

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ**  
**Факультет педагогічної освіти**  
**Кафедра технологічної та професійної освіти**

«Допущено до захисту»

Завідувач кафедри ТПО

\_\_\_\_\_ Олег Цись  
« \_\_\_ » \_\_\_\_\_ 2024 р.

Реєстраційний № \_\_\_\_\_  
« \_\_\_ » \_\_\_\_\_ 2024 р.

**РОЗРОБКА Й ВИГОТОВЛЕННЯ МОДЕЛІ ГІДРОПОННОЇ  
УСТАНОВКИ НА БАЗІ ARDUINO ТА МЕТОДИКА ЇЇ  
ВИКОРИСТАННЯ ПІД ЧАС ВИВЧЕННЯ ТЕХНОЛОГІЇ В  
ПРОФІЛЬНІЙ ШКОЛІ**

Кваліфікаційна робота студента  
групи ТНм-23  
ступінь вищої освіти магістр  
спеціальності  
014.10 Середня освіта (Трудове навчання  
і технології)  
Петрикiна Дмитра Сергiйовича  
Керiвник: к.пед.н., доц.  
Цись Олег Олександрович

Оцiнка:

Нацiональна шкала \_\_\_\_\_

Шкала ECTS \_\_\_\_\_ кiлькiсть балiв \_\_\_\_\_

Голова ЕК \_\_\_\_\_

Члени ЕК \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

## ЗМІСТ

ВСТУП .....	3
1. ОСОБЛИВОСТІ ПРОГРАМНО-АПАРАТНОЇ ПЛАТФОРМИ ARDUINO .....	5
<b>1.1 Історія створення програмно-апаратної платформи Arduino.....</b>	<b>5</b>
<b>1.2 Апаратне і програмне забезпечення платформи Arduino. ....</b>	<b>8</b>
2. РОЗРОБКА І ВИГОТОВЛЕННЯ МОДЕЛІ ГІДРОПОННОЇ УСТАНОВКИ НА БАЗІ ARDUINO .....	14
<b>2.1 Особливості автоматизації поливу рослин з використанням     мікроконтролера керування. ....</b>	<b>14</b>
<b>2.2 Розробка й виготовлення моделі гідропонної установки на базі     Arduino. ....</b>	<b>20</b>
3. МЕТОДИКА ПРОВЕДЕННЯ УРОКІВ ТЕХНОЛОГІЙ З ВИКОРИСТАННЯМ ARDUINO .....	33
<b>3.1 Передумови вивчення мікроконтролерів у профільній школі. ....</b>	<b>33</b>
<b>3.2 Методика застосування Arduino на уроках технології у профільній     школі.....</b>	<b>37</b>
ВИСНОВКИ.....	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	43
ДОДАТКИ.....	46

## ВСТУП

Мікроконтролери в сучасному світі широко використовуються для керування різноманітними технічними пристроями та технологічними процесами у житті людини, в тому числі і у сфері вирощування рослин.

У світі, де кожна хвилина є цінною, турбота про домашні рослини іноді стає викликом. Автополив для кімнатних рослин – це інноваційний спосіб забезпечити їм необхідний догляд за відсутності безпосереднього втручання людини. Автоматичний полив надає низку переваг, зробивши догляд за рослинами зручнішим та ефективнішим. Це особливо актуально у ситуаціях, коли власники рослин змушені залишати свій дім на час відпустки чи ділової подорожі.[1]

Беручи до уваги те, що цифрові технології не тільки активно застосовуються в наукових дослідженнях але й входять у наше життя: автоматичний полив квітів, автоматичне та дистанційне управління електричними системами, робототехніка. Саме поява мікропроцесорних систем з відкритим вихідним кодом таких як Arduino, заснованих на простому у використанні апаратному та програмному забезпеченні, створила сприятливі умови для їх активного використання в не тільки у повсякденному житті, а і в освіті.[7]

Особливість систем автоматичного поливу рослин, з точки зору використання їх в освітньому процесі полягає в можливості об'єднати констрування та програмування в одному уроці технології, що сприяє формуванню міжпредметних зв'язків природничих наук з технологіями через технічну творчість, забезпечуючи при цьому розвиток інженерного мислення учнів.

Це зумовило вибір *теми* кваліфікаційної роботи: **«Розробка й виготовлення моделі гідропонної установки на базі Arduino та методика її використання під час вивчення технології в профільній школі.»**

**Мета роботи:** розробити модель гідропонної установки на базі Arduino та методику її використання під час вивчення технології в профільній школі.

**Об'єкт:** процес розробки й виготовлення гідропонної установки на базі Arduino.

**Предмет:** особливості застосування гідропонної установки на базі Arduino при вивченні технології у профільній школі.

У відповідності до мети визначено такі **завдання:**

1. Дослідити особливості програмно-апаратної платформи Arduino.
2. Розробити проєкт і виготовити модель гідропонної установки на базі Arduino.
3. Описати методику проведення уроків технологій з використанням Arduino.

У процесі виконання кваліфікаційної роботи були застосовані такі **методи:** аналіз, порівняння та узагальнення вітчизняного й світового досвіду, вивчення технічної документації, проєктування, моделювання.

**Практична значущість** роботи полягає в тому, що створена модель гідропонної установки на базі Arduino може бути використана в практиці навчання учнів при вивченні технології у профільній школі.

# 1. ОСОБЛИВОСТІ ПРОГРАМНО-АПАРATНОЇ ПЛАТФОРМИ ARDUINO

## 1.1 Історія створення програмно-апаратної платформи Arduino.

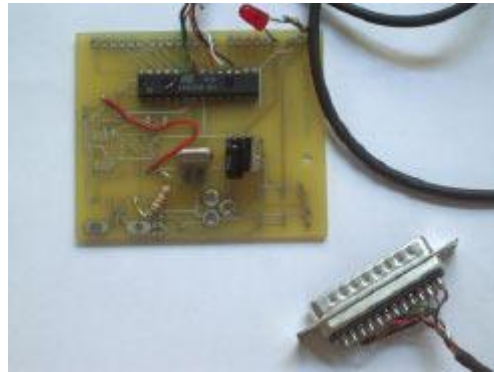
Ардуїно був створений в італійському місті Івреа, що стоїть на річці Дора Балтея. Заклад громадського харчування Pe Ардуїно, розташований в історичній частині цього містечка, названий на честь короля, який колись тут правив, і стоїть на тому самому місці, де, за переказами, народився король. У цьому кафе любляв обідати Массімо Банці (Massimo Banzi), італійський співзасновник проекту у сфері електроніки. Тому проект був названий на честь цього місця – «Ардуїно».[6]

Ардуїно – недорога мікроконтролерна плата, яка дозволяє навіть новачкові робити по-справжньому дивовижні речі. Ви можете підключити до Arduino різні типи датчиків, джерел світла, електромоторів та багато інших пристроїв, і використовувати програмне забезпечення, що легко освоюється, щоб запрограмувати поведінку вашого творіння. Ви можете створити інтерактивний дисплей або робота, що рухається, а потім розповісти про нього іншим, розмістивши звіт або відео в мережі. [6]

Випущений у 2005 році як скромний інструмент для студентів Банці в Інституті проектування взаємодій міста Івреа (Interaction Design Institute Ivrea, IDII), Arduino породив міжнародну революцію у сфері міжнародних електронних саморобок. Ви можете купити цю плату за \$30 або зібрати її з нуля. Усі схеми та вихідні коди доступні безкоштовно на умовах відкритих ліцензій. В результаті Arduino став найвпливовішою апаратною системою свого часу з відкритим кодом.

Перша плата прототипу, зроблена 2005 року, мала найпростіший дизайн (Рис. 1.1) і ще не називалася Arduino. Трохи пізніше, того ж року, Массімо Банці придумав їй ім'я.

Рис. 1.1



Маленька друкована плата тепер є, свого роду, джерелом натхнення для «художників» в електроніці, людей, захоплених електронними виробами, студентів та будь-кого, хто має мрію зібрати щось таке. враховуючи безліч клонів. «Це дозволило людям робити речі, які б вони не зробили будь-яким іншим способом» – говорить Девід А. Меліс (David A. Mellis), який до того, як виконав кваліфікаційну роботу в Медіа, був студентом IDІІ лабораторії Массачусетського технічного інституту, а тепер є провідним розробником програмного забезпечення для Arduino. [6]

Ядро команди розробників Arduino (зліва направо): Девід Куар-Тіллз (David Cuartielles), Джанлука Мартіно (Gianluca Martino), Том Іго (Tom Igoe), Девід Меліс (David Mellis), і Массімо Банці (Massimo Banzi) на конференції Maker Faire Нью-Йорку:

Рис. 1.2



Сьогодні є безліч цікавих розробок на базі Arduino, таких як алкотестери, світлодіодні куби, системи домашньої автоматизації, дисплеї

відображення Twitter-повідомлень і навіть набори для аналізу ДНК. Вже з'явилися цілі клуби та спільноти любителів Arduino. Google нещодавно випустив заснований на Arduino комплект розробника свого смартфона на базі Android.

Але Arduino – це не тільки загальнодоступний проект, який прагне зробити технології доступнішими. Це також стартапова компанія, якою керує Банці та невелика група його друзів.

Своїм успіхом Arduino багато в чому зобов'язаний Processing і Wiring, що до нього існував. Від цих проектів Arduino успадкував одну сильну рису – зручне для користувача середовище розробки. До появи Ардуїно програмування мікроконтролерів супроводжувалося складним та рутинним навчанням. З Arduino навіть ті, хто не мав досвіду роботи з електронними пристроями, можуть проникнути в загадковий для них світ електроніки. Тепер новачкам не потрібно витратити багато часу на вивчення відповідного матеріалу, вони можуть швидко розробити прототип, який буде повноцінно робітником. Це потужний крок вперед, у той час як деякі досить популярні гаджети є «чорними ящиками», закритими і захищеними патентами.

Але не всі інженери люблять Ардуїно. Найбільш прискіпливі з них критикують цей продукт за досить примітивний, з погляду інжинірингу, підхід до розробки та наповнення ринку аматорської радіоелектроніки сірим дилетантським товаром. Однак, не слід розглядати Ардуїно як загрозу знецінення ролі інженера взагалі. Массімо Банці каже: «Маючи платформу, яка дозволяє проектувальнику чи будь-якій іншій творчій особистості трішки наблизитися до своєї мети, люди можуть тепер спростити собі взаємодію з фахівцями та сказати «Ось те, що я хочу в результаті зробити». Я не думаю, що це заміна інженера, це лише полегшує співпрацю».

Саме тому Ардуїно підходить впершу чергу для навчальних цілей та учнівських проектів.

## 1.2 Апаратне і програмне забезпечення платформи Arduino.

Апаратна частина платформи Arduino. Плата Arduino Uno.

