

**Іванов С.А.**

**Програмування та технологія  
виробів електронного текстилю  
(STEAM-підхід)**



**Запоріжжя**  
Видавництво “Освіта”

**2021**

УДК  
ББК

**Іванов С.А.**

Програмування та технологія виробів електронного текстилю. (STEAM-підхід) / С.А. Іванов. – Видавництво “Освіта”, 2021. – 90 с.: **іл.**  
ISSN

Посібник призначено насамперед для вчителів трудового навчання та вчителів інформатики, які прагнуть використовувати навчальний набір Arduino LilyPad на STEM-уроках. У посібнику розглянуто опис архітектури мікроконтролера Arduino LilyPad; компоненти для створення проєктів та дослідження фізичних явищ, які покладені в основу їх роботи; програмування мікроконтролера в середовищі Arduino IDE; основи теорії електричних кіл; практичні поради щодо вимірювання електричних параметрів за допомогою мультиметра. Важливою складовою посібника є практичний матеріал із питань технології шиття виробів, що розроблені на базі платформи Arduino LilyPad.

Головною метою посібника є мотивувати вчителів трудового навчання, вчителів інформатики та працівників позашкільних освітніх закладів створювати вироби з розумного текстилю, освоїти програмування поведінку його компонентів, ознайомити з елементами теорії електрики, розвинути творче мислення, створюючи власні проєкти за допомогою платформи Arduino LilyPad.

**ЗМІСТ**

ВСТУП	
Розділ 1. Апаратна частина платформи Arduino LilyPad	
1.1. Загальний огляд платформи Arduino LilyPad	
1.2. Апаратне забезпечення мікроконтролера Arduino LilyPad USB	
1.3. Контакти та сигнали	
1.4. Окремі компоненти для проєктів розумного текстилю	
Розділ 2. Програмне забезпечення мікроконтролера Arduino LilyPad	
2.1. Установка програмного забезпечення	
2.2. Інтерфейс інтегрованого середовища розробки IDE	
2.3. Структура програми Arduino	

2.4. Програмування базових компонентів набору Arduino LilyPad	
Розділ 3. Розрахунок параметрів електричного кола проєктів LilyPad	
3.1. Електричний струм	
3.2. Основні параметри електричного кола	
3.3. Фізичні процеси у світлодіодах	
3.4. Електричне з’єднання компонентів проєкту	
3.5. Використання мультиметра для виміру параметрів електричного кола	
Розділ 4. Технологічні основи шиття електронного текстилю	
4.1. Основні технологічні операції	
4.2. Поради щодо шиття струмопровідними нитками	
4.3. Перевірка працездатності частини готового проєкту	
Розділ 5. Приклади проєктів на платформі Arduino LilyPad	
5.1. Простий проєкт без програмування — “Ілюмінована шпилька”	
5.2. Проєкт без програмування “Підсвічена маска”	
5.3. Проєкт без програмування “Підсвічений плюш”	
5.4. Проєкт “Програмоване включення світлодіодів”	
ВИСНОВКИ	

## ВСТУП

Електронний текстиль (е-текстиль), який також називають розумним текстилем, це нове покоління одягу та аксесуарів із вбудованими мікроконтролерами. Такий інноваційний продукт являє собою впорядковану сукупність компонентів (світлодіодів, датчиків або інших пристроїв), програмування роботи яких створює нові можливості для творчих розробок виробів, що здійснюють інтелектуальний зв’язок між тілом, одягом та навколишнім середовищем. У поєднанні з відкритим кодом платформи Arduino така технологія використовується художниками, дизайнерами, інформатиками, музикантами, що насправді можна розглядати як приклади реалізації STEAM-діяльності [1].

Даний посібник має за мету набуття певного досвіду зі створення виробів з електронного текстилю, використовуючи відповідний набір на базі мікроконтролера Arduino LilyPad. Він надає можливість ознайомитися з основами програмування мікроконтролерів платформи Arduino LilyPad, фізичними принципами роботи основних електричних компонентів та елементами технології шиття розумного одягу струмопровідними нитками.

Посібник може бути використаний на уроках інформатики та уроках трудового навчання, на заняттях у відповідних гуртках та на заняттях у системі позашкільної освіти. Оволодіння теоретичними знаннями, практичними вміннями і навичками у сфері конструювання виробів розумного текстилю надає можливості учням сформувати компетентності для заняття майбутніх вакансій на ринку STEAM-спеціальностей.

## **Розділ 1. Апаратна частина платформи Arduino LilyPad**

### **1.1. Загальний огляд платформи Arduino LilyPad**

Навчальна платформа Arduino LilyPad - це відкрита сукупність електронних компонентів, призначених для розробки функціональних та інтерактивних виробів із тканини (електронний текстиль). Мозком платформи в цих výroбах виступає мікроконтролер Arduino LilyPad, який сприймає інформацію з компонентів набору у вигляді електричних сигналів, обробляє її та видає управлінські команди, що впливають на поведінку виробу. Платформу було створено Leah Buechley та SparkFun Electronics і побудовано на мікроконтролерах Atmega, зокрема, на Atmega328V, Atmega32U4, ATtin85 та інших.

Існує навчальний набір Arduino LilyPad, в якому представлено основні компоненти платформи (рис. 1.1).

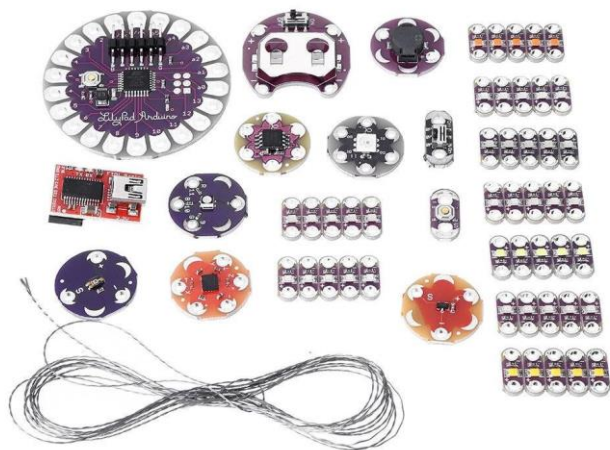


Рис. 1.1. Набір Arduino LilyPad

Основна плата Arduino LilyPad виробляється в легкому круглому корпусі, діаметром 50 мм з товщиною 0,8 мм. Її електричні контакти являють собою широкі виступи, що спрощує пришивання до тканини та забезпечує надійне електричне з'єднання з компонентами (датчиками та актуаторами) струмопровідною ниткою (рис. 1.2).

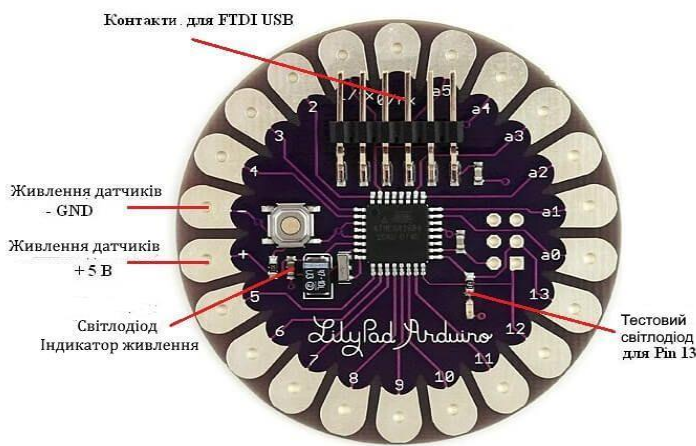


Рис. 1.2.Контакти мікроконтролера Arduino LilyPad

Сьогодні є кілька варіантів плати Arduino LilyPad:

LilyPad 328 є варіантом мікроконтролера без USB-входу, тому для прошивки (завантаження програмних кодів) застосовується адаптер Serial To USB (рис. 1.3а);

Lilypad Arduino USB базується на мікроконтролері ATmega32U4 і має відповідний вхід для зв'язку з комп'ютером (рис. 1.3б);

LilyPad Arduino SimpleSnap - плата з кнопками-скріпками для прикріплення та зняття з одягу, додатково до неї існує плата Simple Snap ProtoBoard, яка пришивається до тканини; сама ж платформа містить вбудований літійовий акумулятор і пранню, відповідно, не підлягає; для прошивки необхідний Serial To USB Adapter (рис. 1.3в);

LilyTiny - це полегшена версія LilyPad Arduino. Замість ATmega328 тут встановлено мікроконтролер ATtiny85 (рис. 1.3.г). Завдяки цьому LilyTiny займає менше місця, споживає менше енергії і коштує дешевше.

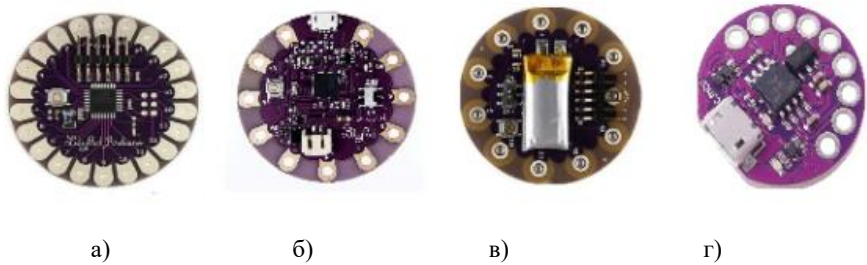


Рис. 1.3. Варіанти плати Arduino LilyPad

**1.2. Апаратне забезпечення мікроконтролера Arduino LilyPad USB**

До складу програмно-апаратної платформи Arduino входить апаратне і програмне забезпечення. Плата Arduino LilyPad USB складається з мікроконтролера Atmel AVR (ATmega32u4) та елементної обв'язки (аналогові та цифрові контакти вводу-виводу) для програмування та інтеграції з іншими компонентами. Відмінними рисами програмно-апаратної платформи Arduino є:

- 1) багатоплатформеність (здатна працювати на ОС Windows, Macintosh, Linux);
- 2) спрощеність оболонки програмного середовища і мови, **доступного** для використання в освітньому процесі;

спрощеність оболонки програмного середовища, доступної для використання в освітньому процесі, і мови; або

спрощеність оболонки програмного середовища і мови, доступних для використання в освітньому процесі.

3) Usb-подібний інтерфейс для організації комунікацій із зовнішнім світом і для живлення;

4) відкрите апаратне та програмне забезпечення;

5) освітній характер платформи.

Arduino дозволяє учням зосередитися на розробці проєктів, а не на вивченні пристрою та принципів функціонування окремих елементів. Розробник може просто підключити до Arduino необхідні компоненти. Усі зусилля розробник спрямовує на розробку керувальної програми мовою високого рівня. Отже, доступ до розробок мікропроцесорних приладів отримали не тільки професіонали, але й просто аматори, включаючи учнів середніх шкіл.

Технічні характеристики мікроконтролера ATmega32u4:

Робоча напруга	3.3 В
Напруга живлення: 3.7-5 В	3.7-5 В
Максимальний постійний струм через контакт вводу/виводу	40 мА
Кількість цифрових входів / виходів	9
Канали ШІМ	4
Кількість аналогових входів	4
Обсяг Flash-пам'яті	32 кб
Обсяг SRAM-пам'яті	2.5 кб
Обсяг EEPROM-пам'яті	4 кб
Тактова частота	8 МГц

Живлення плати LilyPad Arduino USB може здійснюватися від комп'ютера через вхід USB, від батареї CR2032 або від літєвого акумулятора (підключається до роз'єму JST на платі). При застосуванні інших джерел живлення використовується вбудований стабілізатор MIC5219, що знижує напругу живлення до необхідних 3.3 В.

Для операцій включення/виключення пристрою на платі передбачений спеціальний перемикач: у позиції "ON" мікроконтролер підключається до джерела живлення, при перемиканні в позицію "CHG" - відключається. LilyPad Arduino USB також містить мікросхему MCP73831, яка керує процесом зарядження LiPo-акумулятора. Процес заряду активізується при підключенні пристрою до встановленого акумулятором до USB-порту комп'ютера, про що буде свідчити світлодіод над словом "CHG". Підзарядка акумулятора здійснюється незалежно від положення перемикача живлення та автоматично припиняється, коли акумулятор повністю заряджений (рис. 1.4.).

