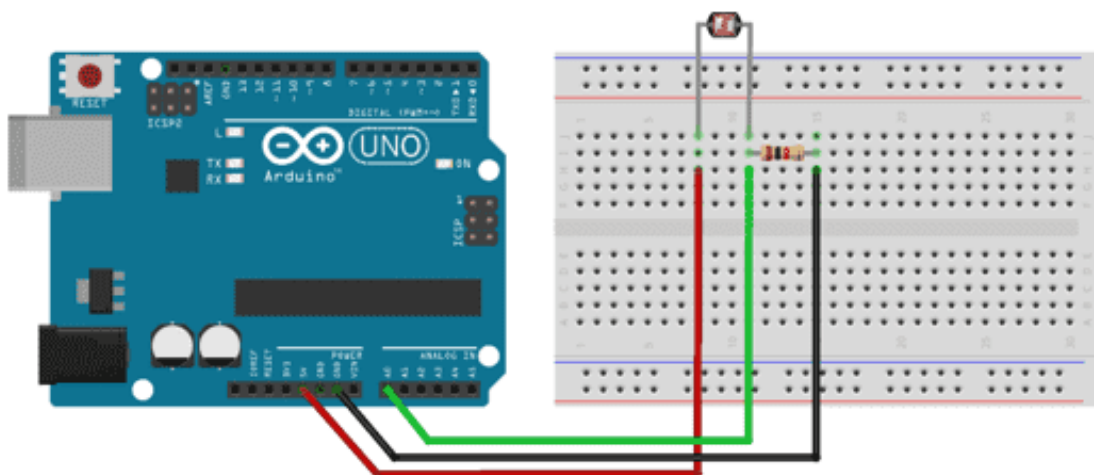


Рис. 4.17. Схема підключення фоторезистора

Як можна бачити зі схеми, для підключення датчика необхідно буде використати резистор опором 10 КОм . Для полегшення використання фоторезистора необхідно впаяти резистор між виходом який веде від виходу GND на платі та виходом фоторезистора, який підключається до аналогового виходу на платі. Інший вихід фоторезистора напряму підключити до виходу $5V$ на платі. Для точності обчислення було використано 3 датчики для вимірювання ступеня освітлення, тому дану процедуру необхідно виконати для всіх трьох датчиків.

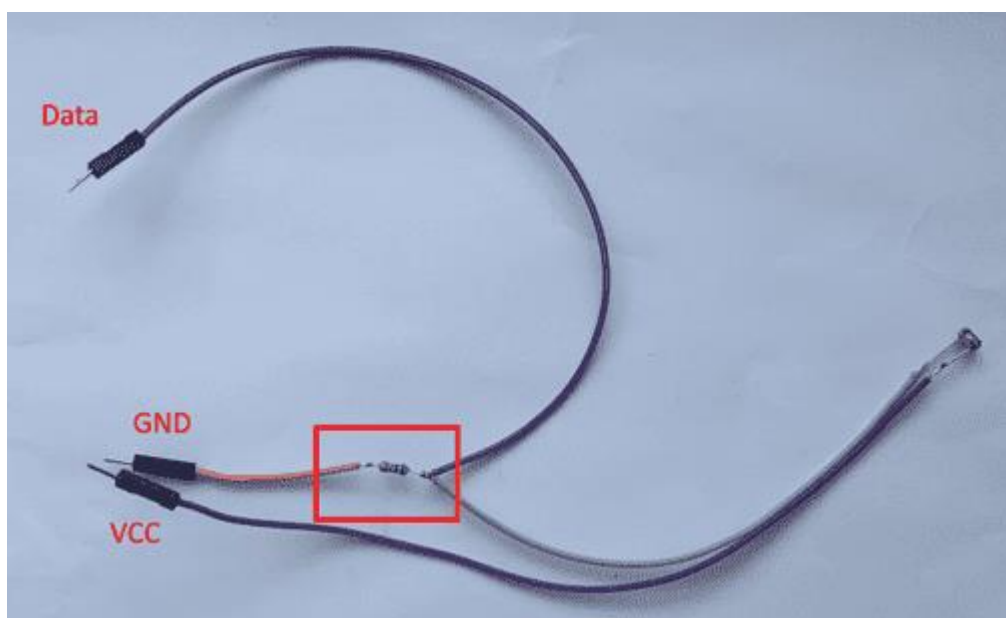


Рис. 4.18. Результат впаювання резистор

Після спаювання необхідно підключити вихід *Data* до аналогового виходу на платі, вихід *VCC* до виходу *5V* на платі, вихід *GND* до аналогічного виходу на платі.