Існує кілька версій платформ Arduino. Остання версія Leonardo базується на мікроконтролері ATmega32u4. Uno, як і попередня версія Duemilanove, побудовані на мікроконтролері Atmel ATmega328. Старі версії платформи Diecimila та перша робоча Duemilanoves були розроблені на основі Atmel ATmega168, більш ранні версії використовували ATmega8. Arduino Mega2560, у свою чергу, побудована на мікроконтролері ATmega2560. На всю документацію Arduino поширюється ліцензія 3.0 Creative Commons Attribution.[3]

Нижче наведено основні версії плат Arduino:

- Due – нова плата на базі ARM мікропроцесора 32bit Cortex-M3 ARM SAM3U4E.
- Leonardo – остання версія платформи Arduino на ATmega32u4 мікроконтролері. Відрізняється роз'ємом microUSB, за розмірами збігається з UNO.
- Yun – нова плата, з вбудованою підтримкою WiFi на базі ATmega32u4 and the Atheros AR9331
- Micro – нове компактне рішення на базі ATmega32u4.
- Uno – найпопулярніша версія базової платформи Arduino USB. Uno має стандартний порт USB. Arduino Uno багато в чому схожа на Duemilanove, але має новий чіп ATmega8U2 для послідовного підключення по USB і нове, зручніше маркування вхід/виходів. Платформа може бути доповнена платами розширення, наприклад, платами користувача з різними функціями.



- Arduino Ethernet – контролер із вбудованою підтримкою роботи по мережі та з опціональною можливістю живлення по мережі за допомогою модуля POE (Power over Ethernet).
- Duemilanove є передостанньою версією базової платформи Arduino USB. Підключення Duemilanove здійснюється стандартним кабелем USB. Після підключення вона готова до використання. Платформа може бути доповнена платами розширення, наприклад, платами користувача з різними функціями.
- Diecimila – попередня версія базової платформи Arduino USB.
- Nano – це компактна платформа, яка використовується як макет. Nano підключається до комп'ютера за допомогою USB-кабелю Mini-B.
- Mega ADK – версія плати Mega 2560 з підтримкою USB host інтерфейсу для зв'язку з телефонами на Android та іншими пристроями з USB інтерфейсом.
- Mega2560 – нова версія плати серії Mega. Побудована на базі Atmega2560 та з використанням чіпа ATmega8U2 для послідовного з'єднання USB портом.
- Mega – попередня версія серії Mega на базі Atmega1280.
- Arduino BT платформа з модулем Bluetooth для бездротового зв'язку та програмування. Сумісна з платами розширення Arduino.
- LilyPad – платформа, пурпурового кольору, розроблена для перенесення, може зашиватися в тканину.
- Fio – платформа розроблена для бездротового використання. Fio містить роз'єм для радіо XBee, роз'єм для батареї LiPo та вбудовану схему підзарядки.

- Mini – найменша платформа Arduino. Прекрасно працює як макетна модель, або в проектах, де простір є критичним параметром. Платформа підключається до комп'ютера за допомогою адаптера Mini USB.
- Адаптер Mini USB – плата, що конвертує підключення USB у лінії 5 V, GND, TX та RX для з'єднання з платформою Arduino Mini або іншими мікроконтролерами.
- Pro – платформа, розроблена для досвідчених користувачів, може бути частиною більшого проекту. Вона дешевша, ніж Diecimila і може живитися від акумуляторної батареї, але в той же час потребує додаткового збирання та компонентів.
- Pro Mini – як і платформа Pro розроблена для досвідчених користувачів, яким потрібна низька ціна, менші розміри та додаткова функціональність.
- Serial – базова платформа з інтерфейсом RS232 для зв'язку та програмування. Плата легко збирається навіть початківцями. (включає схеми та файли CAD)
- Serial Single Sided – платформа розроблена для ручного збирання. Вона має трохи більший розмір, ніж Diecimila, але сумісна з платами розширення Arduino.
- USB Serial Light Адаптер – адаптер, що дозволяє підключати плати Arduino до комп'ютера для обміну даними та заливання скетчів. Зручний для програмування таких плат, як Arduino Mini, Arduino Ethernet та інших, які не мають роз'єму USB.[3]

Плати розширення.

Плати розширення, що встановлюються на платформи, є плати, що розширюють функціональність Arduino для управління різними пристроями, отримання даних та ін.:

- Плата розширення WiFi використовується для з'єднання з бездротовими мережами стандарту 802.11 b/g.
- Плата розширення Xbee Shield забезпечує за допомогою модуля Maxstream Xbee Zigbee бездротовий зв'язок кільком пристроям Arduino в радіусі до 35 метрів (в приміщенні) і до 90 метрів (поза приміщенням).
- Плата розширення Motor Shield забезпечує керування двигунами постійного струму та читання датчиків положення.
- Плата розширення Ethernet Shield забезпечує підключення до Інтернету.

#### Програмування Ардуїно.

Мова програмування пристроїв Ардуїно ґрунтується на C/C++. Він простий у освоєнні, і зараз Arduino – це, мабуть, найзручніший спосіб програмування пристроїв на мікроконтролерах. Середовище розробки для Arduino це спеціальна програма останню версію якої завжди можна завантажити на сторінці завантаження.

Для підключення до комп'ютера Arduino Leonardo або Micro знадобиться кабель USB з роз'ємом Micro-B. Такий USB-кабель служить як передачі даних, так живлення плати. Щоб прошити Arduino Leonardo, у середовищі розробки Arduino IDE з меню Tools > Board необхідно вибрати пункт Arduino Leonardo або Arduino Micro.[13]

#### Відмінності від Arduino Uno.

В цілому, порядок роботи та програмування Leonardo і Micro такий самий, як і в інших моделях Ардуїно. Проте існує й кілька важливих відмінностей:[13]

#### *Єдиний процесор.*

Leonardo і Micro відрізняються від інших моделей Ардуїно, перш за все тим, що в них для виконання всіх функцій використовується один

мікроконтролер – і для виконання програм, і для взаємодії з комп'ютером за інтерфейсом USB. В Arduino Uno та інших моделях для виконання цих двох функцій використовують різні мікроконтролери, відповідно, USB з'єднання з комп'ютером завжди залишається активним, незалежно від поточного стану головного мікроконтролера. Об'єднання обох функцій в один мікроконтролер дозволило зробити Leonardo більш гнучким у взаємодії з комп'ютером, а також зменшити собівартість за рахунок відсутності додаткового процесора.

*Перепризначення порту під час скидання плати.*

У складі пристрою немає окремої мікросхеми, що обслуговує послідовний зв'язок, тому послідовні порти, що використовуються, є віртуальним – як в операційній системі ПК, так і на самому пристрої. Під час запуску завантажувача Leonardo/Micro автоматично створює екземпляр класу для роботи з послідовним інтерфейсом, так само, як комп'ютер створює екземпляр драйвера послідовного порту при підключенні Ардуїно. У системі підключена плата є екземпляром спеціального класу драйвера для роботи з USB (Connected Device Class – CDC).

Така організація призводить до того, що при кожному скиданні плати USB-з'єднання з комп'ютером буде розриватися і встановлюватися знову. У системі пристрій буде зникати зі списку доступних портів, відповідно послідовні порти перенумеровуватимуться. Тому всі програми, в яких було відкрито порт Ардуїно, втратять з'єднання з пристроєм. Це одна з очевидних відмінностей Leonardo/Micro від Arduino Uno, в якому скидання мікроконтролера (ATmega328P) не призводить до розриву USB-з'єднання (яке підтримується додатковим процесором ATmega8U2 або ATmega16U2). Ці відмінності приводять до деяких особливостей установки драйвера, процесу прошивки та взаємодії з пристроєм, які описані нижче.

*Відсутність скидання під час відкриття послідовного порту.*

На відміну від Arduino Uno, Arduino Leonardo та Micro не перезапускають завантажений скетч при кожному відкритті послідовного порту на комп'ютері. Це означає, що комп'ютер не отримає дані, надіслані платою до відкриття операційної системи послідовного порту (у тому числі, наприклад, дані, що надсилаються в блоці `setup()`).

Така зміна призводить до того, що будь-які функції передачі послідовних даних (такі, як `print()`, `println()`, `write()` та інших.), написані у блоці `setup`, нічого очікувати викликатися в останній момент відкриття послідовного порту. Щоб обійти цю проблему, можна програмно перевіряти статус послідовного порту після виклику функції `Serial.begin()`, наприклад:

```
Serial.begin(9600);
```

```
// пока последовательный порт не будет открыт - ничего не делаем:
```

```
while (!Serial) ;
```

*Емуляція клавіатури та миші.*

Одна з переваг використання єдиного процесора для виконання програм та USB-зв'язку – це гнучкість при взаємодії пристрою з комп'ютером. Зокрема, завдяки цьому Ардуїно може визначатися в системі не тільки як віртуальний послідовний порт (також званий CDC), що використовується для прошивки та передачі даних (подібно до Arduino Uno), але і працювати як HID-пристрій, емулюючи клавіатуру або мишу.

*Роздільна функціонування USB та UART.*

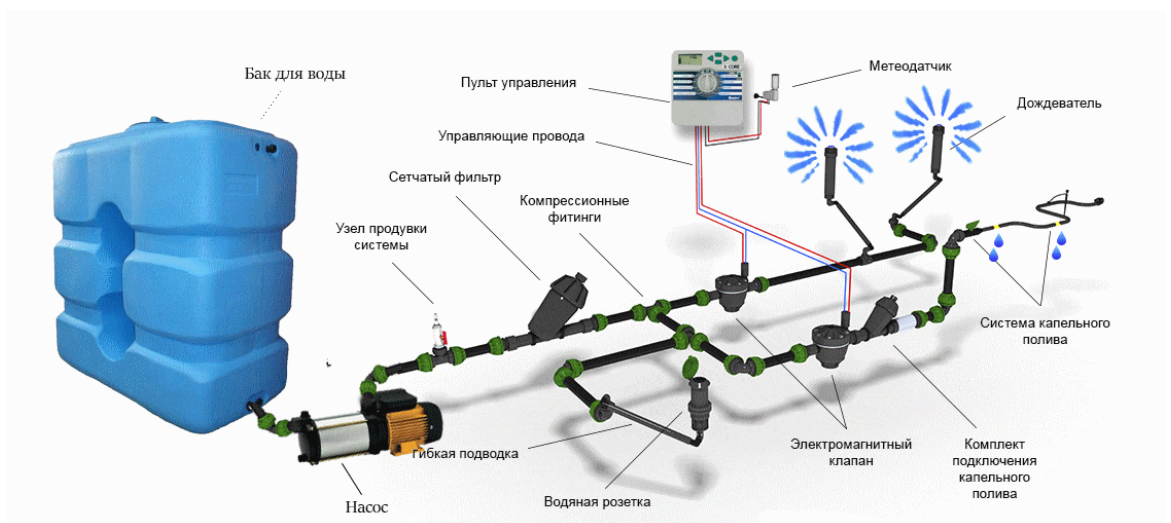
Arduino Leonardo і Micro, основний клас `Serial` асоційований з драйвером віртуального послідовного порту для з'єднання з комп'ютером через USB. Фізично він не пов'язаний з висновками 0 і 1, як це зроблено на платах Arduino Uno і більш старих версіях. Для користування апаратним послідовним портом (висновками 0 і 1, RX і TX) використовуйте клас `Serial1`. (Для отримання додаткової інформації див. довідку класу `Serial`).