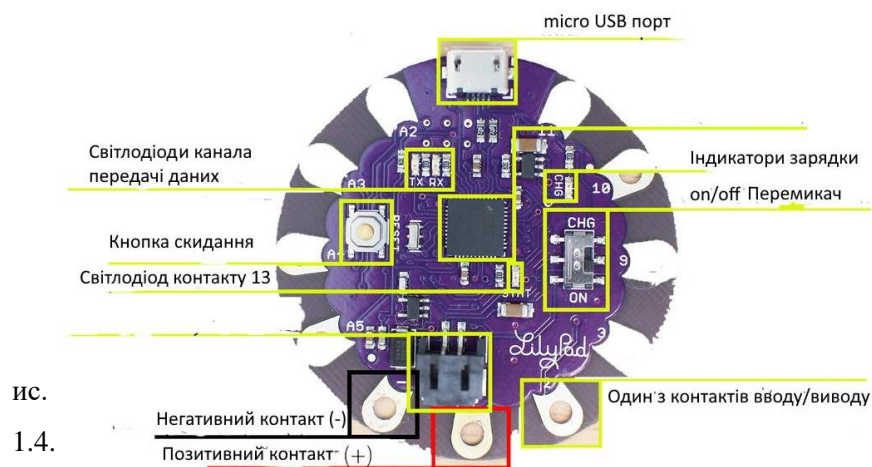


рис. 1.4. Інтерфейс мікроконтролера Arduino LilyPad USB

**1.3. Контакти та сигнали.** Контакти (pin) на мікроконтролері – це те, до чого підключаються компоненти проекту (датчики та актуатори). На платі Arduino LilyPad USB це металеві накладки з отворами через них, що полегшує пришивання та закріплення цих вкладок. Не всі контакти на мікроконтролері здатні виконувати однакові функції. Щоб встановити, який контакт



використовувати, потрібно знати, з яким компонентом він об'єднується та яка поведінка цього компоненту програмується. Зауважимо, що всі представники платформи Arduino мають можливість працювати з цифровими та аналоговими сигналами (рис. 1.5).

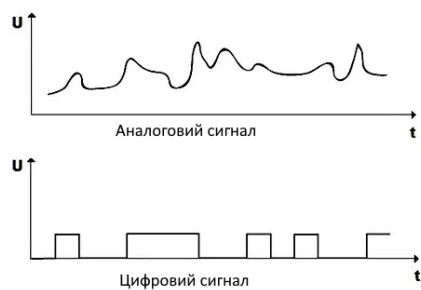


Рис. 1.5. Аналоговий та цифровий сигнали

Нижче перераховані контакти на платі:

**Вивід "+".** Стабілізована напруга 3.3 В, що використовується для живлення мікроконтролера та інших компонентів на платі. Надходить від вбудованого стабілізатора напруги, вхід якого може бути підключений або до USB, або до акумулятора. Напруга на цьому виводі присутня тільки тоді, коли перемикач живлення знаходиться в положенні "ON".

**Вивід "-".** Вивід землі.

**Цифрові входи і виходи.** Кожен із 9 цифрових контактів LilyPad Arduino USB може працювати як вхідний або як вихідний (логічні значення HIGH або LOW (0 або 1)). Рівень напруги на виводах обмежений - 3.3 В. Максимальний струм, який може віддавати або споживати один вивід, становить 40 мА. Усі виводи пов'язані з внутрішніми підтягувальними резисторами (за замовчуванням відключеними) номіналом 20-50 кОм. Підтягувальний резистор використовується для захисту мікроконтролера від випадкових спрацьовувань: він чітко визначає логічні рівні 0 та 1 (або 3.3В, коли цифровий вихід не підключений до схеми).

**Світлодіод вивід 13.** Вбудований світлодіод, приєднаний до контакту 13. При відправленні значення HIGH (3.3 В) світлодіод вмикається, при відправленні LOW (0 В) - вимикається.

**Вводи з широтно-імпульсною модуляцією (ШІМ):** 3, 9, 10, 11 і 13. За допомогою певних функцій програмування (про що далі) ці виводи можуть мати так званий цифровий ШІМ-сигнал. Використання ШІМ-виводів забезпечує проміжні значення напруги на вказаних контактах від 0 до 3.3 В з певним кроком. У Arduino LilyPad USB використовується 8-бітна логіка, тобто сигнал може бути представлений як  $2^8$ , що дорівнює 255. Це означає, що на вказаних контактах напруга може бути запрограмована у діапазоні від 0 до 3.3 В з кроком  $3.3/255$ , що складає приблизно 0.013 В. Іншими словами, ШІМ-контакти дозволяють формувати практично аналогові сигнали.

**Аналогові виводи.** LilyPad Arduino USB має 4 аналогових виводи (A2 - A5), кожен з яких може використовуватися також як звичайний цифровий вивід. Кожен із входів може формувати аналогову напругу у вигляді 10-бітного числа:  $2^{10}$  (1024 різних значення). У цьому випадку крок зміни напруги складає:  $3.3 / 1024$ , тобто приблизно 0,003 В, що точніше, ніж на ШІМ-виводах.

**1.4. Окремі компоненти для проєктів розумного текстилю**

**Світлодіодні модулі (LilyPad LED)** є мінімалістичні компоненти із світлодіодом певного кольору (білого, синього, червоного, рожевого, пурпурного, зеленого), струмообмежуючими резисторами та отворами-вухками для пришивання. Зворотна напруга – 5 В, прямий струм світлодіоду — 30 мА, піковий струм — 150 мА. Такі елементи з їх красивим світінням можуть виступати в ролі маячків на одязі й змусять його сигналізувати. Прості проєкти з ними — це прикраси одягу, освітлення жіночої сумочки в темний час, оснащення костюмчика для улюбленої собачки тощо (рис. 1.6.).



Рис. 1.6. Світлодіодні модулі різного кольору

**RGB-модуль (LilyPad Tri-Color LED RGB Module)** з групою

світлодіодів, кожним з яких можна управляти окремо з мікроконтролера. Завдання з підсвічуванням розширюється: колір маячків на одязі може змінюватися залежно, наприклад, від часу або від настрою (рис. 1.7.).

Кольорові канали на цьому світлодіоді RGB з'єднані через загальний катодний (негативний) контакт: окремі червоні, зелені та сині світлодіоди мають спільну вкладку заземлення. Щоб запалити кожен колір окремого світлодіода треба приєднати його до джерела живлення. Зворотна напруга світлодіодів складає 5 В, прямий струм — від 30 до 50 мА залежно від кольору.



Рис. 1.7. RGB- плата

**Розумні світлодіоди LilyPad Pixel Board**

Використання адресних плат LilyPad Pixel Boards дозволяє демонструвати різні кольори з меншою кількістю проводів - деякі з них можуть бути пов'язані один з одним і управлятися одним контактом на LilyPad Arduino (рис. 1.8).

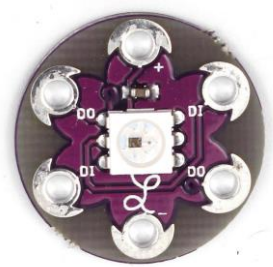


Рис. 1.8. Адресна плата LilyPad Pixel Board

Піксельна плата оснащена чипом WS2812B, вбудованим прямо в

світлодіод. На відміну від світлодіодів LilyPad, які спалахують при підключенні до джерела живлення, піксельні плати LilyPad повинні бути підключені до мікроконтролеру, який керує спалахуванням за програмою.

**Датчик освітлення Light Sensor T6000** - це прошивна плата із вбудованим світлодіодним датчиком ALS-PT19, яка готова до використання одразу. Кожен датчик виводить напругу від 0 В до 3,3 В залежно від рівня освітлення навколишнього середовища. Струм колектора залежить від освітленості й дорівнює 10 мА (20 люкс) і 50 мА (100 люкс), діапазон спектрального виявлення потоку - 440 nm-800nm, тобто, у діапазоні видимого випромінювання. Якщо на датчик надходить більше світла, буде надходити більше струму з плати до мікроконтролера, до якого підключено датчик. Якщо датчик не отримує світла, через нього не проходить струм. У типових умовах освітлення в приміщенні датчик буде виводити сигнал від 1 до 2 В (рис. 1.9).



Рис. 1.9. Датчик освітлення

**Маленький модуль-вимикач (LilyPad Board button)** дозволяє вмикати/розмикати електричне коло пристрою для початку/закінчення роботи та є незамінним для збереження заряду батареї. Він дуже компактний і не має гострих кутів, що важливо при використанні на одязі (рис. 1.10.).



Рис. 1.10. Кнопка вмикання/вимикання

**Тактова кнопка-перемикач (LilyPad Slide Switch)** є також досить

мініатюрним модулем, розташованим на невеликій платі з вкладками для пришивання (рис. 1.11). Кількість механічних та електричних циклів дорівнює 10000, опір у контакту — 70 мОм. За допомогою цієї кнопки можна управляти режимами миготіння світлодіодів на одязі та аксесуарах: наприклад, тривожний режим - часте миготіння (актуально при наближенні транспортного засобу в темний час доби для того, щоб зробити людину більш помітною на узбіччі дороги). У подібних проєктах зручно буде використовувати мініатюрний аналоговий датчик освітленості, сигнал з якого під час вступу на контролер дозволяє управляти режимами миготіння світлодіодів.

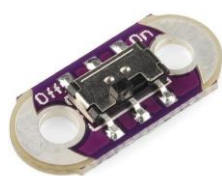


Рис. 1.11. Перемикач режимів роботи

**LilyPad Buzzer** (зумер) є компонентом, що використовується для відтворення звуків, які створюються на основі різної частоти перемикання вводу-виводу, змусивши одяг вже не просто миготіти, але і передавати чутні сигнали (рис. 1.12.). Наприклад, такий сигнал-пищалка може нагадати хворому випити потрібну таблетку, а спортсмену - зробити черговий підхід до снаряду. Слід зазначити, що зумер не підлягає пранню, тому потрібно передбачити його демонтаж при чищенні одягу. Робочий діапазон напруги датчика складає від 2.5 В до 4.5 В, максимальний струм — 110 мА. Коли зумер не використовується, рекомендується встановлювати обидва контакти вводу-виводу на низький рівень (0 В). Це індуктивний зумер, що означає, що він буде діяти як коротке замикання на землю.



Рис. 1.12. LilyPad зумер

**Компактний вібротор** (LilyPad Vibe Board) – актуатор, який може нагадувати лише власнику одягу таємною підказкою будь-яку з умов, зокрема, перевищення будь-якого параметру вимірювання, наприклад, часу роботи (рис. 1.13.). Операційна напруга вібротора становить 2.5-3.8 В, номінальний струм - 75 мА.



Рис. 1.13. Вібротор для таємного оповіщення

**Аналоговий датчик температури** (LilyPad Temperature Sensor) дозволяє відстежувати температуру навколишнього середовища або температуру тіла власника (рис. 1.14). Діапазон вимірювання становить  $-40^{\circ}\text{C}$  -  $+125^{\circ}\text{C}$ , операційний струм споживання - 6 мА. Цей датчик буде корисним для мами, яка уважно стежить за здоров'ям своєї дитини. Плата датчика температури буде виводити питому напругу при встановлених температурах - 10 мВ для кожного градуса Цельсія ( $^{\circ}\text{C}$ ). При нульовому градусі Цельсія датчик видає 0,5 В. Струм, що протікає через вкладку сигналу, можна прочитати за допомогою аналогової вкладки на платі LilyPad Arduino і перетворити за формулою в градуси за Цельсієм або Фаренгейтом.



Рис. 1.14. Датчик температури

**Модуль-акселерометр** (LilyPad ADXL335 three-axis gravity acceleration sensor) - тривісний акселерометр для системи LilyPad може виявляти рух суглобів, а також нахили та вібрації. ADXL335 виводить аналоговий сигнал

від 0 до 3.3 В на кожну з осей X, Y та Z (операційний діапазон — 1.8-3.6 В). Струм споживання становить 350 мА.