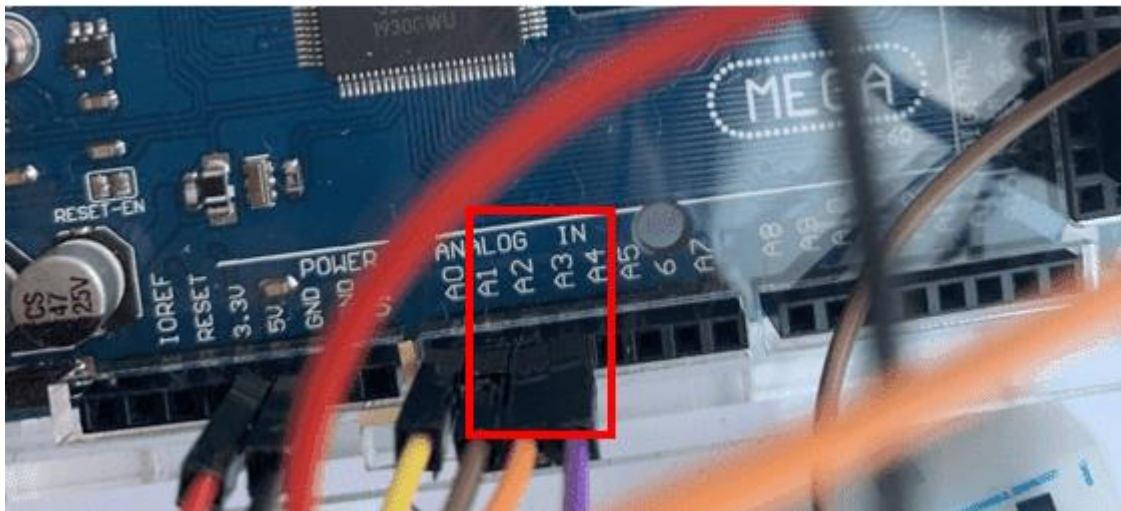


Рис. 4.19. Результат підключення трьох фоторезисторів

Написання коду після підключення фоторезисторів. Для початку необхідно вказати виходу, до яких підключені фоторезисторів:

```
#define PIN_PHOTO_SENSOR1 A1
#define PIN_PHOTO_SENSOR2 A2
#define PIN_PHOTO_SENSOR3 A3
```

Після цього необхідно виконати зчитування з трьох датчиків за допомогою функції *analogRead()* та обрахувати середнє арифметичне опору, який чинять три фоторезистори:

```
fotores1 = analogRead(PIN_PHOTO_SENSOR1);
fotores2 = analogRead(PIN_PHOTO_SENSOR2);
fotores3 = analogRead(PIN_PHOTO_SENSOR3);
ser_fotores=(fotores1+fotores2+fotores3)/3;
```

Після обрахування необхідно виконати переведення опору у відсотки та люкси за допомогою функції *map()*:

```
ser_fotores_percent=map(ser_fotores,LIGHT_MIN,LIGHT_MAX,0,100);
ser_fotores_lux=map(ser_fotores, LIGHT_MIN,LIGHT_MAX,0.1,10000);
```

Після переведення виводимо результат на дисплей:

```
lcd.setCursor(0, 0);  
lcd.print("ligt_lvl: ");  
lcd.print(ser_fotores_percent);  
lcd.print(" %");  
lcd.setCursor(0, 1);  
lcd.print("lux_lvl: ");  
lcd.print(ser_fotores_lux);  
lcd.print(" %");
```



Рис. 4.20. Результат обчислення ступеня освітлення

4.4.6. Підключення п'єзоелемента, для виведення звукового сповіщення

Підключення динаміка відбувається наступним чином: вихід «+» на елементі підключається до цифрового виходу на платі *Arduino*, а вихід «-» підключається до виходу *GND*.