## 2. РОЗРОБКА І ВИГОТОВЛЕННЯ МОДЕЛІ ГІДРОПОННОЇ УСТАНОВКИ НА БАЗІ ARDUINO

### 2.1 Особливості автоматизації поливу рослин з використанням мікроконтролера керування.

Гарний та регулярний полив рослин – запорука гарного росту рослин та врожаю. Вода життєво необхідна рослинам, без неї рослини просто зав'януть і загинуть. На жаль, вчасно полити свої рослини можуть не всі. В наш час це вже не є проблемою, адже нові технології, засновані на використанні електроніки та мікропроцесорів, увійшло міцно в наш побут. Навіть звичайний догляд за рослинами, зараз можна автоматизувати та довірити автоматичним системам.[2]

Рис. 2.1



Автоматизація поливу – це особливий технічний комплекс, який самостійно здатний забезпечити рівномірний та регулярний полив певної території. Насамперед, правильний полив залежить від показання датчиків. Більше того, вони мають бути розташовані по всій ділянці. Тобто це можуть бути датчики зволоження ґрунту, температури, вологості навколишнього середовища та датчик опадів. Щоб створити автоматизацію поливу на своїй

ділянці, необхідно чітко уявляти, які частини вашої ділянки потребують цього. [2]

Так, є три основні види системи поливу: дощувальний, краплинний та внутрішньогрунтове зрошення. Для уточнення, розглянемо переваги та недоліки кожної системи.

Перший вид (дощувальний) має особливий попит, тому що нагадує природні опади, що дуже подобається рослинам.

Рис. 2.2



Адже, основна складова даної системи є невеликі фонтанчики води, які розташовуються по ділянці. Вони можуть мати різний кут зрошення і бути поворотними або статичними. Таким чином, комбінуючи на своїй ділянці статичні та поворотні системи, можна легко забезпечити рівномірний полив навіть у найвіддаленіших місцях або, навпаки, обмежити територію. Потрібно враховувати, що основний принцип розміщення розпилювачів у дощувальній системі полягає в тому, що радіус поливу сусідніх розпилювачів

повинен повністю перекриватися. Тобто, після поливу на території практично не повинно залишатися сухих ділянок.

#### Переваги:

- Полив за допомогою розсіяної води не шкодить структурі ґрунту та не підмиває кореневу систему рослин;
- Ґрунт зволожується глибоко та якісно;
- Підвищується вологість повітря;
- Відсутній рух води поверхнею, тобто родючий шар не підмивається;
- Змивається пил із листя, що нормалізує їх обмін речовин;
- Вологою насичується і надземна частина рослин – стовбури та листя, що у свою чергу підвищує їхню врожайність;
- Великий радіус дії водяних бризок.

#### Недоліки:

- Якщо вчасно не вимкнути систему поливу, на ділянці локально утворюються калюжі зі стоячою водою. І як наслідок, ґрунт перестає вбирати воду. Тому необхідно дотримуватися часу поливу – не більше 30 хвилин.
- В результаті, при недотриманні сказаного вище на ґрунті утворюється тверда кірка, яка перешкоджає доступу кисню до коренів;
- А також, при сильному вітрі водні бризки будуть зноситися в різні боки, що унеможливить рівномірний полив;
- Окрім того того, неправильний радіус дії буде і за слабкого тиску води в магістралі.

Другий вид (краплинний), являє собою зрошувальні системи, що доставляють воду безпосередньо в зону висадки рослини.



Рис. 2.3



Насамперед, спрямовано зрошуючи її кореневу систему. Безумовно, цей спосіб є найбільш щадним та економічним. Насправді його доцільно використовувати для прикореневого поливу рослин, які особливо чутливі до посухи. Безперечно, подібна система зрошення ділянки в основному використовується для поливу дерев, чагарників, теплиць та городніх рослин (для поливу представників флори, що мають глибоку кореневу систему). Як правило, принцип розміщення поливального обладнання в подібних системах полягає в тому, що водяні магістралі з поливальними крапельницями (крапельні стрічки) розташовуються вздовж посадкових рядів на невеликій відстані від стовбурів рослин.

Краплинний полив має безліч переваг:

- Завдяки тому, що вода відразу потрапляє на коріння, витрата її мінімальна, а зволоження при цьому більш ніж достатнє;
- Забезпечується рівномірна подача води до рослин;

- Система практично не чутлива до перепадів тиску;
- Рослини не обов'язково поливати рано-вранці або ввечері. Так як стебла та листя залишаються сухими, і рослини не обгорять під палючим сонцем;
- Зберігається повноцінний доступ кисню до коренів;
- На поверхні землі не утворюється тверда кірка.

До недоліків можна віднести лише часті забруднення дрібних отворів на нижній частині крапельного шланга. Крім того, що в отворах і так осідатиме наліт, нижні ряди забиватимуться частинками вологого ґрунту. У зв'язку з цим багато досвідчених садівників рекомендують прокладати шланг на невеликій висоті над рівнем ґрунту. Тоді цієї проблеми вдасться уникнути.

Третій вид. Внутрішньогрунтове зрошення.

Рис. 2.5



Це поливальні системи для підземного (внутрішньогрунтового) зрошення. Пластикові труби розподілені по всій ділянці під землею та доставляють воду безпосередньо до кореневої системи рослин.

Зазвичай таку систему використовують на ділянках, які не піддаються перекопуванню. Найоптимальнішим матеріалом будуть поліетиленові труби (з круглими або щілинними отворами). Які розташовуються на глибині 20-30 см. Відстань між двома сусідніми магістралями становить 40-90 см (залежить від індивідуальних особливостей зрошуваної культури і від типу ґрунту). Проміжок між отворами зволожувача дорівнює 20...40 см. Система

внутрішньогрунтового зрошення проблематична в плані експлуатації, тому мало хто вирішується її встановлювати на власній ділянці.

Однак він має низку переваг:

- Внутрішньогрунтовий полив дуже економічний, тому що вода надходить відразу до коріння.
- Він має мінімальний коефіцієнт випаровування;
- Повністю виключено утворення шкідливої кірки на ґрунті та забезпечений вільний доступ кисню до коріння;
- Немає потреби постійно пушити верхній шар.

До недоліків можна віднести:

- Відсутність зрошення надґрунтових частин рослин, яке покликане підвищувати їхню родючість;
- Не використовується на піщаних ґрунтах;
- Трудомісткість процесу та витрати, пов'язані з придбанням і монтажем обладнання.

Одна з найголовніших деталей системи автоматичного поливу – контролер. Безперечно, він є мозком всієї системи. Насправді вся система поливу працює завдяки контролеру. А саме він відповідно до заданої програми, керуватиме пристроями та регулюватиме кількість поливів.

Зокрема, контролери з цифровим інтерфейсом спрощують процес програмування поливом і мають невеликі габарити. Вони призначені для роботи із системами поливу різних конфігурацій.

Насамперед, до корисних функцій цифрових мікроконтролерів відносяться:

- наявність різноманітних програм запуску систем поливу;
- застосування різних графіків роботи з урахуванням сезону;
- регулювання та обмеження тривалості поливу із забезпеченням затримок між включеннями різних режимів;

- можливість введення та зберігання параметрів запрограмованого ручного режиму роботи у пам'яті контролера;
- встановлення та збереження налаштувань програми при використанні додаткового живлення від батарей;
- прописаний алгоритм дій у разі пропадання електричного живлення;
- можливості підключення зовнішніх датчиків популярних виробників, включаючи моделі бездротового керування датчиками морозу та дощу;
- вбудовану діагностику електричних з'єднань.

Таким чином, основним завданням автоматизованої системи поливу є забезпечення рослин необхідною їм кількістю води з урахуванням атмосферних опадів, що реально випали. Для її виконання встановлюється в контрольному місці ґрунту датчик вологості. Тим самим він постійно аналізує наявність вологи в ґрунті і видає інформацію на контролер. Отримана інформація автоматично обробляється контролером і далі він автоматично регулює тривалість та обсяг подачі води.

## **2.2 Розробка й виготовлення моделі гідропонної установки на базі Arduino.**

Поставлені вимоги до проєктованої моделі установки для поливу рослин:

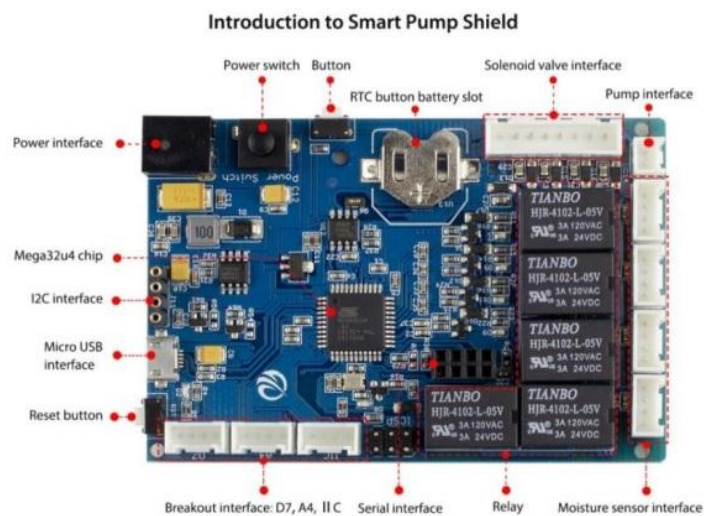
1. Вбудована плата керування (Arduino Leonardo). Не потрібне зовнішнє втручання.
2. Водяний насос із дуже низьким рівнем шуму. Менше 65 дБ під час роботи порівняно з 70 дБ при роботі настільного вентилятору.
3. Довговічні датчики вологості ґрунту, які можуть постійно перевіряти вологість ґрунту.

4. Контроль поливу до 4 рослин.
6. Один блок живлення для живлення всіх компонентів.
7. 1,3-дюймовий OLED-екран, що має відображати відсоток вологості для кожної рослини.
8. Годинник реального часу та резервна батарея.

*Наобхідні компоненти та їх призначення (Додаток А):*

1. Плата керування.

Рис. 2.6



Це мозок комплекту. Він зчитує інформацію з датчиків вологості та вирішує, яка рослина потребує води, і активує її компоненти для поливу рослини.

2. Блок живлення на 12 В.

Рис. 2.7



Він забезпечує живлення всіх компонентів комплекту.

3. Водяний насос з низьким рівнем шуму.

Рис. 2.8



Забирає воду з спеціальної їмності і подає через випускний клапан.

4. П'ятиходова водопровідна трубка.

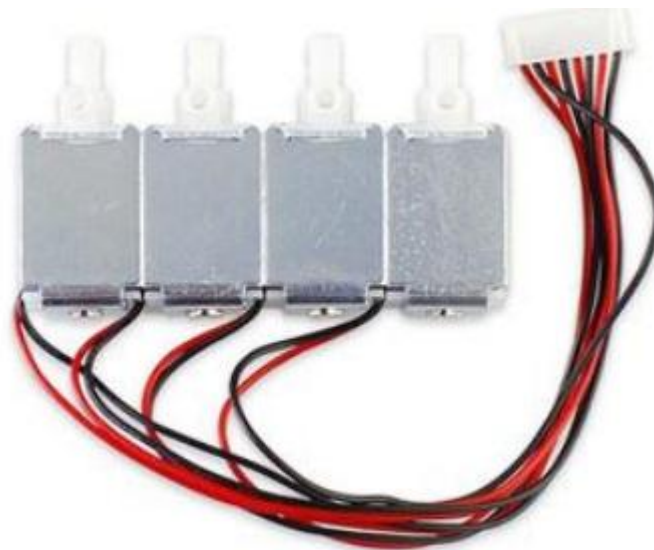
Рис. 2.9



Дозволяє розподіляти воду, що надходить із вихідного клапана водяного насоса, на чотири виходи. Цей компонент у поєднанні з чотириходовим клапаном дозволяє поливати окремі рослини.