Робота датчиків акселерометра заснована на вимірюванні зміни ємності між рухомими пластинами: коли прискорення застосовується в будь-якому напрямку, ємність між пластинами змінюється. Arduino LilyPad USB зчитує ці значення прискорення з датчика, а потім обчислює значення тангажу (кутовий рух) та нахилу (рис. 1.15).

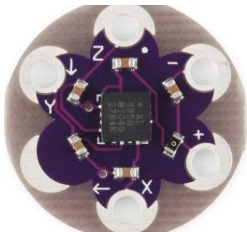


Рис. 1.15. Модуль-акселерометр

**Розділ 2. Програмне забезпечення Arduino LilyPad**

**2.1. Установка програмного забезпечення**

Перш ніж підключати плату Arduino LilyPad USB до комп'ютера, необхідно встановити інтегроване середовище розробки плати IDE (Integrated Development Environment). Порядок установки IDE викладений у багатьох інтернет-джерелах, але у зв'язку з тим, що релізи постійно оновлюються, цей порядок змінюється. Тому слід вибрати порядок, який відповідає обраному типу плати і відповідній версії програмного забезпечення. Найкраще слід звернутися до офіційного сайту <https://www.arduino.cc> та слідувати порядку, інсталяції. Між іншим, сьогодні вже існує можливість роботи з Arduino IDE в онлайн режимі [<https://create.arduino.cc/editor/>].

Порядок завантаження здійснюється так:

- а) Знайдіть останню версію на сторінці завантаження: <https://www.arduino.cc/en/Main/Software>). Після закінчення завантаження розпакуйте завантажений файл. Переконайтеся, що не порушена структура папок. Відкрийте папку подвійним кліком на ній.
- б) Під'єднати плату. Arduino LilyPad отримує живлення автоматично від будь-якого USB-підключення. Має засвітитися зелений світлодіод живлення, позначений PWR.

в) Встановіть драйвера. Порядок установки драйверів для Arduino LilyPad наданий у <https://doc.arduino.ua/ru/guide/LilyPadWindows#toc4>:

При підключенні плати Arduino до комп'ютера під керуванням Windows 10, треба впевнитися, що в системі встановлений драйвер. Драйвер налаштовує Arduino LilyPad як віртуальний COM-порт, який можна побачити в диспетчері пристроїв.

Відкрийте вікно Device Manager (Диспетчер пристроїв), клацнувши правою кнопкою миші кнопку запуску Windows 10 в лівому нижньому кутку екрану. У Диспетчері пристроїв розгорніть порти (COM і LPT), там можна побачити COM-порт, який буде Arduino LilyPad, як показано на рис. 2.1.

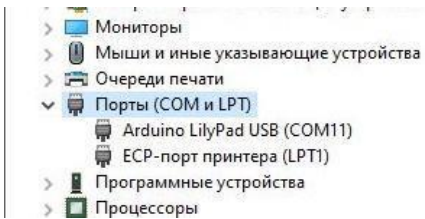


Рис. 2.1. Диспетчер пристрою для COM-порта 11

### 2.2. Інтерфейс інтегрованого середовища розробки IDE

Інтегроване середовище Arduino IDE дозволяє розробникам проєктів розумного одягу створювати програми взаємодії з мікроконтролерами платформи Arduino з одного боку для передачі даних із датчиків, з іншого - для прошивки коду в плату мікроконтролера.

Програми, що створюються в середовищі Arduino IDE (скетчі), пишуться в текстовому редакторі та зберігаються у файлах із розширенням .ino. Вбудований текстовий редактор має стандартні інструменти копіювання, вставки, пошуку та заміни тексту. Область повідомлень у вікні програми інформує розробника про події, у тому числі і про помилки, що виникають у процесі трансляції і виконання написаного коду. У нижньому правому куті вікна програми показується модель поточної плати та послідовний порт, до якого вона підключена. Кнопки на панелі інструментів призначені для



створення, відкриття, збереження та прошивки програм у плату Arduino. Окрема кнопка запускає програму SerialMonitor. Структура Інтерфейсу середовища IDE містить такі пункти (рис. 2.2):

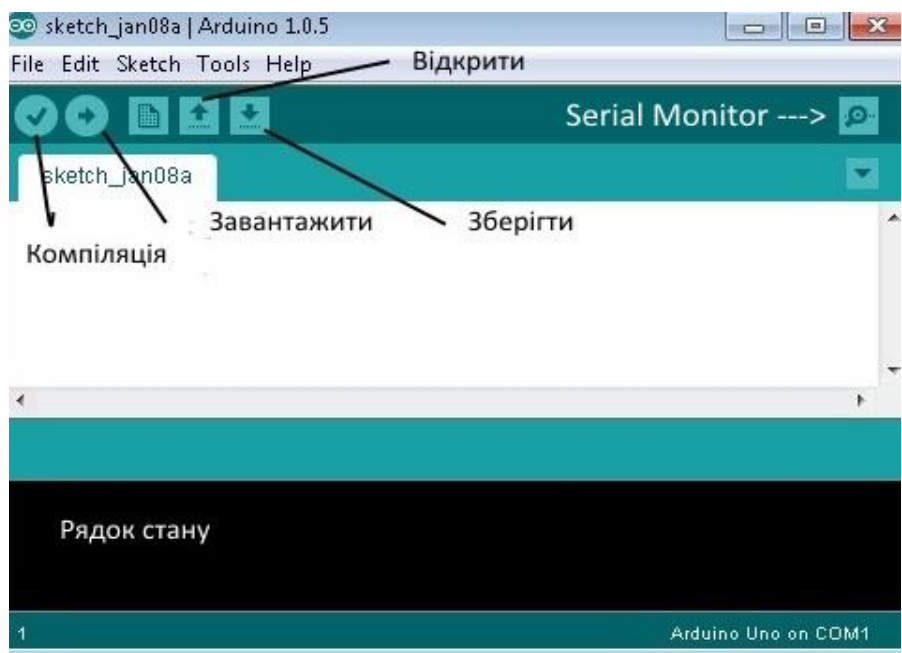


Рис. 2.2  
Структура Інтерфейсу середовища IDE

Додаткові команди знаходяться в меню: File, Edit, Sketch, Tools і Help. У цих опціях меню завжди активні тільки ті пункти, які можна застосувати до поточного елементу або фрагменту коду. Відзначимо деякі з них, з якими найчастіше зустрічається розробник. Найбільш важливим елементом опції «Інструменти» є можливість вибору відповідної плати, у нашому випадку Arduino LilyPad USB. Взагалі, у списку знаходяться всі офіційні версії Arduino. В опції «Інструменти» також можна встановити послідовний COM-порт, до якого підключена плата Arduino. Найчастіше пакет Arduino IDE сам визначає COM-порт, але іноді потрібно вручну встановити номер порту в налаштуваннях.

Дуже корисною особливістю програми є наявність вбудованого набору прикладів програм, який можна побачити в меню в опції **File**. Це насправді зручно, тому що приклади програм можна відразу перевірити, завантаживши їх в мікроконтролер. При необхідності можна зберегти приклад і змінити його відповідно до потреб користувача. Внаслідок набуття певного досвіду

використання Arduino багатьма авторами були створені проблемно-орієнтовані бібліотеки, які можна підключати до власного проєкту. Підключення бібліотек виконується в меню в опції **Sketch**. Більш детальну інформацію щодо структури середовища Arduino IDE надано в [<https://doc.arduino.ua/ru/guide/Environment> ].

### 2.3. Структура програми Arduino

Як було показано вище, програмування плати Arduino відбувається в середовищі IDE Arduino. Програма, що написана в цьому середовищі, має назву *скетч* (Sketch). Кожний скетч містить щонайменше дві програмні функції `setup()` та `loop()` :

```
void setup() { послідовність інструкцій; }  
void loop() { послідовність інструкцій; }
```

На початку програми, перед функцією `setup()`, зазвичай, оголошуються змінні, які будуть використані у програмі. Після включення живлення плати першим виконується оператор `setup()`, який використовується для ініціалізації змінних та визначення режимів роботи портів (налаштування на ввід або вивід). Ця функція також використовується для запуску необхідних бібліотек, що містять велику кількість скетчів для управління тими чи іншими елементами електронних пристроїв. Оператор `setup()` виконується лише один раз після кожної подачі живлення плати Arduino.

Після завершення роботи функції `setup()`, починає виконуватися код (скетч), написаний в тілі функції `loop()`. Важливо зазначити, що функція `loop()` виконується в нескінченному циклі, тобто знову і знову, чекаючи сигнали на виходах або опитуючи виходи пристроїв та виконуючи різні обчислення.

Написання тексту скетчу має відбуватися згідно з синтаксисом програми, тобто, згідно з певними вимогами до форматування, які є дуже простими. А саме:

// - ( коментар). Часто використовується для пояснення змісту кожного рядка програми. Все, що розміщено після символу `//` і до кінця рядка буде ігноруватися компілятором, спеціальною програмою, яка створює виконуваний файл.

{ } - (фігурні дужки). Використовуються для визначення початку і кінця блоку команд (використовуються у функціях і циклах).

/\* \*/ - (багаторядковий коментар). Ця структура використовується для розширених коментарів, що займають більше одного рядка. Все, що знаходиться між цими символами буде ігноруватися компілятором.

;- (крапка з комою). Кожна команда повинна закінчуватися цим символом. Відсутність крапки з комою в кінці команди - найбільш поширена помилка, що призводить до неможливості компіляції!

Будь-яка програма використовує дані (напруга, стан тих чи інших параметрів тощо). У мові Wiring існують різні типи даних, серед яких найчастіше використовуються такі типи даних (повний перелік типів даних можна переглянути за адресою [http://arduino.net.ua/Arduino\\_articles/](http://arduino.net.ua/Arduino_articles/)):

**int** - цілочисельний тип даних, який займає 2 байти і може приймати значення від -32 768 до 32767;

**boolean** - може знаходитися в одному із двох станів - **true** або **false** (хоча і займає цілий байт ОЗУ);

**float** - тип даних для зберігання чисел із плаваючою точкою. Числа із плаваючою точкою мають набагато більшу роздільну здатність ніж цілочисельні змінні;

**char** - цей тип даних займає 1 байт ОЗУ і зберігає один символ.

У тексті програми використовується велика кількість стандартних операторів і функцій. Зокрема, у функції `setup()` найчастіше використовуються оператори `pinMode(pin, OUTPUT)` та `pinMode(pin, INPUT)`, які визначають режим цифрових та аналогових портів (чи вони працюють або як OUTPUT (на цей порт подаються ті чи інші дані), або як INPUT (з цього порту зчитуються ті чи інші дані)).

У функції `loop()` використовуються стандартні функції, наприклад, `digitalWrite (pin, value)` або `digitalRead (pin, value)`, що означає, що на цифровий контакт `pin` (від 2 до 13) подається (або зчитується) потенціал `value`: або HIGH (5 В) або LOW (0 В). Подібні функції `analogWrite(pin, value)` або `analogRead(pin, value)` стосуються цифрових портів з позначкою ~ (ШІМ хвиля), яка видає аналогову величину (або вона зчитується) на порт входу/виходу.

Стандартні функції не охоплюють всі варіанти розробки Arduino-проектів. Велика кількість користувачів плати Arduino прагнуть розширити можливості плати, створюючи власні функції за допомогою додаткових бібліотек. Бібліотека - це набір функцій, призначених для того, щоб максимально спростити роботу з різними датчиками, приладами чи пристроями: ЖК-екранами, сервомоторами, кроковими двигунами, іншими модулями. Наприклад, вбудована бібліотека LiquidCrystal дозволяє легко взаємодіяти з символьними LCD-екранами. Знайти вбудовану бібліотеку можна у списку, який відкривається через опції “Скетч” > “Підключити бібліотеку”.

Існують два способи підключення бібліотек. По-перше, бібліотеку можна додати безпосередньо у середовищі Arduino IDE з офіційного репозиторію Arduino (репозиторій - це місце, де зберігаються будь-які дані). Для цього треба зайти в меню Arduino IDE опції “Скетч” > “Підключити бібліотеку” > “Управляти бібліотеками” (рис. 2.3).