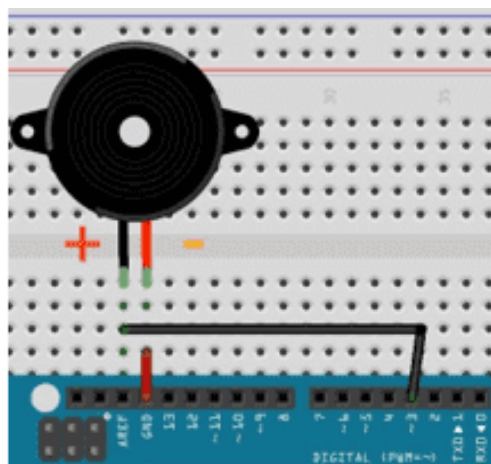


Рис. 4.21. Схема підключення динаміка

Після підключення п'єзоелемента необхідно описати код для роботи з елементом. Перш за все необхідно вказати номер цифрового піна на платі до якого підключений елемент:

```
int piezo=32;
```

Після цього вказуємо, що даний елемент виводить інформацію, а не зчитує її.

```
pinMode(piezo,OUTPUT);
```

Після вказування даної інформації можемо виводити звук за допомогою функції *tone()*, яка приймає аргументом номер піна та частоту звука та глушити звук за допомогою функції *noTone()*:

```
tone(piezo,200);
```

```
delay(500);
```

```
noTone(piezo);
```

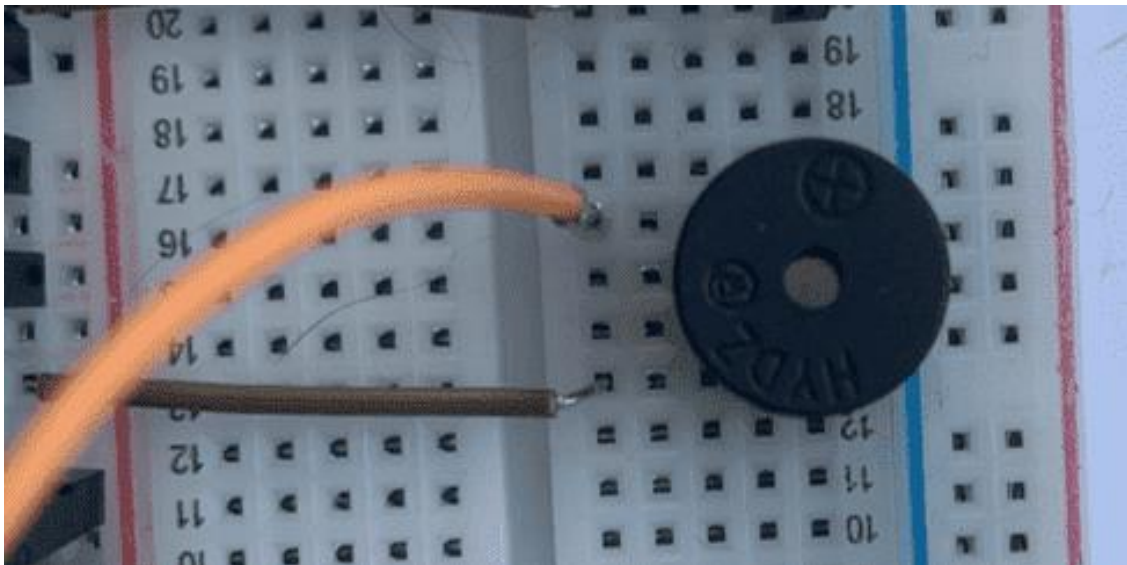


Рис. 4.22. Підключення елемента на практиці

4.4.7. Підключення діода, для відображення діапазонів рівня освітлення

Підключення діода відбувається за наступною схемою:

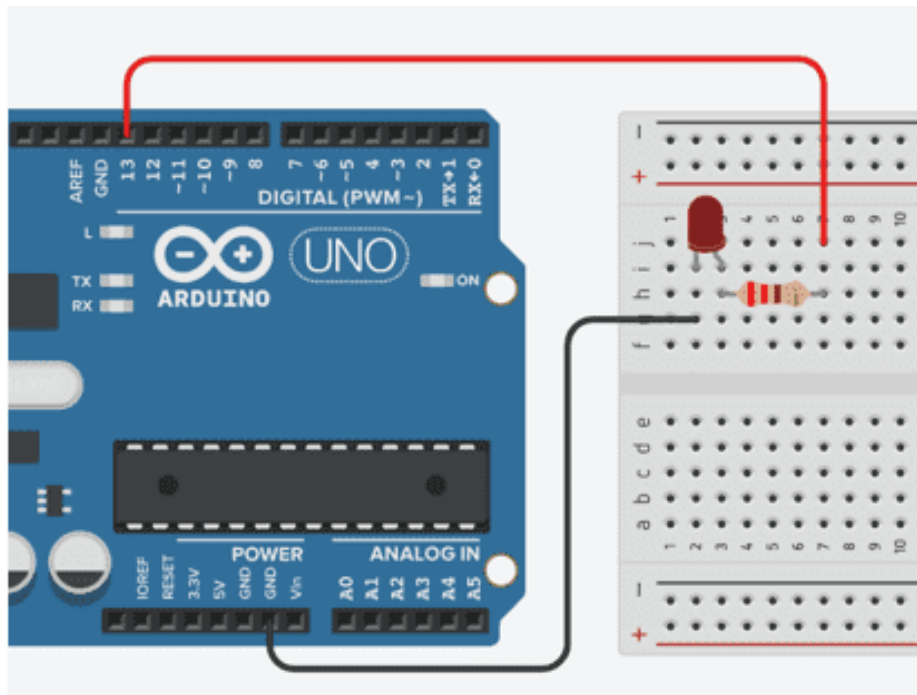


Рис. 4.23. Схема підключення діода

Як видно зі схеми для підключення діода необхідний резистор на 220 Ом, від опору резистору залежить наскільки яскраво буде світити діод. Для легшого підключення діоду резистор необхідно впаяти в даний ланцюг підключення.

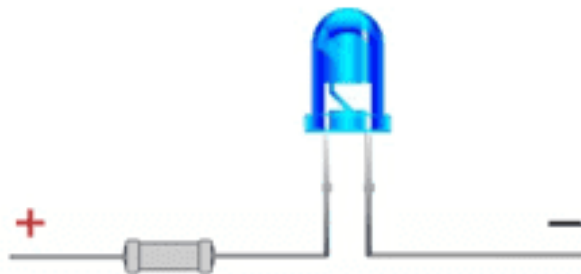


Рис. 4.24. Схема спаювання діоду і резистору

На практиці результат спаювання буде виглядати наступним чином (для уникнення замикання на вихід із припаяним резистором було одягнуто пластикову трубку), оскільки для реалізації приладу необхідні були три діоди, виконуємо дану процедуру для трьох діодів:



Рис. 4.25. Результат спаювання діоду і резистору

Програмний опис роботи діоду. Спочатку необхідно вказати номери виходів до яких підключаються діоди:

```
int red_diod_ft=53;
```

```
int yellow_diod_ft=51;
```

```
int green_diod_ft=49;
```

Після цього необхідно вказати, що вихід, до якого підключений діод, слугує для виводу інформації, а не зчитування:

```
pinMode(red_diod_ft,OUTPUT);
```

```
pinMode(yellow_diod_ft,OUTPUT);
```

```
pinMode(green_diod_ft,OUTPUT);
```

Після цього можна користуватись діодом, оскільки діод слугує для відображення інтервалу ступеня освітлення (0-30% - темно(червоний діод), 30%-70% - середнє освітлення(жовтий діод), 70%-100% - гарне освітлення(зелений діод)) одразу необхідно описати функцію для реалізації даної операції:

```
void light_func(float light_var){
```

```
if(light_var<=30){
```

```
digitalWrite(red_diod_ft, HIGH);
```

```
digitalWrite(green_diod_ft, LOW);
```

```
digitalWrite(yellow_diod_ft, LOW);
```

```
}
```

```
else if(light_var>30 && light_var<70){
```

```
digitalWrite(red_diod_ft, LOW);
```

```
digitalWrite(green_diod_ft, LOW);  
digitalWrite(yellov_diod_ft, HIGH);  
}  
else{  
digitalWrite(green_diod_ft, HIGH);  
digitalWrite(red_diod_ft, LOW);  
digitalWrite(yellov_diod_ft, LOW);  
}  
}
```