5. Чотириходовий водяний клапан.

Рис. 2.10



Кожен клапан має вхід і вихід. Вхід бере воду з єдиного виходу води з п'ятиходової водопровідної труби, але дозволяє воді проходити через вихідний клапан лише за командою керуючої плати. Це забезпечує контрольований розподіл води, дозволяючи розподілити воду між рослинами, які її потребують уникаючи поливу рослин, яким це потрібно, водночас економлячи воду в процесі.

#### 6. Пластикова водопровідна трубка.

Рис. 2.11



Прозора пластикова водопровідна трубка. 5 метрів діаметром 5 мм.

#### 7. Датчики вологості ґрунту.

Рис. 2.12



Ці датчики вимірюють вологість у ґрунті та на основі їх показань керуюча плата вирішує, потрібна рослині вода чи ні.

## 8. Дроти.

Рис. 2.13



Кожен кабель має довжину 1 метр. Ці кабелі під'єднані до датчиків вологості ґрунту на одному кінці та до керуючої плати на іншому кінці.

### *Огляд роботи системи.*

Після завантаження програми в контролер (ArduinoLeonardo), і з'єднання всіх компонентів, слід підключити живлення до моделі. Після підключення живлення контролер ініціалізує дисплей, зчитує інформацію з датчиків вологості ґрунту та визначатиме вміст вологи в ґрунті для кожної з рослин. Якщо вміст вологи в ґрунті будь-якої з рослин становить менше 20% (можна встановити будь-яку), контролер ініціює процес поливу відповідних рослин.

Етапи процесу поливу такі:

а. Контролер запускає реле, що контролює живлення водяного насоса. Це активує водяний насос, надсилаючи 12 вольт постійного струму від джерела живлення до водяного насоса.

б. Водяний насос забирає воду з джерела у вхідний клапан і подає її з легким тиском до вихідного клапана.

в. Через вихідний клапан вода надходить у вхідний клапан п'ятиходової водопровідної труби та проходить через вихідні клапани, які з'єднані з чотирьохходовим клапаном.



г. Контролер також відкриває відповідний вихід у чотириходовому клапані, що відповідає датчикам, які повідомляють про низьку вологість ґрунту. Таким чином рослини, які потребують поливу, отримують воду.

д. Контролер продовжуватиме поливати рослини, поки не виявить, що вміст вологи в ґрунті досягне 20%.

е. Коли вміст вологи досягне потрібного рівня, контролер зупинить процес поливу, відправивши сигнал на відповідні реле.

є. Контролер відновить процес моніторингу.

### *Установка та налаштування Arduino IDE*

#### 1. Завантаження Arduino IDE

Необхідно використовувати останню версію Arduino IDE для вашої комп'ютерної системи. Посилання на останню версію (на момент створення цієї кваліфікаційної роботи) Windows (XP і вище):

[https://www.arduino.cc/download\\_handler.php?f=/arduino-1.8.9-windows.exe](https://www.arduino.cc/download_handler.php?f=/arduino-1.8.9-windows.exe)

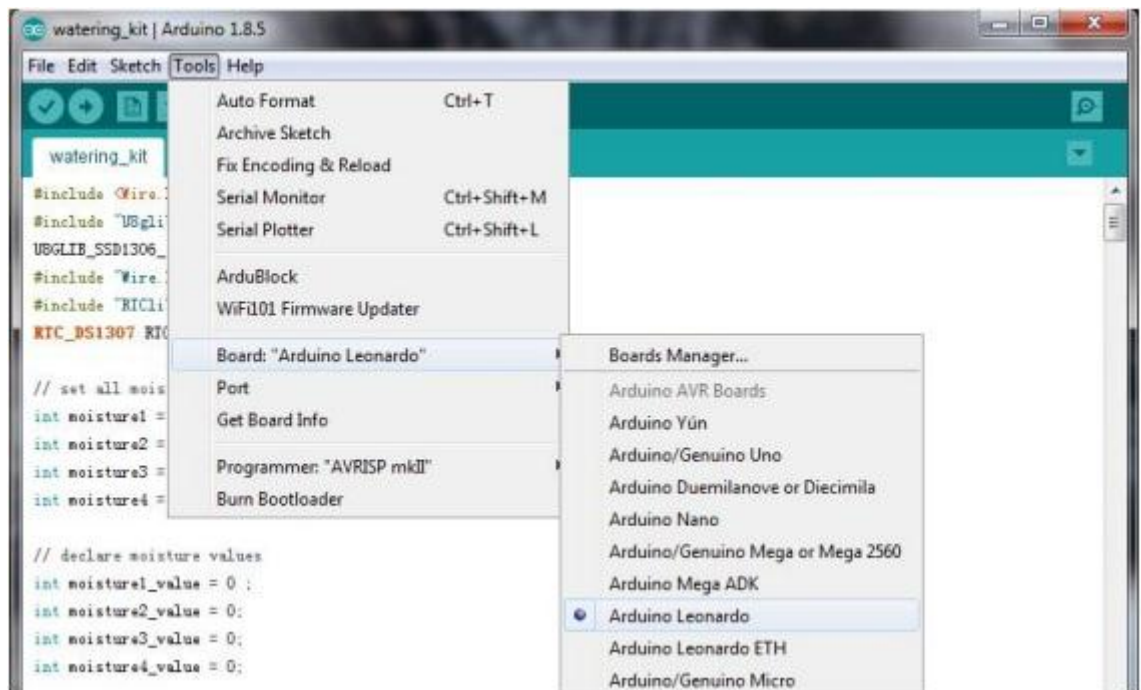
#### 2. Встановлення Arduino IDE

Користувачі Windows, мають дотримуватись інструкцій зі встановлення.

3. Налаштування плати контролера, яка використовуватиметься в Arduino IDE

У верхній частині Arduino IDE натисніть «Інструменти> Плата>» та виберіть «ArduinoLeonardo» з доступних варіантів.

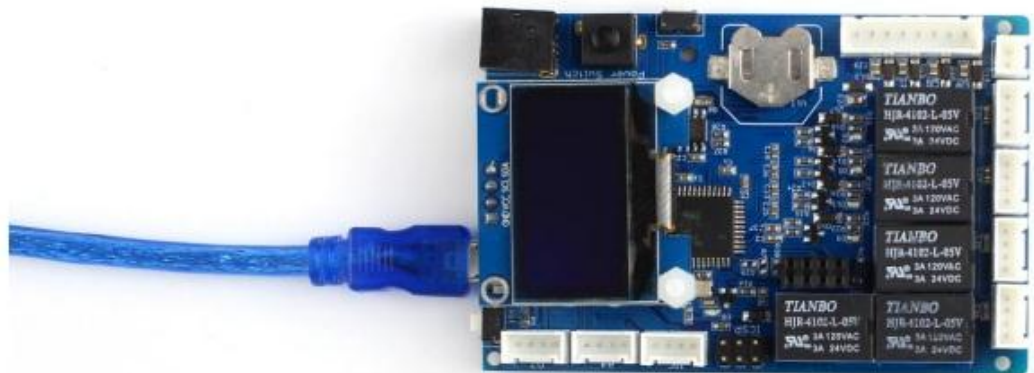
Рис. 2.14



4. Підключення керуючої плати до комп'ютерної системи.

– Приєднайте кабель micro-usb до плати Smart Pump Board.

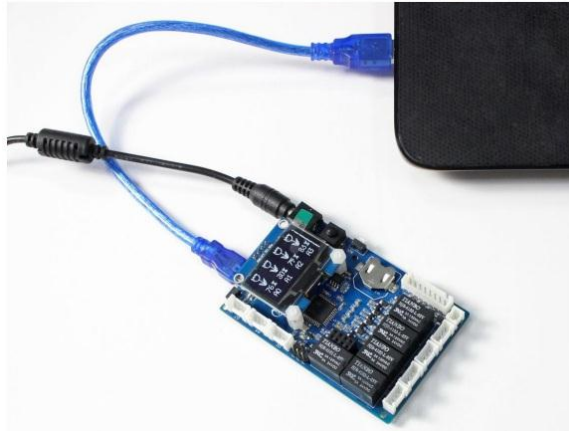
Рис. 2.15



– Під'єднайте інший кінець USB-кабелю до USB-порту комп'ютера.

(Примітка: під час завантаження коду живіть плату від джерела постійного струму)

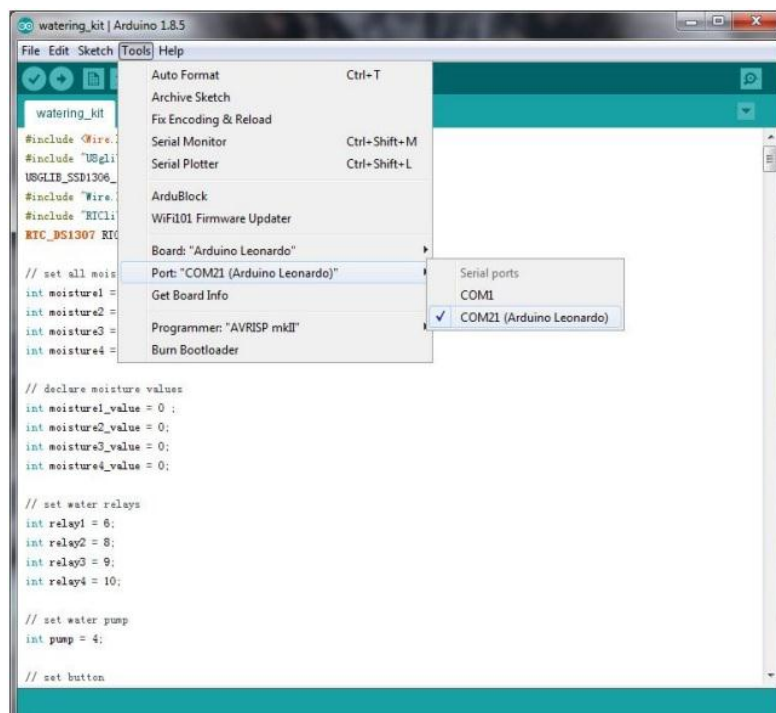
Рис. 2.16



### 5. Вибір комунікаційного порту.

Arduino IDE має знати, до якого USB-порту комп'ютера підключено ArduinoLeonardocontroller, щоб з'єднатися з ним. Для цього виберіть «Інструменти>Порт» і виберіть COM-порт, який вказує на ArduinoLeonardo. Зауважте, що фактичні цифри після слова «COM» відрізнятимуться від комп'ютера до комп'ютера, тому вони можуть відрізнятися від показаних на малюнку.

Рис. 2.17



6. Після завершення необхідно завантажити програму в ArduinoIDE та запрограмувати контролер Arduino Leonard на платі Smart Pump Board.