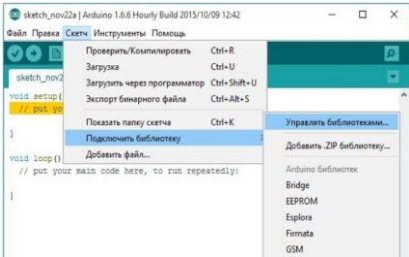


Рис. 2.3. Підключення бібліотеки з репозиторію

Після цього відкривається вікно пошуку (менеджер бібліотек) зі списком бібліотек, які або вже встановлені (позначка (INSTALLED)), або які можна встановити (через **info**). Для спрощення пошуку можна вказати тип та тему пошуку або відфільтрувати запитання (рис. 2.4). Якщо це зроблено, статус бібліотеки змінюється на INSTALLED, її можна приєднати через розділи меню “Скетч” > “Підключити бібліотеку” і знайти у списку.

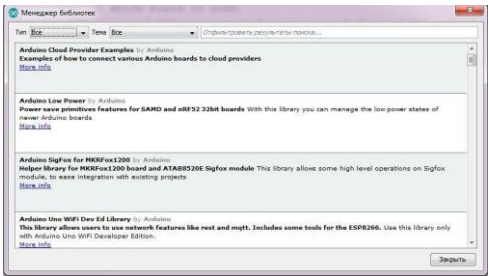


Рис. 2.4. Пошук бібліотеки в репозитарію

Існує інший спосіб підключення бібліотеки у вигляді папки, упакованої в zip-архів. Після завантаження бібліотеки у вигляді zip-архіву, його не треба розпаковувати. Далі потрібно вказати шлях до zip-файлу бібліотеки. Цей zip-файл буде розпакований і поміщений у папку Libraries в директорії зі скетчами Arduino. Після установки бібліотека стає доступною через меню “Скетч” > “Підключити бібліотеку”. Важливо зазначити, що більшість бібліотек містять відповідні приклади використання, але для того, щоб вони стали доступними в меню “Файл” → “Приклади”, необхідно перезапустити оболонку середовища Arduino IDE.

## 2.4. Програмування базових компонентів Arduino LilyPad

В інтегрованому середовищі Arduino IDE код, що виконується мікроконтролером, пишеться на мові Wiring, основна логіка якої базується мовою програмування C++ з урахуванням особливостей платформи Arduino. Нагадаємо, що в будь-якій програмі в середовищі Arduino IDE є дві принципові частини: підготовча частина та основний цикл. У підготовчій частині `setup()` виконуються операції, спрямовані на налаштування елементів схеми, які не будуть змінюватися впродовж усього проєкту, наприклад, оголошення контактів основної плати як вхід (INPUT) або вихід (OUTPUT) у процесі обміну між мікроконтролером та компонентами. Наприклад, якщо до контакту 3 було підключено світлодіод, цей контакт має бути налаштований на стан виходу (OUTPUT). Якщо ж в іншому проєкті до цього контакту буде приєднано кнопку, то треба налаштувати цей контакт на вхід (INPUT), тому що цей контакт буде зчитувати стан кнопки: чи вона включена? Тобто кожний контакт може бути налаштований або на вхід (INPUT), або на вихід (OUTPUT). Ця частина виконується лише один раз на початку роботи.

Мікроконтролер запам'ятовує всі призначення і переходить до основного циклу.

Основний цикл - це те, що відбувається у функції `loop()`. Мікроконтролер послідовно виконує всі команди, які розташовані між першою і останньою фігурними дужками. Як тільки команди закінчуються, він повертається в початковий цикл і все повторюється. І так до нескінченності. Це дуже важливо: цикл `loop()` є безперервним. Його можна зупинити лише через відключення живлення. Проте у процесі роботи програмний код не може по команді робити часові затримання.

В основному циклі описується поведінка компонентів проєкту, яка зумовлюється мікроконтролером або діями людини: миготіння світлодіодів, підключення датчиків, натискання кнопки, регулювання перемикачем тощо.

### Миготіння світлодіоду

З основними функціями середовища Arduino IDE краще ознайомитися на прикладі реальних проєктів, найпростішим з яких є миготіння вбудованого в плату світлодіоду, який підключений до контакту 13. Цей скетч знаходиться у середовищі Arduino IDE, де цей контакт визначено як `LED_BUILTIN`. Нагадаємо, що світлодіод являє собою схемотехнічний елемент, який пропускає електричний струм лише в одному напрямку і при цьому спалахує. У цьому скетчі в підготовчій частині `void setup()` використовується функція `PinMode(13, OUTPUT)`, яка налаштовує контакт 13 на стан виходу. В основному циклі використовуються функції: `digitalWrite(13, HIGH)`, яка включає світлодіод, та `digitalWrite(13, LOW)`, яка виключає світлодіод. Між цими операціями використовується функція `delay(t)`, де `t` — час затримання, який вимірюється в мілісекундах.

Ось цей скетч:

```
void setup() {  
    // оголошуємо цифровий контакт 13 як вихідний (OUTPUT)  
    pinMode(13, OUTPUT);  
}  
  
// безперервний цикл  
void loop() {  
    digitalWrite(13, HIGH); // включення світлодіода на контакті 13  
    delay(500);             // затримання на пів секунди  
    digitalWrite(13, LOW);  // виключення світлодіода на контакті 13  
}
```

```
    delay(1000);          // затримання на секунду  
}
```

У попередньому проєкті світлодіод приймає лише два електричних рівні: або 3.3, або 0 В (горить або гасне). Якщо треба плавно змінювати яскравість горіння світлодіода у діапазоні від 0 до 3.3 В, тоді треба використати контакт, який має властивість ШІМ-модуляції, наприклад, 3-ий контакт. Розглянемо відповідний скетч:

```
int led = 3;    // ШІМ-контакт, до якого приєднується світлодіод  
int brightness = 0;    // рівень яскравості  
int fadeAmount = 5;    // крок зміни яскравості  
void setup() {  
  pinMode(led, OUTPUT);    // декларуємо pin 3 як вихідний:  
}  
  
// нескінченний цикл  
void loop() {  
  // встановити рівень яскравості на контакті 3, позначений як led:  
  analogWrite(led, brightness);  
  // змінюємо яскравість на крок  
  brightness = brightness + fadeAmount;  
  // якщо світлодіод гасне (яскравість дорівнює 0) або  
  // або яскравість максимальна (255), починається все навпаки  
  if (brightness <= 0 || brightness >= 255) {  
    fadeAmount = -fadeAmount;  
  }  
  // чекаємо 30 мілісекунд  
  delay(30);  
}
```

### Програмування кнопки

Ускладнимо попередній проєкт - тепер світлодіод буде включатися після натискання на кнопку (рис. 2.5.).

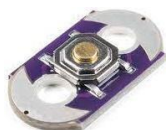


Рис. 2.5. Кнопка

Логіка роботи компонентів проекту реалізується у програмному коді шляхом використання блока умовного оператора *if*, який має таку конструкцію:

```
if (умова)

{

    // У цьому блоці список команд, що виконуються, якщо умова є істиною
    або має значення, відмінне від 0;

}

else

{

    // У цьому блоці список команд, що виконуються, якщо умова є
    помилковою або має значення, яке дорівнює 0;

}
```

Якщо послідовність команд складається з однією команди, то дужки {} можна не ставити (хоча категорично рекомендується ставити їх завжди, щоб уникнути помилок):

```
if (умова)

// Команда

else

// Команда
```

Можна обійтися і без блоку *else*, якщо ви хочете робити щось тільки при виконанні умови і не будете нічого робити, якщо умова не виконується.

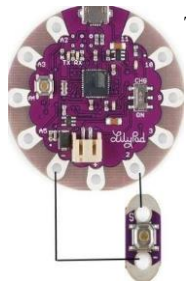
```
if (умова)

{
```

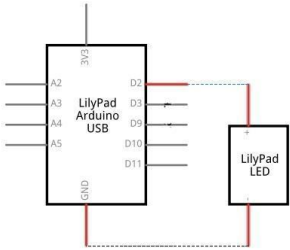


```
// Команди
}
```

Схематично проєкт з кнопкою виглядає так (рис. 2.6):



а)



б)

Рис. 2.6. Проєкт “Світлодіод - кнопка” (а) та його принципова схема (б)

Слід зауважити, що у цьому проєкті миготіння відбувається на вбудованому світлодіоді (контакт 13). Програмний код цього проєкту виглядає так:

```
// Підготовча частина
ledPin = 3
buttonPin = 2
void setup() {
    pinMode(ledPin, OUTPUT);    // ініціалізуємо контакт 3 як вихідний
    pinMode(buttonPin, INPUT);  // ініціалізуємо контакт 2 як вхідний
}

// Основний цикл
void loop() {
    // read the state of the pushbutton value:
    buttonState = digitalRead(buttonPin);

    /* перевірка чи натиснута кнопка, тобто чи поданий потенціал 3.3. В на
    контакти кнопки
    */
    if (buttonState == HIGH) {
        digitalWrite(ledPin, HIGH); // Світлодіод запалюється
    }
}
```

```
} else {  
    digitalWrite(ledPin, LOW); // Світлодіод гасне  
}  
}
```

### Програмування RGB-світлодіода

Кольорові канали на цьому світлодіоді RGB з'єднані через загальний анодний (позитивний) контакт. Щоб запалити кожен колір окремого світлодіода, треба приєднати швейні вкладки R, G або B до контактів 9, 10, 11 і програмно встановити їх на рівень HIGH (для цифрового виходу) або 255 (для аналогового виходу), щоб їх увімкнути (рис. 2.7).

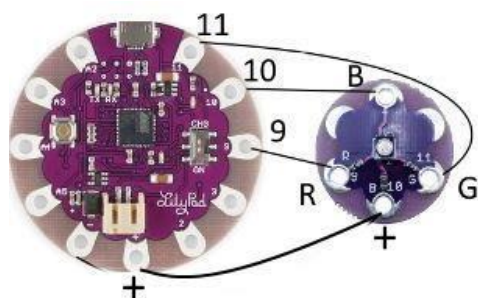


Рис. 2.7. Підключення RGB-світлодіода до плати Arduino LilyPad

Після завантаження коду світлодіод RGB запалюється послідовністю кольорів, починаючи з вимкнення всіх світлодіодів («чорний»), червоного, жовтого, зеленого, блакитного, блакитного, пурпурного та білого кольорів. Як тільки послідовність кольорів буде завершена, програма повернеться до початку і повторить послідовність.

Увімкнення різних комбінацій трьох світлодіодів всередині світлодіода RGB створює нові кольори. Поєднання основних кольорів світла (червоного, зеленого та синього) дає інші результати, ніж поєднання пігментів у фарбах або чорнилі. Увімкнення всіх трьох кольорів створить білий колір - це називається адитивним кольором. На рис. 2.8. показано як поєднуються кольори, щоб створити основні та вторинні кольори зі світлом.