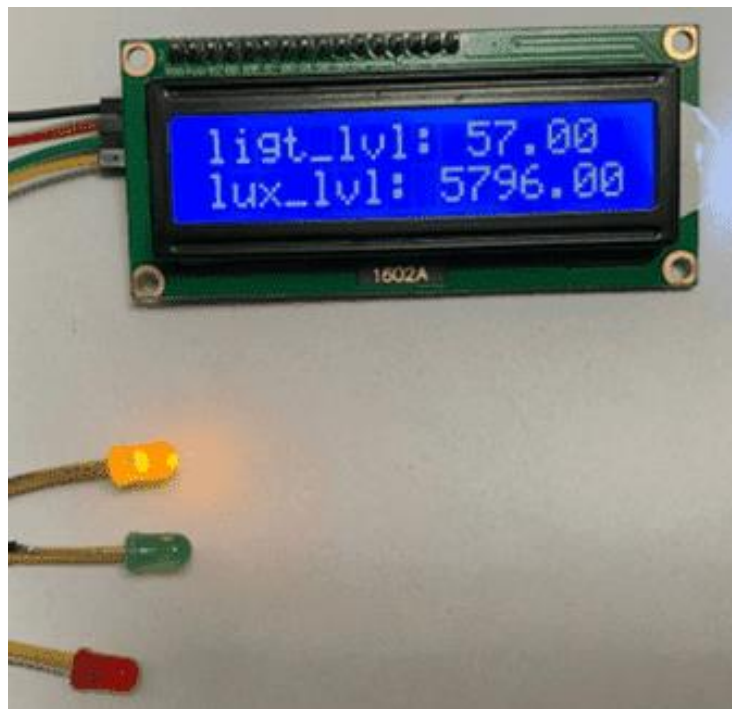


Рис. 4.26. Результат підключення діодів

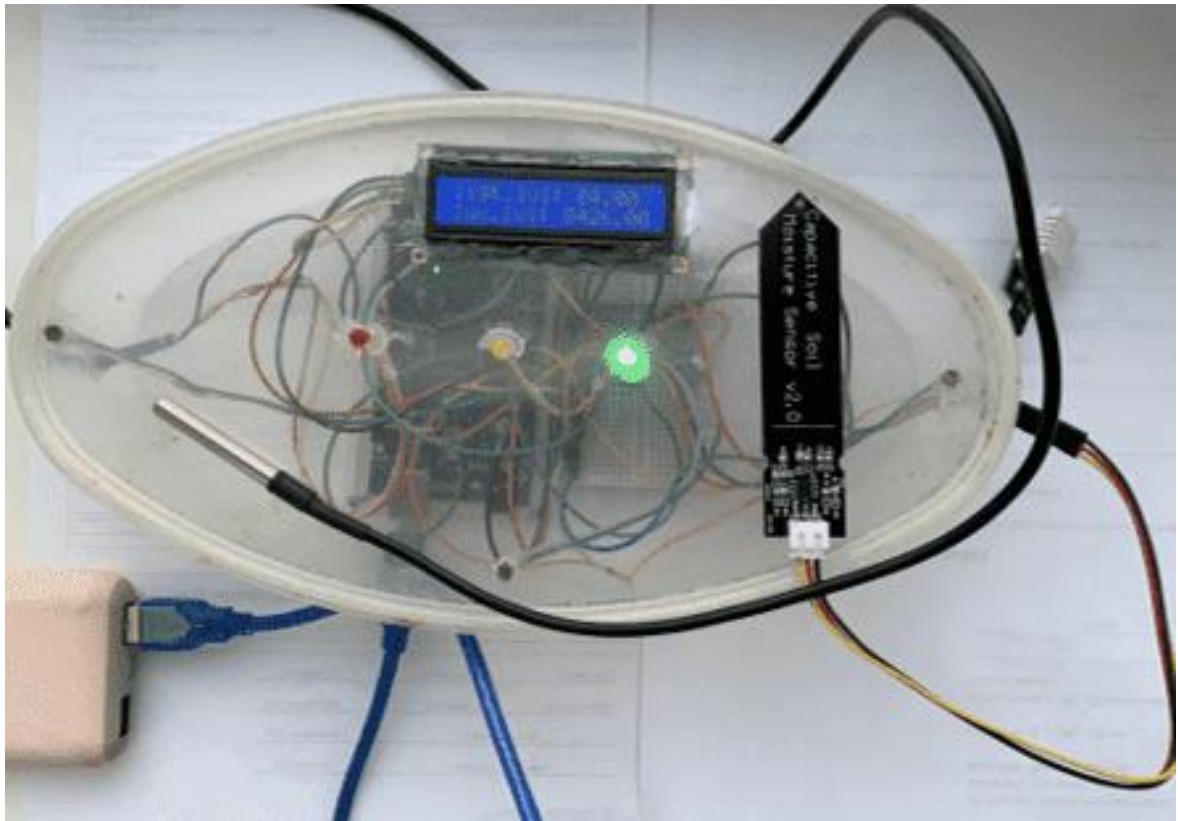


Рис. 4.27. Аналізатор параметрів повітря та ґрунту після підключення датчиків, діодів та п'єзоелемент

4.5. Висновки до розділу

В даному розділі було описано практичну реалізацію аналізатора параметрів повітря та ґрунту. Спочатку було встановлено плату у пластмасовий корпус, для уникнення фізичних пошкоджень мікроконтролера, після цього було підключено *LCD*-дисплей та датчики: датчик температури та вологості повітря *DHT22*, датчик температури *DS18B20*, ємнісний датчик вологості ґрунту, фоторезистор та світлодіоди.

Після підключення було написано код для роботи з даними датчика та дисплеями та виконано тестування коректної роботи приладів.

ВИСНОВКИ

Кліматичні фактори відіграють визначну роль лише тому, що вони глибоко впливають на всі інші фактори. На розвиток ґрунту, безпосередньо впливає клімат. Температура, кількість та розподіл опадів на рік в основному визначаються поділом земельного покриву. Під кліматичними факторами ми розуміємо: умови температури, вологості повітря та ґрунтів, світла, вітру, тиску. Всі ці фактори діють, в першу чергу, не ізольовано. Одна і та ж температура має різне значення залежно від кількості опадів, на яку вона діє. Людина самостійно не здатна кардинально змінити клімат на території де вона мешкає, але для вирощування певних екзотичних рослин або рослин, які притаманні для даного регіону, можна створити певні кліматичні умови локально, наприклад у теплиці. Для створення даних умов людині необхідний прилад, який буде допомагати контролювати кліматичні умови.

Під час написання дипломного проекту було створено аналізатор параметрів повітря та ґрунту. Для досягнення цієї мети було виконано такі завдання:

1. Проведено аналіз основних параметрів повітря та ґрунту.
2. Розроблено вимоги до приладу та інструкції користувача.
3. Визначено технічні та програмні засоби для реалізації пристрою.
4. Розроблено *UML*-діаграми відповідно до вимог та роботи системи.
5. Розроблено прилад на основ.