### *Завантаження програми в Smart Pump Board*

1. Завантажте бібліотеку:

[www.elecrow.com/download/Watering\\_kit\\_library.zip](http://www.elecrow.com/download/Watering_kit_library.zip) і розпакуйте її.

Скопіюйте його до бібліотек файлів у файлі IDE arduino, тобто D:\Program Files (x86)\Arduino\libraries.

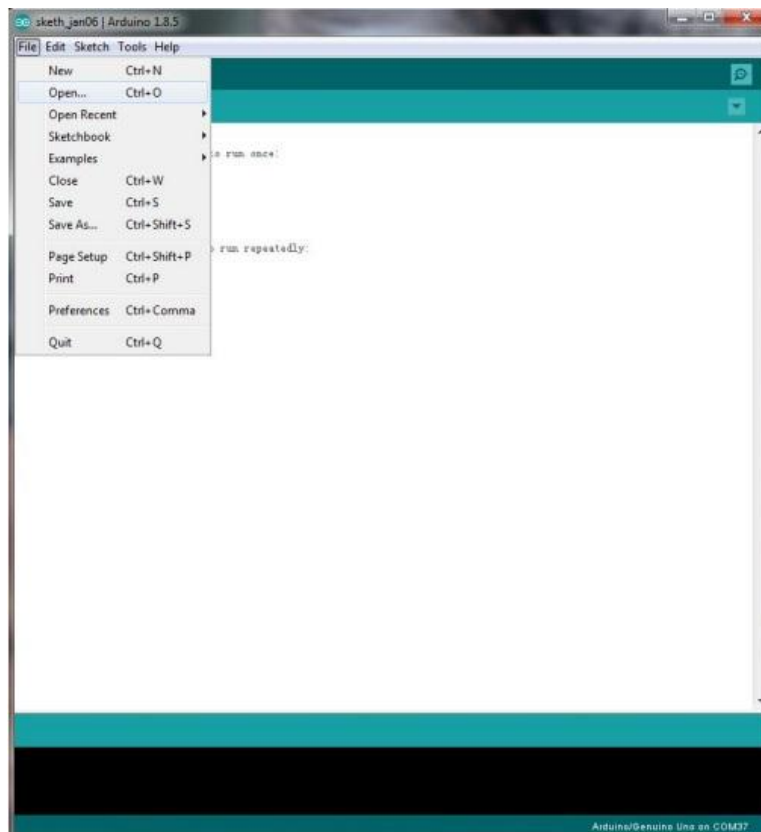
2. Завантажте програму за адресою:

[https://www.elecrow.com/download/watering\\_kit.zip](https://www.elecrow.com/download/watering_kit.zip)

3. Розпакуйте файл із zip-файлу. Ім'я файлу watering\_kit.ino

4. Завантажте програму в Arduino IDE.

Рис. 2.18



5. Завантажте код на плату Arduino.

Рис. 2.20

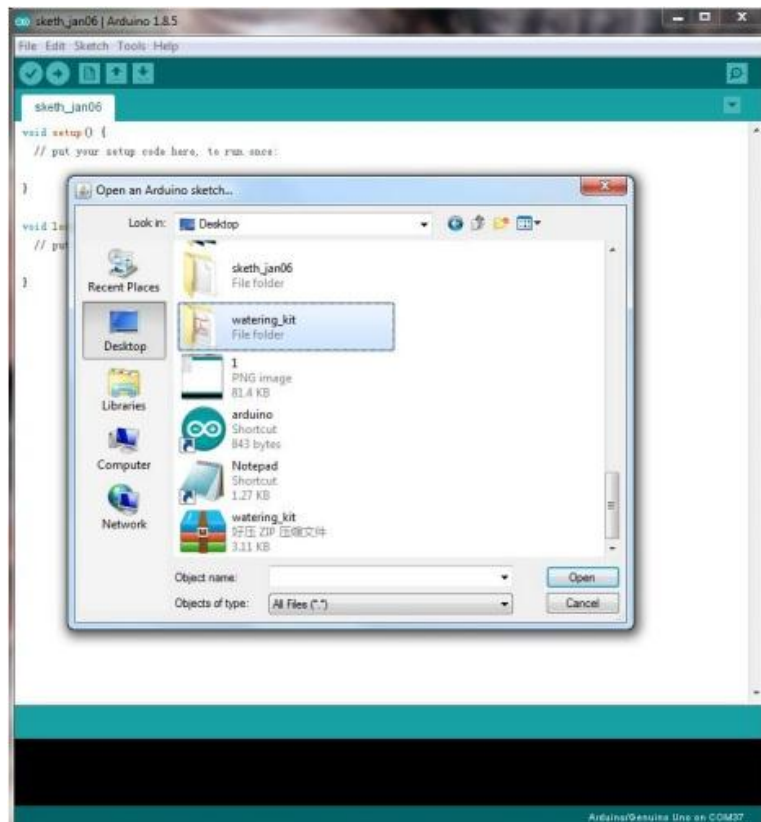


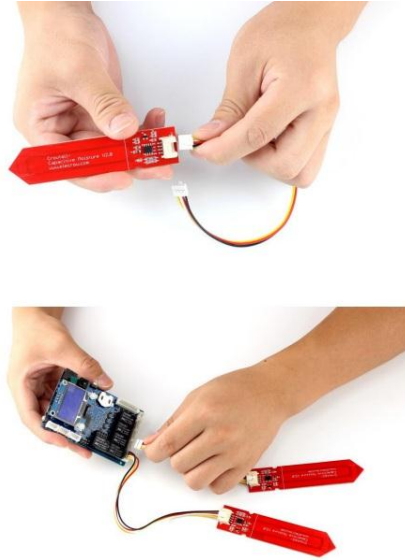
Рис. 2.21



*Складання компонентів.*

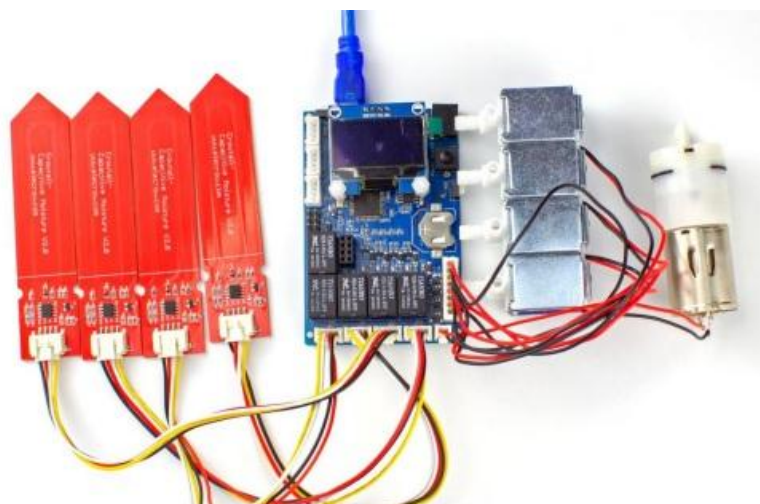
1. Приєднайте кабель до датчиків вологості ґрунту, як показано на малюнку. Повторіть для кожного датчика.
2. Підключіть датчики вологості ґрунту до плати розумного насоса.

Рис. 2.22



3. Підключіть водяний насос і чотириходовий клапан до Smart PumpBoard.

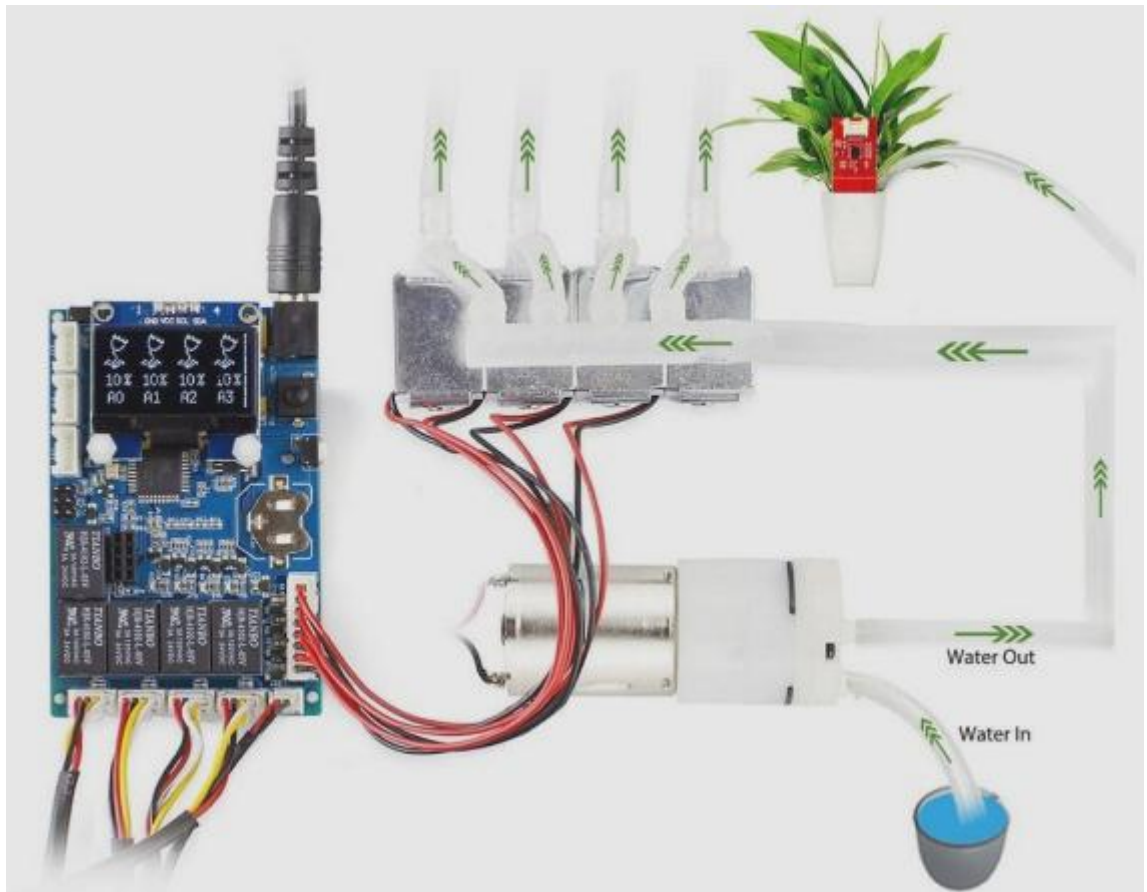
Рис. 2.23



4. Розріжте пластикову водопровідну трубу

Перший зріз: від джерела води до входу водяного насоса (залежить від висоти джерела води). Другий розріз: вихід водяного насоса на вхід п'ятиходового клапана. (залежить від того, як ви розташуєте рослини). Третій розріз: 4 частини для випускних клапанів у п'ятиходовому клапані до входу чотириходового водяного клапана. (залежить від того, як ви розмістили рослини). Четвертий розріз: 4 штуки для випускних клапанів чотириходових водяних клапанів у ґрунт, де висаджується кожна рослина. (залежить від того, як ви розмістили рослини).