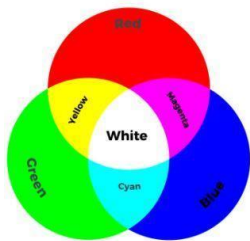


Рис. 2.8. Принципи формування спектру кольорів

У програмному коді проєкту з RGB-світлодіодом, представленому нижче, використовуються директива `#define`, яка дозволяє дати ім'я константі перед тим як програму буде скомпільовано.

```
// визначаємо контакти 9, 10, 11 як RGB_red, RGB_green, RGB_blue

#define led_red 11
#define led_green 10
#define led_blue 9

void setup() {
    // Установимо режими всіх контактів (9, 10, 11) як вихідні:
    pinMode(led_red, OUTPUT);
    pinMode(led_green, OUTPUT);
    pinMode(led_blue, OUTPUT);
} //end setup()

void loop()
{
    // світлодіоди виключені
    analogWrite(led_red, 0);
    analogWrite(led_green, 0);
    analogWrite(led_blue, 0);
    delay(1000);
    // Червоний колір
    analogWrite(led_red, 255);
    analogWrite(led_green, 0);
    analogWrite(led_blue, 0);
```

```
delay(1000);  
  
// Помаранчевий колір  
analogWrite(led_red, 255);  
analogWrite(led_green, 128);  
analogWrite(led_blue, 0);  
delay(1000);  
  
// Жовтий колір  
analogWrite(led_red, 255);  
analogWrite(led_green, 255);  
analogWrite(led_blue, 0);  
delay(1000);  
  
// шарттрес (жовто-зелений) колір  
analogWrite(led_red, 128);  
analogWrite(led_green, 255);  
analogWrite(led_blue, 0);  
delay(1000);  
  
// Зелений колір  
analogWrite(led_red, 0);  
analogWrite(led_green, 255);  
analogWrite(led_blue, 0);  
delay(1000);  
  
// колір ціан  
analogWrite(led_red, 0);  
analogWrite(led_green, 255);  
analogWrite(led_blue, 255);  
delay(1000);  
  
// фіолетовий колір  
analogWrite(led_red, 128);  
analogWrite(led_green, 0);  
analogWrite(led_blue, 255);  
delay(1000);  
  
// колір Маджента  
analogWrite(led_red, 255);  
analogWrite(led_green, 0);
```

```
analogWrite(led_blue, 255);  
delay(1000);  
// білий колір  
analogWrite(led_red, 255);  
analogWrite(led_green, 255);  
analogWrite(led_blue, 255);  
delay(1000);  
} //end loop
```

Представлений код є найпростішим, його можна ускладнювати, наприклад, використати контакти 9, 10, 11, які мають властивості ШІМ-модуляції, для поступового горіння світлодіодів від 0 до 3.3 В. Усі ці експерименти можна робити, вносячи певні зміни до скетчу.

### Програмування піксельної плати LilyPad Pixel Board

Піксельну плату оснащено чипом WS2812B, вбудованим безпосередньо у світлодіод. На відміну від світлодіодів LilyPad, які спалахують просто при підключенні до джерела живлення, піксельні плати LilyPad повинні завжди бути підключені до мікроконтролера, який керує спалахуванням за програмою (рис. 2.9).

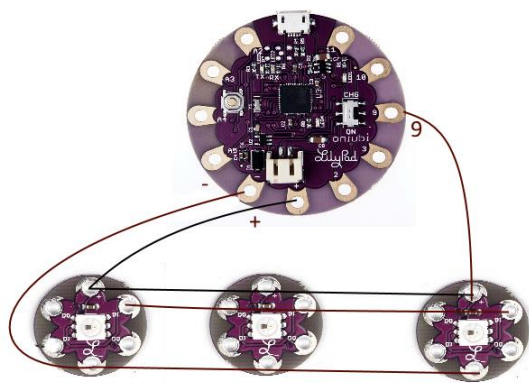
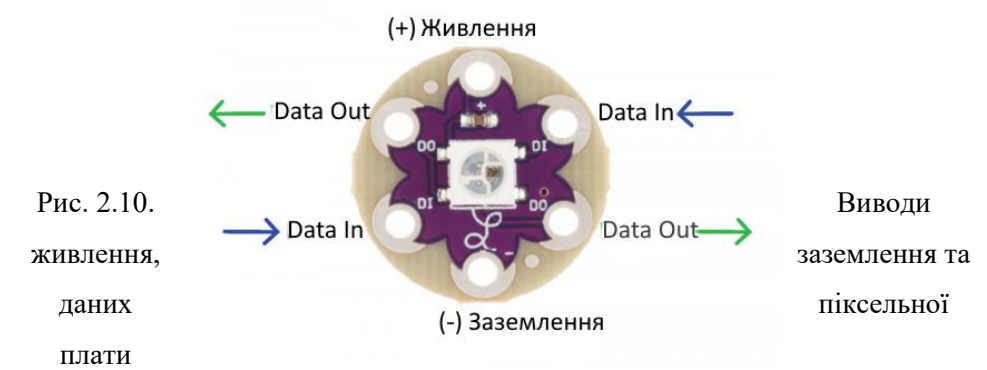


Рис. 2.9. Підключення піксельних плат до мікроконтролера

Піксельні плати LilyPad вимагають підключення трьох типів: живлення, заземлення та передачі даних. Виводи плати (+) і (-) підключаються до відповідних контактів на мікроконтролері LilyPad USB. Кожна плата має дві пари контактів даних: DI - для введення даних та DO - для виведення даних.

Одним із контактів Data IN на пікселях буде підключатися до свого контакту на LilyPad USB (рис. 2.10).



Щоб з'єднати більше одного пікселя, слід зв'язувати вихідні дані одного пікселя з наступним пікселем. Пікселі будуть пронумеровані у програмному коді середовища Arduino IDE. Для спрощення цього коду слід застосувати бібліотеку NeoPixel, в якій є відповідні функції. Підключення бібліотеки до скетчу відбувається за допомогою команди `#include <Adafruit_NeoPixel.h>`. Підключення бібліотеки до середовища Arduino IDE відбувається в опціях мен

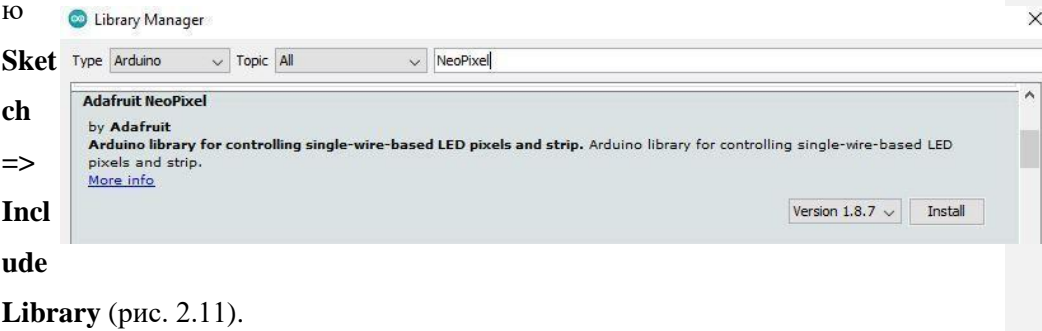


Рис. 2.11. Інсталяція бібліотеки NeoPixel

Структура бібліотеки Adafruit\_NeoPixel.h забезпечує створення класів та об'єктів — основних понять об'єктно орієнтованого програмування (ООП). Клас — тип, що описує структуру об'єктів. Об'єкт — це екземпляр класу. Клас можна порівняти із кресленням, за яким створюються об'єкти. У нашому випадку клас — це сама бібліотека Adafruit\_NeoPixel.h, а об'єкти (pixels) - конкретні світлодіоди. У програмному коді це виглядає так:

```
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(LED_COUNT, PIN, NEO_GRB + NEO_KHZ800);
```

Враховуючи зроблені пояснення, програмний код керування компонентами LilyPad Pixel Board виглядає так:

```
#include <Adafruit_NeoPixel.h>

#define PIN 9 // вивід LilyPad USB для підключення плати
#define LED_COUNT 3 // кількість LED, PIN – номер виводу
// Створюємо екземпляр pixels класу Adafruit_Neo
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(LED_COUNT, PIN, NEO_GRB + NEO_KHZ800); // прапорець піксельного типу

void setup()
{
  leds.begin(); // Запускаємо піксельну плату
  leds.show(); // Оновлення світлодіодів
}

void loop()
{
  // встановлюємо червоний колір першого світлодіода
  pixels.setPixelColor(0, 255, 0, 0);
  // встановлюємо зелений колір другого світлодіода
  pixels.setPixelColor(1, 0, 255, 0);
  // встановлюємо синій колір першого світлодіода
  pixels.setPixelColor(2, 0, 0, 255);
  pixels.show(); //Друк кольору
}
```

Крім програмування кольору бібліотека Adafruit\_NeoPixel.h забезпечує

встановлення яскравості світіння пікселів, використовуючи функцію `leds.setBrightness(brightness)`, де `brightness` визначається у межах від 0 до 3,3 В за допомогою потенціометра: `brightness = map(analogRead(potentiometer), 0, 1023, 0, 255)`. Тут функція `map` переносить значення з поточного діапазону значень аналого-цифрового перетворювача (0..1023) у новий діапазон (0..255) керування яскравістю світлодіода.

### Датчик освітлення ТЕМТ6000

Як випливає з назви, датчик світла ТЕМТ6000 визначає стан освітлення навколишнього середовища. Хоча існує багато властивостей світла, які можуть допомогти класифікувати його яскравість, ТЕМТ6000 вимірює освітленість у спектрі чутливості від 360 до 970 nm, що цілком достатньо для виробів з електронного текстилю. Датчик має аналоговий вихід. Перевага цього датчика освітленості полягає в тому, що він дуже чутливий до видимого спектра світла (від 380 до 780 nm) і майже нечутливий до інфрачервоного випромінювання, що робить його більш точним порівняно з фоторезисторами.

В основі роботи датчика освітлення ТЕМТ6000 лежить високоточний мініатюрний NPN кремнієвий планарний фототранзистор, принципову схему якого представлено на рис. 2.12.

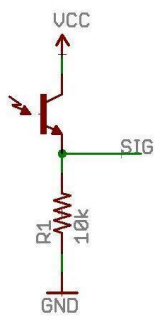


Рис. 2.12. Принципова схема фототранзистора

Фізика такого фототранзистора полягає у наступному. На датчик падає світло, між кінцями VCC та GND існує напруга 3.3 В. Залежно від інтенсивності світла змінюється струм фототранзистора та напруга між



кінцем SIG та GND. Чим більша яскравість світла, тим більший струм у фототранзисторі, тим більший сигнал на виході SIG. Схема підключення датчика освітленості до плати Arduino LilyPad 2.13.

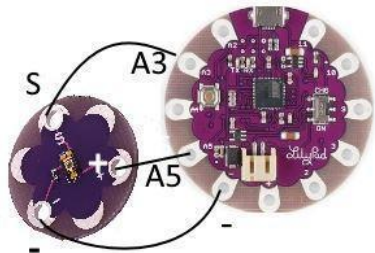


Рис. 2.13. Схема підключення датчика освітлення

Нижче представлений програмний код цього проєкту. Змінювати рівень освітленості можна просто прикриваючи рукою датчик (немає потоку світла) та підсвічуючи його лампою або ліхтариком смартфона. У цьому проєкті на контакт A5 програмно подається напруга 3.3 В, далі вимірюється напруга на контакті A3, яка змінюється залежно від рівня освітленості. Виміряна напруга порівнюється з пороговим значенням, яке задається як константа. Якщо напруга на контакті A3 стає меншою ніж порогове значення, загоряється вбудований світлодіод 13. У процесі експерименту значення напруги друкується на екрані Serial Monitor (рис. 2.14)

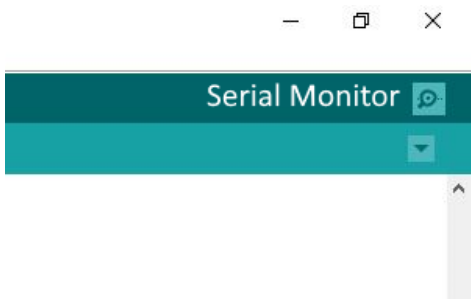


Рис. 2.14. Serial Monitor для відображення результатів

Тож, програмний код проєкту вимірювання значень температури навколишнього простору виглядає так:

```
const int darkLevel = 50;
```

```
// Створюємо змінну для величини, яка зчитується з контакту S датчика
int lightValue;

// Визначаємо до якого контакту основної плати приєднаний датчик
освітленості

int sensorPin = A3;

// Визначаємо контрольний контакт 13
int ledPin = 13;

void setup()
{
    // Встановлюємо sensorPin (A3) як вхідний (INPUT)
    pinMode(sensorPin, INPUT);

    // Встановлюємо контакт ledPin як вихідний (OUTPUT)
    pinMode(ledPin, OUTPUT);

    // Встановлюємо контакт A5 як вихідний
    pinMode(A5, OUTPUT);
    digitalWrite(A5, HIGH);

    // Встановлюємо швидкість передачі даних 9600 біт/
    Serial.begin(9600);
}

void loop()
{
    // Зчитуємо значення сигналу з A3 та зберігаємо як змінну lightValue
    lightValue = analogRead(sensorPin);

    // Друкуємо отримане значення lightValue на екрані Serial Monitor
    Serial.print("Light value is:");
    Serial.println(lightValue);

    // Compare "lightValue" to the "darkLevel" variable
    if (lightValue <= darkLevel)

    // Якщо отримане значення менше ніж порогове значення darkLevel
    {
        digitalWrite(13, HIGH); // Вмикається 13 світлодіод
    }
    else // якщо інакше значення lightValue більше ніж darkLevel
    {
```

```
digitalWrite(13, LOW); // 13 світлодіод гасне
    }

// затримка для забезпечення друку тексту в Serial Monitor
delay(100);
}
```

### Плата LilyTiny як альтернатива LilyPad

Плата LilyTiny - це крихітна плата, яка має власний мікроконтролер Atiny85, тобто цю плату можна також програмувати та використовувати без мікроконтролера Arduino LilyPad. Водночас, ця плата дозволяє створювати проекти, не вимагаючи надання власного коду. У найпростішому варіанті програмувати плату не треба, тому що плата попередньо запрограмована, що забезпечує миготіння світлодіодів цікавими візерунками (рис. 2.15).