В результаті виконання даних завдань було розроблено систему, яка контролює основні кліматичні умови, необхідні для вирощування рослин: вологість та температура ґрунту, вологість та температура повітря та освітлення.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *C. Platt Make:Electronics - Make Community*,2015. – 352с.
2. Положення про дипломні роботи (проекти) випускників національного авіаційного університету - СМЯ НАУ П 03.01(10) – 02 – 2017
3. ДСТУ ГОСТ 7.1:2006.Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання.
4. ДСТУ 3008-95.Дослідження. Звіти у сфері науки і техніки. Структура і правила оформлення.
5. Бойченко С.В., Іваненко О.В. Положення про дипломні роботи (проекти) випускників Національного авіаційного університету. – К.: НАУ, 2017. – 63 с.
6. ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.
7. А.А. Биняковский, В. Петин Практическая энциклопедия *Arduino* – ДМК Пресс,2020. – 166с.
8. В. Петин Проекты с использованием контроллера *Arduino*, 2-е издание - БХВ-Петербург, 2015. – 464с.
9. Дж. Боксел Изучаем *Aruiino*/ 65 проектов своими руками– ПитерЯзык, 2017. – 400с.
10. Б. Хуанг, Д. Ранберг *Arduino* для изобретателей. Обучение электронике на 10 занимательных проектах – БХВ-Петербург,2019. – 288с.
11. Монк С Шерц П. Электроника теория и практика 4-е издание – БХВ-Петербург, 2017. – 1168с.
12. *J.J.Purdum Beginning C for Arduino/Second Edition* – Apress, 2015. – 414с.
13. *A.J. Boloor Arduino by Example* – Packt Publishing, 2015. – 242с.
14. С. Монк Програмируем *Arduino* – Питер, 2017. – 208с.
15. У. Соммер Программирование микроконтроллерных плат *Arduino* – БХВ-Петербург, 2016. – 256с.

ДОДАТОК А
ФРАГМЕНТ ПРОГРАМНОГО КОДУ

```
#include <DallasTemperature.h>
#include "DHT.h"
#include <Wire.h>
#include <LiquidCrystal_I2C.h> // Library for LCD
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include <OneWire.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
#define DHTPIN 2

#define SENSOR_GR A0
#define SENSOR_GR_MIN 622.0
#define SENSOR_GR_MAX 263.0

#define PIN_PHOTO_SENSOR1 A1
#define PIN_PHOTO_SENSOR2 A2
#define PIN_PHOTO_SENSOR3 A3
#define LIGHT_MIN 0.1
#define LIGHT_MAX 1023

#define ONE_WIRE_BUS 5 //вихід data на термістори
#define temp_power 4//вихід vcc на термістори
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature t_gr_sensor(&oneWire);
```

```

uint16_t sensor_gr;
int piezo=32;
float ser_fotores_percent=0.0;
float ser_fotores_lux=0.0;
int fotores1=0;
int fotores2=0;
int fotores3=0;
DHT dht(DHTPIN, DHT22); //Инициация датчика
int red_diod_ft=53;
int yellow_diod_ft=51;
int green_diod_ft=49;
float ser_fotores=0.0;
float temp_ground=0.0;
//функція для обробки сухого ґрунту
void piezo_func(uint16_t gr_var){
    if(gr_var<5){
        lcd.clear();
        lcd.setCursor(0,0); // Set the cursor on the third column and first row.
        lcd.print("Not data on");
        lcd.setCursor(0,1);
        lcd.print("hum_gr sensor!");
    }
    if(gr_var>5 && gr_var<40){
        lcd.clear();
        lcd.setCursor(0, 0); // Set the cursor on the third column and first row.
        lcd.print("hum_gr: ");
        lcd.print(float(gr_var));
        lcd.setCursor(2, 1); // Set the cursor on the third column and first row.
        lcd.print("Very dry! ");
        tone(piezo,400);
    }
}

```

```

    delay(500);
    noTone(piezo);
    delay(500);
    tone(piezo,200);
    delay(500);
    noTone(piezo);
    delay(500);
    tone(piezo,100);
    delay(500);
    noTone(piezo);
  }
}

//функція для виводу фоторезисторів
void light_func(float light_var){
  if(light_var<=30){
    digitalWrite(red_diod_ft, HIGH);
    digitalWrite(green_diod_ft, LOW);
    digitalWrite(yellow_diod_ft, LOW);
  }
  else if(light_var>30 && light_var<=70){
    digitalWrite(red_diod_ft, LOW);
    digitalWrite(green_diod_ft, LOW);
    digitalWrite(yellow_diod_ft, HIGH);
  }
  else{
    digitalWrite(green_diod_ft, HIGH);
    digitalWrite(red_diod_ft, LOW);
    digitalWrite(yellow_diod_ft, LOW);
  }
}
}