Рис. 2.24

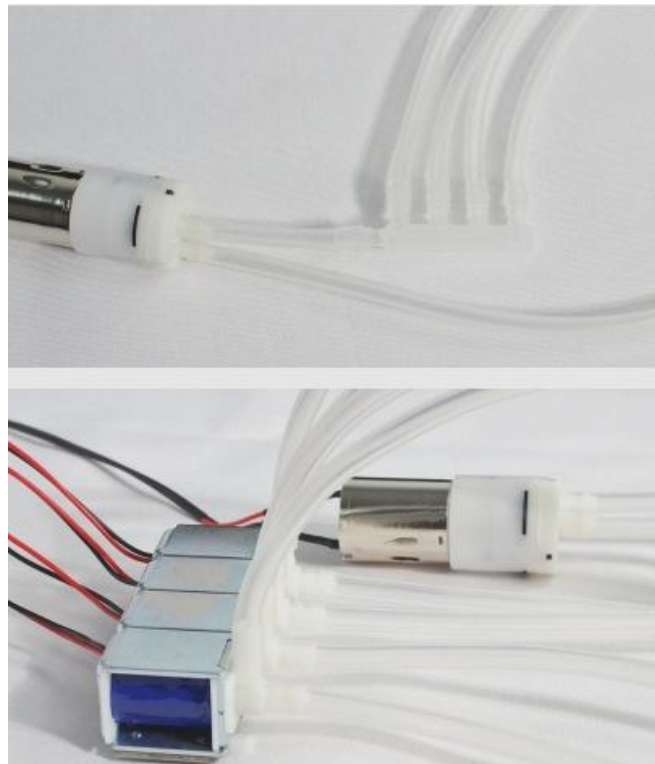


##### 5. Під'єднайте водопровідні труби

Відріжте водопровід і розділіть їх на 6 частин. Переконайтеся, що труба, що йде всередину джерела води, буде достатньо довгою, щоб дістатися до води. Ми під'єднаємо 2 труби до насоса, одна всмоктує воду з джерела води, а інша з'єднується з п'ятиходовою водопровідною трубою.

Підключіть п'ять труб до вентилів.

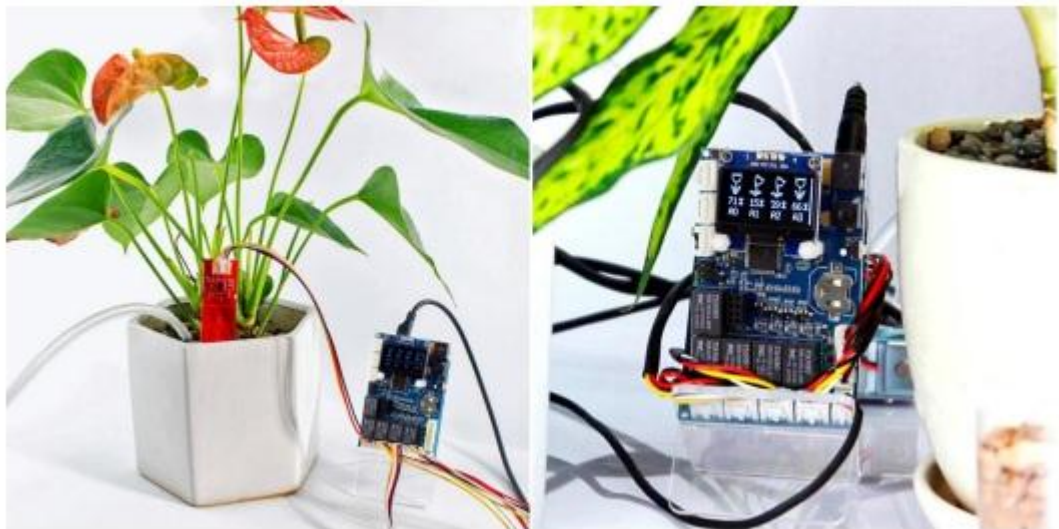
Рис. 2.25



#### 6. Підключення живлення.

Необхідно підключити до плати одне джерело живлення 12 В. Таким чином, Leonardo отримуватиме 5 В, а насос і перемикач отримуватимуть 12 В.

Рис. 2.26





### **3. МЕТОДИКА ПРОВЕДЕННЯ УРОКІВ ТЕХНОЛОГІЙ З ВИКОРИСТАННЯМ ARDUINO**

#### **3.1 Передумови вивчення мікроконтролерів у профільній школі.**

У 50-х ... 90-х роках минулого сторіччя, коли почалася ера бурхливого розвитку систем автоматизації, для автоматичного управління технологічними процесами та установками були потрібні пристрої, які могли здійснювати підтримку технологічних параметрів на заданих значеннях. Такі пристрої назвали регуляторами. Регулятори приймали від первинного перетворювача поточні значення параметра, виробляли аналогову обробку цього сигналу і видавали сигнал керування на виконавчий механізм.[15]

Регулятори можна поділити на регулятори прямої дії (керують виконавчим механізмом за рахунок енергії, що отримується від регульованого середовища) та регулятори непрямої дії (керують виконавчим механізмом за рахунок енергії, що отримується від стороннього джерела). За видом енергії регулятори непрямої дії поділяються на електричні, пневматичні та гідравлічні. Треба сказати, що регулятори прямої дії не мають до нас із вами жодного відношення. Їх підбирають та замовляють у своїх розділах проекту проектант-технологи (ОВ, ВК, ХС, ТМ тощо).[15]

Найбільш поширеним видом регуляторів у схемах автоматизації технологічних процесів у різних галузях промисловості були електричні регулятори. В інженерних системах будівель тільки вони й застосовувалися.

Наприкінці 80-х з'являються пристрої, обробка інформації у яких виконується у цифровому вигляді процесорами. Починається епоха мікропроцесорних контролерів. Хоча й нині поряд фірм випускаються аналогові регулятори.

Мікропроцесорні контролери не лише замінили аналогові регулятори. Їхні можливості набагато ширші. У схемах автоматизації вони виконують як

функції регулювання, а й замінюють релейні схеми управління і сигналізації. Контролери можуть відображати параметри індикаторів і створювати архіви параметрів, тобто. виконувати функції вторинних приладів, що показують і реєструють. А ще вони можуть вести журнали подій (фіксація несправностей обладнання, його включення та відключення, відхилення параметрів від заданих значень тощо), які до появи контролерів заповнювалися операторами технологічних процесів вручну.

Спектр мікропроцесорних контролерів, що випускаються в даний час, дуже великий.

Їх можна класифікувати за:

- типом вхідних та вихідних сигналів;
- відсутністю чи наявністю інтерфейсних виходів;
- типом інтерфейсних виходів;
- відсутністю чи наявністю індикаторів;
- типом індикаторів;
- гнучкістю програмного забезпечення та способів програмування;
- конструктивним виконанням.

За типом вхідних та вихідних сигналів контролери можна розділити на:

- контролери з аналоговими входами/виходами;
- контролери з дискретними входами/виходами;
- контролери з дискретними та аналоговими входами/виходами;
- контролери лише з входами, дискретними та/або аналоговими.

Останні обов'язково мають інтерфейсний вихід та є різновидом контролерів, які називаються пристроями зв'язку з об'єктом.

Установкою контролерів без інтерфейсних виходів вирішуються локальні завдання автоматизації. Контролери з інтерфейсними виходами та пристрої зв'язку з об'єктом працюють як з підключеними до них периферійними пристроями, так і з іншими контролерами, об'єднаними з

ними в інтерфейсну мережу та обладнанням верхнього рівня систем диспетчеризації.

Контролери можуть мати один чи кілька різних інтерфейсних виходів. Інтерфейси характеризуються середовищем передачі та протоколом обміну. Середовищем передачі можуть бути кручена пара, Ethernet, радіоканал, GSM-канал, телефонний канал та ін. Протоколи обміну поділяються на «відкриті» та «закриті». «Відкриті» протоколи стандартизовані і будь-який виробник контролерів може їх застосувати. До них можна зарахувати MODBUS-RTU, LonWorks, BACnet, KNX і т.д. «Закриті» протоколи пишуть виробники обладнання для своїх контролерів і не віддають їх опис іншим виробникам.

На лицьовій панелі контролера може бути встановлений індикатор для відображення поточних параметрів, станів обладнання та налаштувань, а також для роботи з журналами подій та архівами параметрів. Індикатори можуть бути буквено-цифровими та графічними. Графічні індикатори бувають чорно-білими та кольоровими. Наявність графічного індикатора дозволяє для зручності користувача створювати на екрані контролера мнемосхеми.

За типом програмного забезпечення контролери поділяються на «жорстко» запрограмовані, конфігуровані та вільно програмовані.

«Жорстко» запрограмований контролер призначений для виконання певного завдання регулювання та/або управління. Одна й та програма записується у всі контролери даної модифікації підприємством-виробником. При установці на об'єкті користувачем можуть бути змінені заводські налаштування регуляторів та уставки параметрів.

Конфігуровані контролери мають у своєму програмному забезпеченні певний набір технологічних систем, для управління якими може застосовуватися. Для кожної технологічної системи у програмі закладено набір алгоритмів управління та регулювання. Під час встановлення на об'єкті

користувач вибирає свою конфігурацію в програмному забезпеченні, задає налаштування та уставки параметрів.

У вільно програмовані контролери може бути записана будь-яка програма, розроблена користувачем. Обмеження накладаються кількістю та типом входів/виходів, а також обсягом пам'яті.

Налаштування регуляторів, уставки параметрів, конфігурування та програмування контролерів, що мають індикатор, може бути зроблено за допомогою органів управління на лицьовій панелі контролера. Контролери, що жорстко програмуються, можуть також програмуватися за допомогою програмних карт. Програми керування записуються підприємством-виробником на інформаційній карті. Контролер функціонує лише за наявності вставленої карти.

Програми для вільно програмованих контролерів можуть створюватися користувачем на комп'ютері за допомогою спеціальних прикладних програм (середовищ), а потім переноситься в контролер. Перенесення програм у контролер може здійснюватися підключенням контролера до комп'ютера через спеціальний пристрій – програматор або переноситись у контролер на flash-карті.

Конструктивно контролери можуть бути призначені для встановлення на фасаді щитів та пультів, для встановлення всередині щитів на DIN-рейку, для встановлення на стіні або устаткуванні, для встановлення у спеціальні стійки. Конструктивно контролери, що встановлюються на DIN-рейку або спеціальні стійки, можуть бути набірними, тобто. складаються з базових та додаткових модулів.

Таким чином вивчення у профільній школі мікроконтролерів і пристроїв, що працюють на мікроконтролерах на сьогодні є необхідною умовою розвитку в учнів інженерного мислення, і частиною їх базової підготовки для здобуття вищої інженерної освіти.

### **3.2 Методика застосування Arduino на уроках технології у профільній школі.**

Розробка плану-конспекту уроку технології на **тему:** Підключення потенціометра до Ардуїно.

**Тип уроку:** комбінований.

**Мета уроку:**

**Навчальна:** познайомитися як підключається потенціометр до Ардуїно, навчити застосовувати отримані знання на практиці - знаходити необхідні деталі для складання схеми.