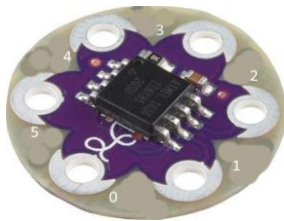


Рис. 2.15. Плата LilyTiny

LilyTiny має 6 вкладок-контактів. Два з них резервуються для живлення (+) і землі (-). Інші чотири контакти - ввід-вивід загального призначення (GPIO). LilyTiny попередньо завантажений зразком скетчу, який показує різноманітні світлодіодні візерунки на кожному з виводів (рис. 2.16):

- візерунок “дихання” (контакт 0);
- візерунок “серцебиття” (контакт 1);
- просте миготіння ввімкнення-вимкнення (контакт 2);
- випадкове затухання (контакт 3).

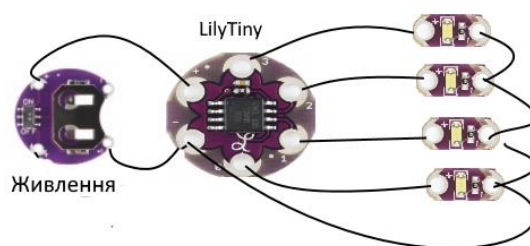


Рис. 2.16. Проєкт з LilyPad battery holder – LilyTiny – Leds

Зауважимо ще раз, що у цьому проєкті нічого програмувати не треба, програмний код вже “зашитий» у пам’ять плати LilyTiny. Цей код можна змінити шляхом програмування мікроконтролеру ATtiny85 за допомогою середовища Arduino IDE та спеціального завантажувача програматора Arduino as ISP [https://tsibrov.blogspot.com/2018/11/attiny85-getting-started.html].

### Зумер (звуковий актуатор) LilyPad Buzzer

Зумер LilyPad Buzzer дозволяє програмувати різні звукові сигнали за допомогою коду при підключенні до LilyPad Arduino. Програмуючи різні тони, можна створювати музичні мелодії, звуки спеціальних ефектів, будильники тощо. Цей зумер не дуже гучний, але його можна буде почути, якщо він вшитий в особистий одяг.

Зумер LilyPad має дві швейні вкладки: Power (+) та Ground (-), з яких перша підключається до контакту 3, друга (-) до відповідного контакту (-) на LilyPad Arduino USB (рис. 2.17).

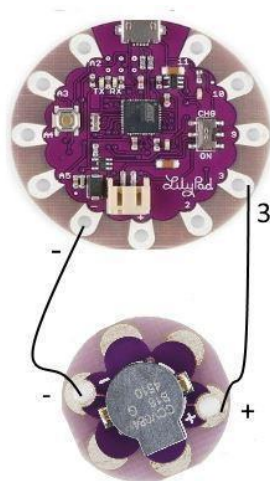


Рис. 2.17. Підключення зумера до плати

Програмний код проєкту представлений нижче:

// Встановлюємо контакт 3 для зв'язку з платою LilyPad

```
int buzzerPin = 3;
```

// Встановлюємо затримку в мілісекундах

```
int delayTime = 200;
```

// Ноти та їх частоти

```
const int C = 1046;
```

```
const int D = 2046;
```

```
void setup()
```

```
{
```

```
    // Встановлюємо контакт buzzerPin = 3 як OUTPUT
```

```
    pinMode(buzzerPin, OUTPUT);
```

```
}
```

```
void loop()
```

```
{
```

```
    // Використовуємо функцію tone() для активації
```

```
    tone(buzzerPin, C);
```

```
    delay(50);
```

```
    tone(buzzerPin, C);
```

```
    delay(100);
```

```
    tone(buzzerPin, D);
```

```
    delay(50);
```

```
    tone(buzzerPin, C);
```

```
    delay(500);
```

```
    // Функція noTone() припиняє роботу зумера
```

```
    noTone(buzzerPin);
```

```
    delay(delayTime);
```

```
}
```

**Акселерометр LilyPad Accelerometer - ADXL335**

LilyPad Accelerometer ADXL335 - це тривісний акселерометр для платформи LilyPad, який може виявляти рух суглобів, а також нахили та вібрації. ADXL335 виводить аналоговий сигнал від 0 до 3 В на кожну з осей X, Y та Z (рис. 2.18).

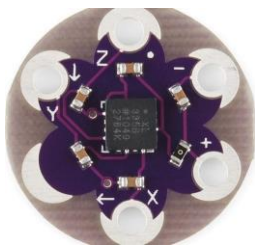


Рис. 2.18. Датчик LilyPad Accelerometer — ADXL335

LilyPad Accelerometer має 5 вкладок-контактів: два контакти для живлення та три контакти для вимірювання сигналів з трьох осей X, Y, Z. Схему підключення датчика представлено на рис. 2.19.

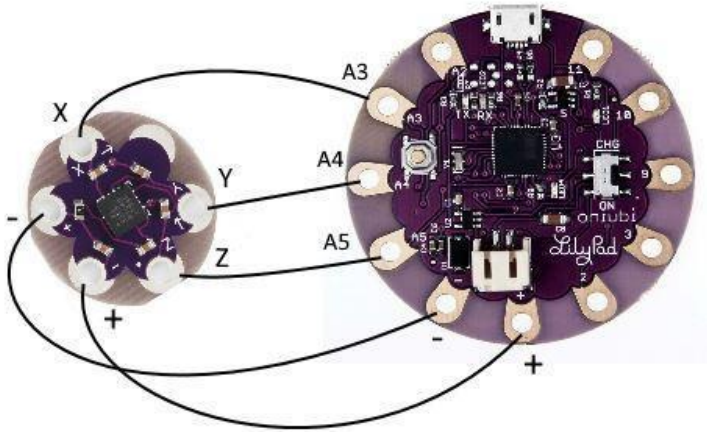


Рис. 2.19. Підключення датчика акселерометра до основної плати

Нижче представлений програмний код роботи акселерометра:

```
// аналогові контакти підключення координат x, y, z
const int pinX=A0;
const int pinY=A1;
const int pinZ=A2;
// змінні для зберігання значень
```

```
unsigned int x, y, z; // unsigned int — ціле без знака
void setup() {
    // запуск послідовного порту Serial
    Serial.begin(9600);
}

void loop() {
    // отримання
    x = analogRead(pinX);
    y = analogRead(pinY);
    z = analogRead(pinZ);
    // вивід у послідовний порт
    Serial.print(x);
    Serial.print(" ");
    Serial.print(y);
    Serial.print(" ");
    Serial.print(z);
    Serial.println();
    // пауза 3 мілісекунди
    delay(3000);
}
```

## Розділ 3. Розрахунок параметрів електричного кола проєктів LilyPad

### 3.1. Електричний струм

Одним із найпоширеніших запитань, які виникають при роботі з електронним текстилем, є таке: "Скільки світлодіодів (або інших компонентів) можна вшити у проєкт?" У цьому розділі розглядатимуться розрахунки для прогнозування вимог до батареї та деякі поради щодо створення реального проєкту з точки зору електрики.

Починаючи досліджувати світ електрики та електроніки, важливо згадати основні параметри електричних кіл: напруга, сила струму та електричний опір, які необхідні для управління та **використання потоками** електроенергії. Їх співвідношення та розрахунок визначаються законом Ома, а вимірювання цих величин можна здійснити, зокрема, мультиметром.

Хоч розділ електрики вивчається у шкільному курсі фізики, нагадаємо деякі основні положення. Взагалі, електрика — це сукупність природних явищ, обумовлених існуванням, взаємодією і рухом електричних зарядів або простіше - потоком електричних зарядів. Саме тут виникають питання: звідки беруться ті заряди, як вони переміщуються, як електричний заряд викликає

механічний рух або змушує речі, зокрема, світлодіоди, запалюватися? Щоб пояснити, що таке електрика, нам потрібно заглянути у світ атомів, з яких складаються молекули, які, своєю чергою, утворюють будь-яку речовину. Атоми існують у більш ніж ста різних формах як хімічні елементи. Атоми багатьох типів можуть об'єднуватися, утворюючи молекули, які будують матерію, яку можна фізично побачити та до якої можна доторкнутися.

Атоми мають крихітний розмір, десь  $3 \times 10^{-10}$  м, (це - 0,0000000003 метра). Мідний шматочок розміром з монету містить близько  $3,6 \times 10^{22}$  атоми (36 000 000 000 000 000 000 000 атомів). Але атом не є найменшою частинкою матерії, для пояснення роботи електрики потрібно опуститися ще на один рівень і подивитися на будівельні блоки атомів: протони, нейтрони та електрони. Кожен атом має центральне ядро, де протони і нейтрони щільно упаковані разом. Навколо ядра є певна кількість електронів, які рухаються орбітами (рис. 3.1.).

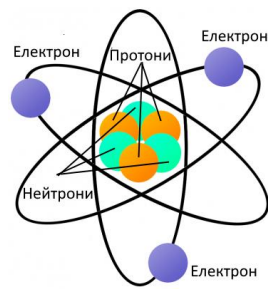


Рис. 3.1. Простіша модель атома

Кожен атом повинен містити принаймні один протон. Кількість протонів у атомі є важливою, оскільки вона визначає, який хімічний елемент являє собою атом. Наприклад, атом лише з одним протоном - це водень, атом з 29 протонами - це мідь, а атом із 94 протонами - це плутоній. Ця кількість протонів називається атомним номером.

Ядро-партнери протона - нейтрони, служать важливій меті, вони утримують протони в ядрі та визначають ізотоп атома. Вони не мають вирішального значення для нашого розуміння електрики, тому перейдемо до електронів. У своєму стабільному, збалансованому стані атоми мають таку ж саму кількість електронів, як і протони. Так атом є електрично нейтральною структурою. Варто ще раз зауважити, що така модель атома є спрощеною,



насправді електрони розташовані навколо ядра на кшталт хмарки (рис. 3.2).