```



```

//функція термістора для ґрунта
float gr_temperature(){
    digitalWrite(temp_power,HIGH); // вмикаємо живлення датчика
    delay(100);//затримка
    t_gr_sensor.requestTemperatures();//зчитування 1 значення
    delay(500);//затримка півсекунди перед повторним визначенням
    t_gr_sensor.requestTemperatures();//зчитування повторного значення
    float t=float(t_gr_sensor.getTempCByIndex(0));//записуємо значення з датчика у
змінну
    digitalWrite(temp_power,LOW);//вимикаємо живлення датчика
    delay(5000);
    return (t);//повернення значення
}

void setup() {
    Serial.begin(9600);
    dht.begin();
    lcd.init();           // ініціалізація LCD
    lcd.backlight();     // вмикаємо підсвітку
    lcd.clear();
    pinMode(piezo,OUTPUT);
    pinMode(red_diod_ft,OUTPUT);
    pinMode(yellow_diod_ft,OUTPUT);
    pinMode(green_diod_ft,OUTPUT);
    t_gr_sensor.begin();
    pinMode(temp_power,OUTPUT);
    lcd.setCursor(1, 0); // Set the cursor on the third column and first row.
    lcd.print("Cheking data!");
}

void loop() {

```

```

delay(4000); // 2 секунды задержки

//Повітря dht
float dht_hum = dht.readHumidity(); //Измеряем влажность
float dht_tem = dht.readTemperature(); //Измеряем температуру
if (isnan(dht_hum) || isnan(dht_tem)) { // Проверка. Если не удастся считать
показания, выводится «Ошибка считывания», и программа завершает работу
  Serial.println("Error DHT22");
  return;
}

//Вологість ґрунту
sensor_gr=analogRead(SENSOR_GR);
  Serial.println("sensor_gr");
Serial.println(float(sensor_gr),1);
  float sensor_gr_per;

sensor_gr_per=float(map(sensor_gr,SENSOR_GR_MIN,SENSOR_GR_MAX,0.0,100.
0));
//temp ground
temp_ground=gr_temperature();
//фоторезистор
fotores1 = analogRead(PIN_PHOTO_SENSOR1);
  fotores2 = analogRead(PIN_PHOTO_SENSOR2);
  fotores3 = analogRead(PIN_PHOTO_SENSOR3);
  ser_fotores=(fotores1+fotores2+fotores3)/3;
  ser_fotores_percent=map(ser_fotores,LIGHT_MIN,LIGHT_MAX,0.0,100.0);
  ser_fotores_lux=map(ser_fotores,LIGHT_MIN,LIGHT_MAX,0.1,10000);

//Вивід на дисплей

```

```
//vozduh
```

```
lcd.clear();
```

```
  lcd.setCursor(2, 0); // Set the cursor on the third column and first row.
```

```
lcd.print("hum: ");
```

```
  lcd.print(dht_hum);
```

```
  lcd.print(" %");
```

```
  lcd.setCursor(2, 1); //Set the cursor on the third column and the second row  
(counting starts at 0!).
```

```
  lcd.print("tem: ");
```

```
  lcd.print(dht_tem);
```

```
  lcd.print(" C");//Вывод показателей на экран
```

```
  delay(4000);
```

```
  lcd.clear();
```

```
//ground
```

```
  lcd.setCursor(0, 0); // Set the cursor on the third column and first row.
```

```
  lcd.print("hum_gr: ");
```

```
  lcd.print(float(sensor_gr_per));
```

```
  lcd.print(" %");
```

```
  lcd.setCursor(0, 1);
```

```
  lcd.print("tem_gr: ");
```

```
  lcd.print(temp_ground);
```

```
  lcd.print(" C");
```

```
  piezo_func(sensor_gr_per);
```

```
    delay(4000);
```

```
  lcd.clear();
```

```
//light
```

```
lcd.setCursor(0, 0);  
lcd.print("ligt_lvl: ");  
lcd.print(ser_fotores_percent);  
lcd.print(" %");  
lcd.setCursor(0, 1);  
lcd.print("lux_lvl: ");  
lcd.print(ser_fotores_lux);  
lcd.print(" %");  
    //перевірка вологості  
    light_func(ser_fotores_percent);  
}
```