**Розвиваюча:** розвиток уважності, пам'яті, дрібної моторики рук учнів, розвиток вміння виділяти головне у завданні, розвиток навичок колективної роботи та взаємодопомоги.

**Виховна:** виховання інформаційної культури учнів, акуратності у роботі.

**Методи навчання:** пояснювально-ілюстративний, наочний, частково-пошуковий, дослідницький.

Очікувані результати:

**Учні повинні знати/розуміти:**

- підключення потенціометра;
- способи маркування;
- як збирається схема, та пишеться скеч;
- заходи безпеки під час роботи з устаткуванням.

**Учні повинні вміти:**

- правильно та швидко знайти потрібні деталі роботи над проектом;
- встановлювати деталі на макетній платі згідно з вимогами;
- правильно написати програму.

**Обладнання:** комп'ютер, проектор, електронні деталі, моделі гідропонної установки на базі Arduino.

**План заняття:**

- I. Організаційний момент. Актуалізація знань (3 хв).
- II. Теоретична частина (12 хв).
- III. Практична частина (20 хв).
- IV. Запитання учнів. Підбиття підсумків уроку (3 хв).

**Хід уроку:**

I. Теоретична частина.

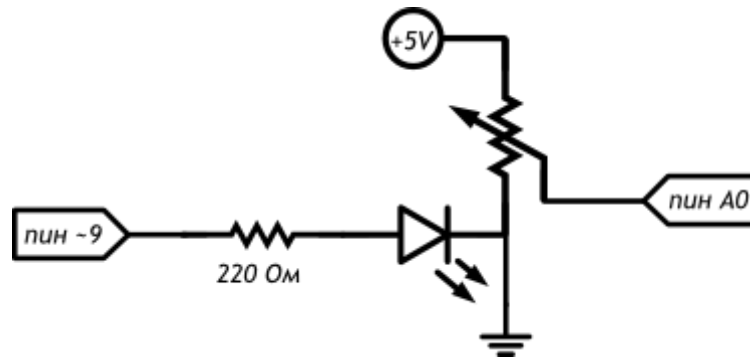
У робототехніці регулювання різних параметрів, таких як гучність звуку, потужність, напруга і т.д. здійснюється за допомогою змінних резисторів з регульованим рівнем опору. Прикладом такого пристрою є ардуїно потенціометр, який при включенні в електричну схему може бути використаний для регулювання параметрів.

**Принцип роботи потенціометра**

Змінний резистор або потенціометр – це електричний пристрій, значення рівня опору якого можна задати у певних межах. Таким чином, ми можемо змінювати параметри електричних схем, гнучко підлаштовуючи їх під певні умови: наприклад, регулювати чутливість датчика або гучність звуку в динаміці. Потенціометри набули широкого поширення в схемах регулювання гучності, напруги, контрастності і т.д., за свою простоту та практичність.

**Підключення потенціометра до плат Ардуїно.**

Підключення потенціометра до ардуїно виконується відповідно до схеми, представленої на малюнку:



Для цього три виводи потенціометра необхідно з'єднати із зазначеними виводами плати:

Чорний – GND;

Червоний – живлення 5В;

Середній – від центрального виведення до аналогового входу А0.

Змінюючи положення валу підключеного потенціометра, відбувається зміна параметра опору, що викликає зміну показника на нульовому піні плати ардуїно. Зчитування отриманого значення напруги аналогового імпульсу відбувається у скетчі за допомогою команди `analogRead()`.

У плату Ардуїно вбудований аналого-цифровий перетворювач, здатний зчитувати напругу і переводити його в цифрові показники зі значенням від нуля до 1023. При повороті покажчика до кінцевого значення в одному з двох можливих напрямків, напруга на піні дорівнює нулю, і, отже, напруга яке генеруватиметься становить 0 В. При повороті валу остаточно в протилежному напрямі на пін надходить напруга величиною 5В, отже числове значення становитиме 1023.

Приклад скетчу:

```
#define PIN_LED 11
#define PIN_POT A0

void setup()
{
  // Пін, до якого приєднується світлодіод визначаємо як вихід
  pinMode(PIN_LED, OUTPUT);
```

```

// Пін із змінним резистором є входом
pinMode(PIN_POT, INPUT);
}

void loop(){
// Визначаємо 2 змінні типи int
int rotat, brightn;

// Зчитування в змінну напругу rotat з змінного резистора:
// мікроконтролер видаватиме числа від 0 до 1023
// пропорційні положенню повороту валу
rotat = analogRead(PIN_POT);

// Перетворимо значення на яскравість. Для цього ділимо rotat на 4, що з
// урахуванням округлення дасть нам число від 0 до 255. Саме це число ми
// подамо на шим-вихід, за допомогою якого можна керувати яскравістю.

brightn = rotat / 4;

// Запис шим значення яскравості на світлодіод
analogWrite(PIN_LED, brightn);
}

```

## II. Практична частина.

Почніть складання пропонованої схеми. На цьому етапі заняття вчитель допомагає, у разі потреби, окремим групам під час її складання. Потім учні виконують креслення зібраної електричної схеми.

## III. Підбиття підсумків уроку.

Перевірка виконаної роботию. Відповіді на питання учнів. Піведення підсумків уроку.



## ВИСНОВКИ

Виконана кваліфікаційна робота передбачала розробку й виготовлення моделі гідропонної установки на базі Arduino та методики її використання під час вивчення технології в профільній школі. За її результатами можна зробити наступні висновки:

1. Досліджено особливості програмно-апаратної платформи Arduino. З'ясовано історію створення програмно-апаратної платформи Arduino. Вона була випущена у 2005 році як скромний інструмент для студентів Банці в Інституті проектування взаємодій міста Івреа (Interaction Design Institute Ivrea, IDII), Arduino породив міжнародну революцію у сфері міжнародних електронних саморобок. Ви можете купити цю плату за \$30 або зібрати її з нуля. Усі схеми та вихідні коди доступні безкоштовно на умовах відкритих ліцензій. В результаті Arduino став найвпливовішою апаратною системою свого часу з відкритим кодом. Визначено складові апаратного і програмного забезпечення платформи Arduino.

2. Розроблено проєкт і виготовлена модель гідропонної установки на базі Arduino. Визначено особливості автоматизації поливу рослин з використанням мікроконтролерного керування. З'ясовано, що а) автоматизація поливу – це особливий технічний комплекс, який самостійно здатний забезпечити рівномірний та регулярний полив певної території; б) є три основні види системи поливу: дощувальний, краплинний та внутрішньогрунтове зрошення; в) одна з найголовніших деталей системи автоматичного поливу – контролер, вся система поливу працює завдяки контролеру.

3. Описана методика проведення уроків технологій з використанням Arduino. Розглянуті передумови вивчення мікроконтролерів у профільній школі. З'ясовано що вивчення у профільній школі мікроконтролерів і

пристроїв, що працюють на мікроконтролерах на сьогодні є необхідною умовою розвитку в учнів інженерного мислення, і частиною їх базової підготовки для здобуття вищої інженерної освіти. Розроблено план-конспект уроку технології на тему: Підключення потенціометра до Ардуїно.

Таким чином, можна зробити висновок, що використання моделі моделі гідропонної установки на базі Arduino під час вивчення технології в профільній школі може стати ефективним інструментом для розвитку потенціалу дітей у різних галузях та сприяти створенню умов для їхнього творчого та політехнічного розвитку.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Автоматичний полив домашніх рослин: догляд за рослинами з легкістю [Електронний ресурс]. Avtopoliv. Режим доступу: <https://avtopoliv.com.ua/avtomatychnyi-polyv-domashnikh-roslyn-dohliad-za-roslynamy-z-lehkistiu/> (дата звернення: 20.11.2024)
2. Автоматизація поливу [Електронний ресурс]. Ар-п. Режим доступу: <https://ar-p.com/avtomatizacija-poliva/> (дата звернення: 20.11.2024)
3. Апаратна частина платформи Arduino [Електронний ресурс]. Arduino. Режим доступу: <https://arduino.com/Hardware> (дата звернення: 20.11.2024)
4. Биков В. Ю. Інформаційні технології і засоби навчання. Київ: Атіка, 2008. 684 с.
5. Вакуленко І. В. Управління самостійною роботою студентів з використанням інформаційно-комунікаційних технологій. *Науковий часопис Національного педагогічного університету імені М. П. Драгоманова. Серія 2: Комп'ютерно-орієнтовані системи навчання*. Київ. 2016. Вип. 18 (25). С. 50–64.
6. Історія створення Ардуіно [Електронний ресурс]. Arduino. Режим доступу: <https://arduino.ua/art2-istoriya-sozdaniya-arduino> (дата звернення: 20.11.2024)
7. Кривонос О. М. Огляд платформи Arduino Nano 3.0 та перспективи використання під час навчального процесу. Інформаційні технології і засоби навчання. Том 56. № 6. Київ. 2016. С.79–80.
8. Лещук С. О. Навчально-інформаційне середовище як засіб організації пізнавальної діяльності учнів. *Науковий часопис Національного педагогічного університету ім. М. П. Драгоманова. Серія 2: Комп'ютерно-орієнтовані системи навчання: До 170-річного ювілею*. Київ: НПУ, 2004. С. 305–313.

9. Онлайн курс Arduino на базі простого стартового набору [Електронний ресурс]. Geektimes. Режим доступу : <https://geektimes.com/post/279860/> (дата звернення: 20.11.2024).
10. Ортинський В. Л. Педагогіка вищої школи: навч. посіб. для студ. вищ. навч. закл. Київ: Центр учбової літератури, 2009. 472 с.
11. Остапчук С.А. До проблеми використання платформи Arduino у вивченні робототехніки. Наукові записки ЦДПУ. Серія: Педагогічні науки. Кропивницький: РВВ ЦДПУ ім. В. Винниченка, 2018. Вип. 168. С. 178-181.
12. Петрицин І. О. Застосування комп'ютерного моделювання у процесі електротехнічної підготовки майбутнього вчителя технологій. *Молодь і ринок*. 2017. № 1. С. 60–64.
13. Посібник до Arduino Leonardo та Micro [Електронний ресурс]. Arduino. Режим доступу: <https://doc.arduino.ua/guide/ArduinoLeonardoMicro> (дата звернення: 20.11.2024)
14. Програмування Arduino [Електронний ресурс]. Arduino. Режим доступу: [arduino.com/Reference](http://arduino.com/Reference) (дата звернення: 20.11.2024).
15. Регулятори і контролери [Електронний ресурс]. Raut-automatic. Режим доступу: <http://www.raut-automatic.kiev.ua/school/?p=27> (дата звернення: 20.11.2024)
16. Умрик М. А. Організація самостійної роботи майбутніх учителів засобами інформаційно-навчального середовища. *Проблеми сучасної педагогічної освіти. Сер. : Педагогіка і психологія: зб. статей*. Ялта: РВВ КГУ, 2007. Вип.16. Ч. 1. С. 88–95.
17. Arduino Products[Електронний ресурс]. Режим доступу: <https://www.arduino.cc/en/Main/Products> (дата звернення: 20.11.2024).