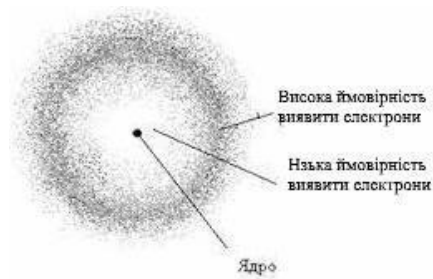
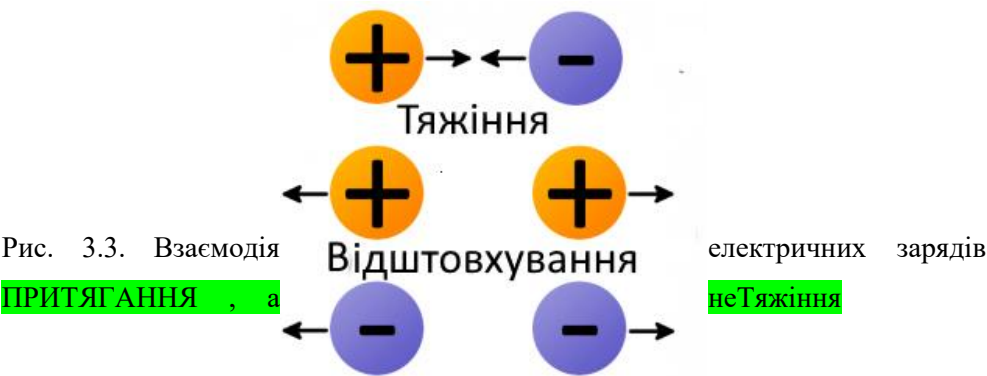


Рис. 3.2. Хмарна модель ядра атома

Електрони не завжди пов'язані з атомом. Електрони на зовнішній орбіті атома можуть вийти з орбіти атома і стати вільним. Якщо до вільних електронів докласти силу (електричне поле), заряди почнуть переміщуватися, а це і є електрика. Нагадаємо, що заряд (протонів або електронів) - це властивість матерії, так само як маса, об'єм або щільність, тобто кількість заряду можна вимірювати.

Для того, щоб примусити потік зарядів рухатися, потрібні носії заряду та електричне поле. Електрони завжди несуть негативний заряд, тоді як протони завжди позитивно заряджені. І електрони, і протони несуть однакову кількість заряду, тільки іншого типу. Електричні заряди взаємодіють між собою, сила взаємодії, яка отримала назву електростатичної сили, формулюється законом Кулона. Цей закон стверджує, що заряди одного типу відштовхуються один від одного, тоді як заряди протилежного типу притягуються (рис. 3.3.).



Величина сили, що діє на два заряди, залежить від того, наскільки вони

далекі один від одного. Чим ближче наближаються два заряди, тим більшою стає сила (або притягання, або відштовхування). Завдяки електростатичній силі електрони відштовхують інші електрони і притягуються протонами. Ця сила є частиною "клею", який утримує атоми разом, але та ж сама сила примушує електрони відриватися від атома та рухатися. Деякі атоми краще інших вивільняють свої електрони. Щоб отримати максимально можливий потік електронів, потрібно використовувати атоми, які не дуже міцно утримують свої електрони на зовнішній орбіті. Провідність речовини, як здатність проводити електричний струм, визначає наскільки міцно електрон пов'язаний з атомом. Речовини з високою провідністю, які мають дуже рухливі електрони, називаються провідниками. Це типи матеріалів, які використовуються для виготовлення дротів та інших компонентів. Такі метали, як мідь, срібло та золото, зазвичай, є найкращим вибором для хороших провідників. Речовини з низькою провідністю називаються ізоляторами. Ізолятори виконують дуже важливу функцію: вони перешкоджають потоку електронів. До найбільш відомих ізоляторів відносяться скло, гума, пластик і повітря (рис. 3.4.).



Рис. 3.4. Модель ізолятора

**3.2. Основні параметри електричного кола**

Використання електрики для живлення мікроконтролерів, світлодіодів та інших компонентів відбувається шляхом перетворення електричної енергії на інші види. Електричні схеми повинні мати можливість зберігати енергію та перетворювати її в інші форми, такі як тепло, світло або рух. Запасена енергія кола є електричною потенціальною енергією. Поняття енергії є

найважливішим у фізиці. У самому загальному розумінні енергія визначається як здатність об'єкта виконувати роботу над іншим об'єктом. Енергія надходить у різних формах, деякі можна бачити (наприклад, механічні), а інші - ні (наприклад, хімічні чи електричні). Незалежно від того, в якій формі вона знаходиться, енергія існує в одному з двох станів: кінетичному або потенційному.

Об'єкт має кінетичну енергію під час руху. Кількість кінетичної енергії, яку має об'єкт, залежить від його маси та швидкості. Потенційна енергія є накопиченою енергією, коли об'єкт знаходиться в стані спокою. Вона описує, скільки роботи об'єкт міг би виконати, якщо його привести в рух. Це загалом енергія, яку ми можемо контролювати. Коли об'єкт приводиться в рух, його потенційна енергія перетворюється на кінетичну.

Щоб краще зрозуміти, що таке потенційна та кінетична енергія, звернемося до явища гравітації. Будь-який предмет, що нерухомо знаходиться на вершині багатоповерхового будинку, має багато потенційної (накопиченої) енергії. Ця енергія утворилася внаслідок роботи з переміщення цього предмету з першого поверху до найвищого. У процесі падіння цього предмету під час його прискорення внаслідок тяжіння Землі, потенціальна енергія перетворюється в кінетичну енергію руху. Зрештою, вся енергія предмета перетворюється з потенційної в кінетичну. Нарешті, коли предмет лежить на землі, він має дуже низьку потенційну енергію.

Так само як маса у гравітаційному полі має потенціал гравітаційної сили, заряди в електричному полі мають потенційну електричну енергію. Ця енергія може стати кінетичною, коли електростатична сила примушує заряди рухатися. Розглянемо для прикладу електричне поле між двома пластинами з рівним за величиною і протилежним за знаком зарядом. Нехай розміри пластин великі порівняно з відстанню між ними, і тому поле між пластинами можна вважати однорідним (рис. 3.5).

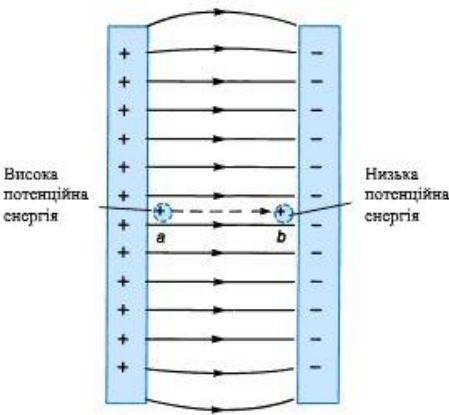


Рис. 3.5. Робота електричного поля по переміщенню заряду

Помістимо в точку  $a$  поблизу позитивно зарядженої пластини точковий позитивний заряд  $q$  (аналог предмету в гравітаційному полі). Електрична сила, що діє на заряд, буде прагнути перемістити його до негативної пластини (у точку  $b$ ), здійснюючи роботу по переміщенню заряду. Під дією сили заряд отримає прискорення і його кінетична енергія зростає; при цьому потенціальна енергія зменшується на величину роботи електричної сили по переміщенню заряду з точки  $a$  в точку  $b$  (аналогічно зменшенню потенційної енергії предмету при падінні з висоти). Внаслідок цього потенційна енергія заряду в електричному полі перейде в кінетичну енергію. Зауважимо, що позитивний заряд  $q$  має найбільшу потенційну енергію поблизу позитивної пластини, тому що в цій точці його здатність здійснювати роботу над іншим тілом або системою максимальна. Для негативного заряду справедливо зворотне: його потенційна енергія буде максимальна поблизу негативної пластини. Різниця потенційних енергій дорівнює роботі по переміщенню зарядів із протилежним знаком, її можна вимірювати. У техніці вона має назву — різниця потенціалів або напруга —  $U$ , яка вимірюється у вольтах (В).

Щоб краще зрозуміти сенс електричного потенціалу, нагадаємо аналогію з предметом, який падає з висоти. Чим вище будинок, тим більшою потенційною енергією володіє предмет і тим більше буде його кінетична енергія, коли він долетить до землі. Величина кінетичної енергії та, відповідно, робота, яку може зробити камінь, залежать від висоти будинку і від маси предмета. Точно так само і в електричному полі зміна потенціальної енергії (і робота, яку можна зробити) залежить від різниці потенціалів (еквівалентної висоті будинку) і заряду (еквівалентного масі).

Якщо пластини з'єднати будь-яким провідником, створюється електричне коло, по якому потече електричний струм, сила якого (швидкість потоку електронів)  $I$  залежить від двох факторів: різниці потенціалів між пластинами (величини електричної напруги)  $U$  та електричного опору  $R$ :

$$I=U/R \tag{3.1}$$

Опір визначається так званим *питомим електричним опором* (ρ), довжиною провідника (L) та величиною площі поперечного перерізу (S) і розраховується за формулою

$$R=\rho L/S , \tag{3.2}$$

Свою чергою питомий електричний опір залежить від типу речовини, а також від температури (T). Для його розрахунку застосовується спеціальне співвідношення, яке має такий вигляд:

$$\rho=\rho_0+ \alpha T, \tag{3.3}$$

де ρ<sub>0</sub> — питомий електричний опір при температурі 20<sup>0</sup> С.

У міжнародній системі одиниць питомий опір ρ<sub>0</sub> визначається як електричний опір 1 метра провідника (в Ом), перерізом 1 мм<sup>2</sup>, при температурі 20<sup>0</sup> С. Значення питомого опору різних металів добре відомі, деякі з них наведені в табл. 3.1.

Таблиця 3.1. Значення питомих електричних опорів металів

Речовина	$\rho_0 , \frac{Ом \cdot мм^2}{м}$
Срібло	0,016
Мідь	0,017
Золото	0,024
Алюміній	0,028
Вольфрам	0,055
Залізо	0,010
Сталь	0,013

Будь-який метал у твердому стані являє собою кристалічну ґратку, у вузлах якої знаходяться позитивно заряджені іони (атоми без електронів, які вже відірвалися) (рис. 3.6).

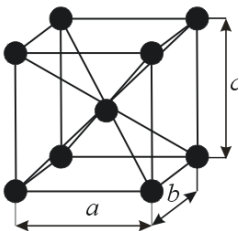


Рис. 3.6. Кристалічна структура металів

Позитивно заряджені іони постійно знаходяться в русі; природно вони впливають на рух вільних електронів, створюючи електричний опір. Тому, електричний струм у металі - це рух відірваних від орбіт електронів у полі позитивно заряджених іонів, що знаходяться у вузлах кристалічної ґратки. При підвищенні температури коливання іонів зростає, внаслідок чого змінюється електричний опір (3.3).

Для проєктів з електронного текстилю здебільшого використовуються волокна з нержавіючої сталі. До переваг струмопровідних волокон (нитки) відноситься низький електричний опір, вони м'які, але при цьому не бояться високої температури, тому нічого не загориться, навіть якщо до них випадково доторкнутися паяльником (рис. 3.7.).



Рис. 3.7. Струмопровідна нитка

Водночас продовжуються спроби розробити струмопровідні нитки на іншій основі. Наприклад, шведські вчені створили струмопровідну нитку на целюлозній основі з електропровідним біосумісним полімером. Висока провідність цих ниток може бути збільшена шляхом додавання срібних нанодротів.

### 3.3. Фізичні процеси у світлодіодах

Серед електронних компонентів розумного одягу чільне місце займають світлодіоди, в яких електрична енергія перетворюється в світлову. Можливо з точки зору створення проєктів з електронного текстилю знання фізичних процесів в компонентах електричного кола не є обов'язковими, але з позицій STEAM-освіти розуміння цих процесів є дуже корисним.

Світлодіод - це напівпровідниковий прилад, що перетворює електричний струм безпосередньо у світлове випромінювання. До речі, англійською мовою світлодіод перекладається light emitting diode, або LED. Світіння світлодіодів засновано на фізичному явищі електролюмінісценції,

тобто нетеплового спонтанного випромінювання світла під впливом електричного поля, яке було виявлено британцем Генри Раундом у 1907 році. Сьогодні існує багато видів та конструкцій світлодіодів, але всі вони мають напівпровідниковий компонент з електронно-дірковим переходом, що створює оптичне випромінювання при проходженні електричного струму в прямому напрямку. Нагадаємо, що до напівпровідників відносяться матеріали, електропровідність яких має проміжне значення між провідностями провідників та ізоляторів. Але за певних умов, а саме — під впливом домішок напівпровідники здатні перетворитися або у провідник, або в діелектрик, тобто будуть або пропускати електричний струм, або не пропускати. Технічну реалізацію цього явища було зроблено на підставі винаходу так званого *p-n* переходу, тобто зони меж двома напівпровідниками з різними типами провідності: *p*-провідність та *n*-провідність.