18. Kushner D. The Making of Arduino [Електронний ресурс]. Режим доступу: <https://web.eecs.umich.edu/~prabal/teaching/resources/eecs582/kushner11arduino.pdf> (дата звернення: 20.11.2024).
19. Margolis Michael. Arduino Cookbook. 2nd Edition. O'Reilly, 2011. 724 p.
20. Microcontroller [Електронний ресурс]. Режим доступу: <https://www.techopedia.com/definition/3641/microcontroller> (дата звернення: 20.11.2024)
21. Nano Плати Ардуіно [Електронний ресурс]. Режим доступу: <http://arduino.ua/hardware/Nano> (дата звернення: 20.11.2024)

## ДОДАТКИ

## Додаток А

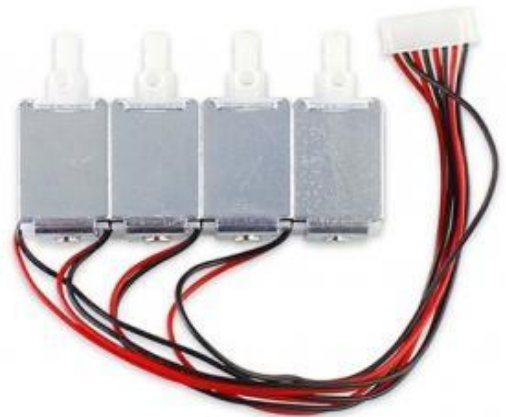
Проект: Модель гідропонної установки на базі Arduino

Перелік компонентів і обладнання:

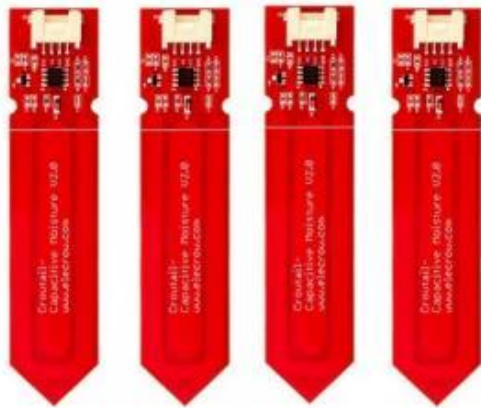




Crowtail-Smart Pump Shield\*1



DC 12V Four-way Valve



Crowtail-Capacitive Moisture Sensors\*4



Water Pump\*1



USB Cable\*1



Water Pipe\*1



Five-Way Pipe\*1



1m 4 Pin Crowtail Cable\*4



DC 12V Adapter\*1

Проект: Модель гідропонної установки на базі Arduino  
Розташування датчиків вологості ґрунту.





## Проект: Модель гідропонної установки на базі Arduino

## Тестовий код:

```

#include <Servo.h>
int pinLB=14;    // Define a 14 Pin
int pinLF=15;    // Define a 15 Pin

int pinRB=16;    // Define a 16 Pin
int pinRF=17;    // Define a 17 Pin

int MotorLPWM=5; //Define a 5 Pin
int MotorRPWM=6; //Define a 6 Pin

int inputPin = 9; // Define the ultrasound signal receiving a Pin
int outputPin =8; //Define the ultrasound signal emission Pin

int Fspeedd = 0;    // go
int Rspeedd = 0;    // The right to
int Lspeedd = 0;    // Turn left to
int directionn = 0; // After the former = 8 = 2 left = 4 right = 6
Servo myservo;     // Set up the myservo
int delay_time = 250; // After the servo motor to the stability of the
time

int Fgo = 8;        // go
int Rgo = 6;        // The right to
int Lgo = 4;        // Turn left to
int Bgo = 2;        // astern

void setup()
{
  Serial.begin(9600);    // Initialize
  pinMode(pinLB,OUTPUT); // Define 14 pin for the output (PWM)
  pinMode(pinLF,OUTPUT); // Define 15 pin for the output (PWM)
  pinMode(pinRB,OUTPUT); // Define 16 pin for the output (PWM)
  pinMode(pinRF,OUTPUT); // Define 17 pin for the output (PWM)

  pinMode(MotorLPWM, OUTPUT); // Define 5 pin for PWM output
  pinMode(MotorRPWM, OUTPUT); // Define 6 pin for PWM output

  pinMode(inputPin, INPUT);    // Define the ultrasound enter pin
  pinMode(outputPin, OUTPUT);  // Define the ultrasound output pin

  myservo.attach(11);    // Define the servo motor output 10 pin(PWM)
}
void advance(int a)    // go
{
  digitalWrite(pinRB,HIGH); // 16 feet for high level
  digitalWrite(pinRF,LOW);  //17 feet for low level
  analogWrite(MotorRPWM,220); //Set the output speed(PWM)
  digitalWrite(pinLB,HIGH); // 14 feet for high level
  digitalWrite(pinLF,LOW);  //15 feet for high level
  analogWrite(MotorLPWM,220); //Set the output speed(PWM)
}

```

```

    delay(a * 1);
}

void right(int b)          //right
{
    digitalWrite(pinRB,LOW);
    digitalWrite(pinRF,HIGH);
    analogWrite(MotorRPWM,220);
    digitalWrite(pinLB,LOW);
    digitalWrite(pinLF,LOW);
    delay(b * 100);
}

void left(int c)          //left
{
    digitalWrite(pinRB,LOW);
    digitalWrite(pinRF,LOW);
    digitalWrite(pinLB,LOW);
    digitalWrite(pinLF,HIGH);
    analogWrite(MotorLPWM,220);
    delay(c * 100);
}

void turnR(int d)         //right
{
    digitalWrite(pinRB,HIGH);
    digitalWrite(pinRF,LOW);
    analogWrite(MotorRPWM,220);
    digitalWrite(pinLB,LOW);
    digitalWrite(pinLF,HIGH);
    analogWrite(MotorLPWM,220);
    delay(d * 50);
}

void turnL(int e)        //left
{
    digitalWrite(pinRB,LOW);
    digitalWrite(pinRF,HIGH);
    analogWrite(MotorRPWM,220);
    digitalWrite(pinLB,HIGH);
    digitalWrite(pinLF,LOW);
    analogWrite(MotorLPWM,220);
    delay(e * 50);
}

void stopp(int f)        //stop
{
    digitalWrite(pinRB,LOW);
    digitalWrite(pinRF,LOW);
    digitalWrite(pinLB,LOW);
    digitalWrite(pinLF,LOW);
    delay(f * 100);
}

void back(int g)         //back
{
    digitalWrite(pinRB,LOW);
    digitalWrite(pinRF,HIGH);
    analogWrite(MotorRPWM,220);
    digitalWrite(pinLB,LOW);

```

```

    digitalWrite(pinLF,HIGH);
    analogWrite(MotorLPWM,220);
    delay(g * 100);
}

void detection()          //Measuring three angles(0.90.179)
{
    int delay_time = 200; // After the servo motor to the stability of
the time
    ask_pin_F();          // Read in front of the distance

    if(Fspeedd < 10)     // If the front distance less than 10 cm
    {
        stopp(1);        // Remove the output data
        back(2);         // The back two milliseconds
    }

    if(Fspeedd < 25)     // If the front distance less than 25 cm
    {
        stopp(1);        // Remove the output data
        ask_pin_L();     // Read the left distance
        delay(delay_time); // Waiting for the servo motor is stable
        ask_pin_R();     // Read the right distance
        delay(delay_time); // Waiting for the servo motor is stable

        if(Lspeedd > Rspeedd) //If the distance is greater than the
right distance on the left
        {
            directionn = Lgo; //Left
        }

        if(Lspeedd <= Rspeedd) //If the distance is less than or equal
to the distance on the right
        {
            directionn = Rgo; //right
        }

        if (Lspeedd < 15 && Rspeedd < 15) /*If the left front distance
and distance and the right distance is less than 15 cm */
        {
            directionn = Bgo; //Walk backwards
        }
    }
    else //If the front is not less than 25 cm
(greater than)
    {
        directionn = Fgo; //Walk forward
    }
}

void ask_pin_F() // Measure the distance ahead
{
    myservo.write(90);
    digitalWrite(outputPin, LOW); // For low voltage 2 us ultrasonic
launch

```

```

        delayMicroseconds(2);
        digitalWrite(outputPin, HIGH); // Let ultrasonic launch 10 us high
voltage, there is at least 10 us
        delayMicroseconds(10);
        digitalWrite(outputPin, LOW); // Maintaining low voltage
ultrasonic launch
        float Fdistance = pulseIn(inputPin, HIGH); // Read the time
difference
        Fdistance= Fdistance/5.8/10; // A time to distance distance
(unit: cm)
        Serial.print("F distance:"); //The output distance (unit: cm)
        Serial.println(Fdistance); //According to the distance
        Fspeedd = Fdistance;
    }
void ask_pin_L() // Measure the distance on the left
{
    myservo.write(5);
    delay(delay_time);
    digitalWrite(outputPin, LOW); // For low voltage 2 us ultrasonic
launch
    delayMicroseconds(2);
    digitalWrite(outputPin, HIGH); // Let ultrasonic launch 10 us high
voltage, there is at least 10 us
    delayMicroseconds(10);
    digitalWrite(outputPin, LOW); // Maintaining low voltage
ultrasonic launch
    float Ldistance = pulseIn(inputPin, HIGH); // Read the time
difference
    Ldistance= Ldistance/5.8/10; // Will be time to distance
distance (unit: cm)
    Serial.print("L distance:"); //The output distance (unit: cm)
    Serial.println(Ldistance); //According to the distance
    Lspeedd = Ldistance; // Will reading Lspeedd distance
}
void ask_pin_R() // Measure the distance on the right
{
    myservo.write(177);
    delay(delay_time);
    digitalWrite(outputPin, LOW); // For low voltage 2 us ultrasonic
launch
    delayMicroseconds(2);
    digitalWrite(outputPin, HIGH); // Let ultrasonic launch 10 us high
voltage, there is at least 10 us
    delayMicroseconds(10);
    digitalWrite(outputPin, LOW); // Maintaining low voltage
ultrasonic launch
    float Rdistance = pulseIn(inputPin, HIGH); // Read the time
difference
    Rdistance= Rdistance/5.8/10; //Will be time to distance
distance (unit: cm)
    Serial.print("R distance:"); //The output distance (unit: cm)
    Serial.println(Rdistance); //According to the distance
    Rspeedd = Rdistance;
}
}

void loop()

```

```
{
  myservo.write(90); /*Make the servo motor ready position Prepare the
next measurement */
  detection();      //Measuring Angle And determine which direction to
go to

  if(directiononn == 2) //If directiononn (direction) = 2 (back)
  {
    back(8);          // back
    turnL(2);        //Move slightly to the left (to prevent
stuck in dead end lane)
    Serial.print(" Reverse "); //According to the direction (reverse)
  }
  if(directiononn == 6) //If directiononn (direction) = 6 (right)
  {
    back(1);
    turnR(6);        // right
    Serial.print(" Right "); //According to the direction (Right)
  }
  if(directiononn == 4) //If directiononn (direction) = 4 (left)
  {
    back(1);
    turnL(6);        // left
    Serial.print(" Left "); //According to the direction (Left)
  }
  if(directiononn == 8) //If directiononn (direction) = 8 (forward)
  {
    advance(1);      // go
    Serial.print(" Advance "); //According to the direction (Advance)
    Serial.print(" ");
  }
}
```