Найчастіше для додавання домішок різних типів використовується кремній (Si), який має на зовнішній орбіті чотири валентних електрони, які належать двом атомам (так званий ковалентний зв'язок). Якщо додати до кремнію п'ятивалентну домішку, наприклад миш'як (As), то один електрон виявиться зайвим, вільним, тобто його дуже легко “відірвати” від орбіти, після чого він під впливом зовнішніх факторів (електричного поля) почне рухатися. Так створюється провідність *n*-типу, а додані домішки називаються донорними.

Напівпровідник *p*-типу характеризується дірковою природою провідності. Для того, щоб кремній мав діркову провідність, до чотиривалентного кремнію додають невелику кількість атомів тривалентного індію (In) - домішку. Атоми індію встановлюють ковалентний зв'язок з трьома сусідніми атомами кремнію. Але у кремнію залишається один вільний електрон в той час, коли у атома індію немає валентного електрона. Тому останній захоплює валентний електрон з ковалентного зв'язку між сусідніми атомами кремнію і стає негативно зарядженим іоном, утворюючи так звану дірку і, відповідно, діркову провідність. Підсумовуючи сказане вище, можна зробити висновок: як кремній *n*-типу, так і *p*-типу мають здатність проводити електричний струм; просто в одному типі кристалу *основними носіями заряду* є надлишкові вільні електрони (*n*-тип кремнію), а в іншому – надлишкові

позитивні дірки (р-тип кремнію).

Товщина шару р-п-переходу в напівпровідниках складає приблизно  $10^{-6}$  –  $10^{-7}$  м, а контактна різниця потенціалів – десяті частки вольт. Носії струму здатні подолати таку різницю потенціалів лише за температури в декілька тисяч градусів, тобто за звичайних температур рівноважний контактний шар є замикаючим (*характеризується підвищеним опором*).

Тепер розглянемо що відбувається, коли до р-п переходу подається напруга. Якщо прикладене до р-п-переходу зовнішнє електричне поле направлено від n-напівпровідника до р-напівпровідника (рис. 3.10а), тобто співпадає з полем контактного шару, то воно викликає рух електронів в n-напівпровіднику і дірок в р-напівпровіднику від межі р-п-переходу в протилежні сторони. В результаті замикаючий шар розшириться і його опір зросте. <https://studfile.net/preview/5366662/page:63/>

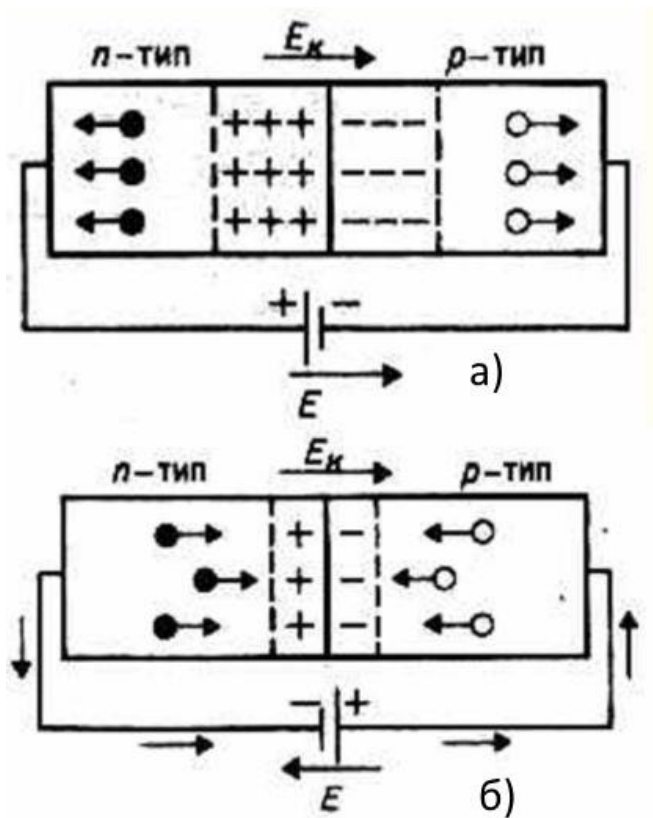


Рис. 3.10. Зона р-п переходу в електричному полі



Напряг зовнішнього поля, що розширює замикаючий шар, називається **замикаючим (зворотним)**. У цьому напрямі електричний струм через  $p$ - $n$ -перехід практично не проходить. Струм у замикаючому шарі в замикаючому напрямі утворюється лише за рахунок неосновних носіїв струму (електронів у  $p$ -напівпровіднику і дірок у  $n$ -напівпровіднику).

Якщо прикладене до  $p$ - $n$ -переходу зовнішнє електричне поле направлене протилежно полю контактного шару (рис. 3.10б), то воно викликає рух електронів у  $n$ -напівпровіднику і дірок у  $p$ -напівпровіднику до границі  $p$ - $n$ -переходу назустріч один одному. У цій області вони рекомбінують, товщина контактного шару і його опір зменшуються. Отже, у цьому напрямку електричний струм проходить крізь  $p$ - $n$ -перехід у напрямку від  $p$ -напівпровідника до  $n$ -напівпровідника; цей напрям називається **пропускним (прямим)**. Отже,  $p$ - $n$ -перехід має **односторонню (вентильну) провідність**.

При нормальному режимі проходження прямого струму інжектуючі електрони та дірки беруть участь у процесі випромінювальної рекомбінації, тобто, зникнення пари вільних носіїв протилежних знаків із виділенням світлової енергії. Для того, щоб  $p$ - $n$  перехід випромінював світло, ширина забороненої зони  $p$ - $n$  переходу світлодіода повинна бути близька до енергії фотонів видимого діапазону випромінювання. Саме тому позначення світлодіодів на електричних схемах має такий вигляд (рис. 3.11):

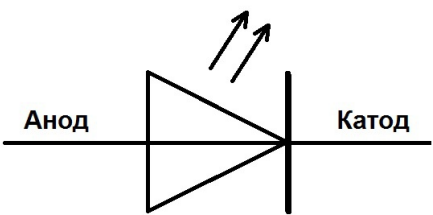


Рис. 3.11. Схематичне позначення світлодіода

Для отримання певного спектра випромінювання у світлодіодах використовують спеціальний хімічний склад напівпровідників та речовину, яка здатна перетворювати енергію, що поглинається нею, у світлове

випромінювання — люмінофор.

### 3.4. Електричне з'єднання компонентів проєкту

Перш ніж шити, підключаючи струмопровідною ниткою живлення до різних компонентів, треба визначитися зі співвідношенням потужності джерела живлення та складом і кількістю компонентів проєкту. Якщо на перших етапах розробки проєкту живлення здійснювалося від комп'ютера або від ноутбука через порт USB, то в реальних проєктах з використанням Arduino LilyPad USB або Lily Tiny або взагалі без мікроконтролера джерелом живлення є літієва батарейка CR2032 (рис. 3.12).



Рис. 3.12. Літієва батарейка CR2032

Спочатку про позначення CR2032: С — літієва, R — кругла, 20 — діаметр (мм), 32 — товщина (мм). Номінальна напруга складає 3 В (нагадаємо, що напруга, яку видає комп'ютер проєктам з Arduino, складає 5 В). Ємність батареї є однією з її найважливіших технічних характеристик, це максимальна кількість електроенергії, яка накопичується за повний цикл зарядки. Одиницею вимірювання ємності є А·г (ампер-година), для невеликих батарей — мА·г (міліампер-година). Ємність батарейки CR2032 складає 210-240 мА·г. Саме цей показник поряд з величиною струму, який протікає в електричному колі проєкту, впливає на тривалість роботи батарейки. Щоб дізнатися, як надовго вистачить батареї, можна скористатися такою формулою:

$$\text{Години} = C \text{ (мА}\cdot\text{г)} / I \text{ (мА)},$$

де С — ємність джерела живлення; I — струм проєкту.

Навіть інтуїтивно зрозуміло, що ця тривалість залежить від кількості елементів у проєкті. Наприклад, приблизний розрахунок тривалості роботи батарейки CR2032 у проєктах тільки зі світлодіодами дав такі результати:

Число світлодіодів	1	2	5	10	20
Тривалість (год)	8,3	5,0	2,3	1,2	0,61

Так, батарейка CR2032 забезпечує надійну роботу проєктів з електронного текстилю, де використовується 20 світлодіодів, лише протягом півгодини. Можливо цього вистачить для проєкту “новорічне вбрання” або для демонстрації на заняттях. Але, якщо проєкт потребує тривалої роботи, слід використати батарейки більшої ємності або акумулятор, який можна перезаряджати (при цьому потрібно забезпечити належне кріплення акумулятора на текстилі). Додамо, що у лінійці Arduino для електронного текстилю є плата LilyPad Arduino SimpleSnap, яка має вбудований літій-полімерний акумулятор, який можна заряджати, просто підключивши перехідну плату FTDI для програмування (рис. 3.13).

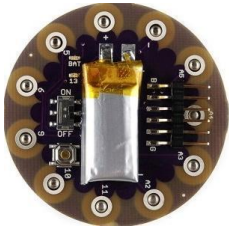


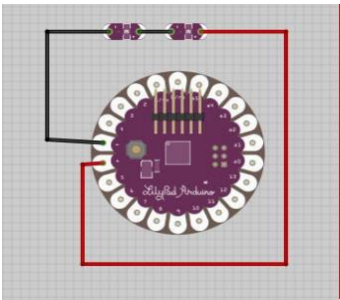
Рис. 3.13. Плата Arduino LilyPad SimpleSnap

Вище вже йшлося про те, що кожна плата LilyPad має специфічні потреби в робочій напрузі, діапазон якої вказано в технічному паспорті. Якщо до електричного кола подається недостатня напруга, проєкт може неправильно працювати, що проявляється в некоректних показаннях датчиків у послідовному порту, дивній поведінці світлодіодів тощо. Навпаки, якщо коло отримує напругу вище рекомендованого діапазону, можна пошкодити плату або суттєво скоротити час роботи. Тому потрібно переконатися, що обране джерело живлення знаходиться у рекомендованому діапазоні живлення плати.

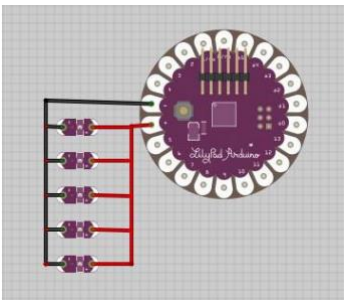
Крім перевірки робочої напруги плат LilyPad у проєкті, треба перевірити поточний струм, який буде отримувати проєкт, порівняно з тим, який може забезпечити джерело живлення (батарейка або акумулятор). Для більшості плат LilyPad Arduino це 40 мА, але у будь-якому випадку краще

припустити, що конкретному проєкту буде потрібно більше струму, ніж не забезпечувати достатнього струму. Далі буде показано як вимірювати напругу, силу струму та електричний опір кола за допомогою мультиметра.

Тепер розглянемо питання щодо електричного з'єднання компонентів. У побудові електричних схем існує два типи конфігурацій з'єднання електричних компонентів: паралельне і послідовне з'єднання. Послідовне з'єднання, наприклад, світлодіодів відбувається шляхом з'єднання позитивного полюсу одного елемента з негативним полюсом наступної плати (рис. 3.14а). Послідовне з'єднання характеризується тим, що через усі елементи електричного кола протікає однаковий струм. Внаслідок цього загальний електричний опір кола зростає з кожним новим світлодіодом, тому потребує більш потужне електричне джерело. Крім того, вихід з ладу навіть одного компоненту призводить до руйнування всього електричного кола. Внаслідок цього паралельне з'єднання є більш енергоефективним (рис. 3.14б). У паралельному з'єднанні всі негативні контакти світлодіодів об'єднуються з негативним виходом (GND) плати Arduino LilyPad, всі позитивні — з одним з контактів, на який подається відповідний сигнал.



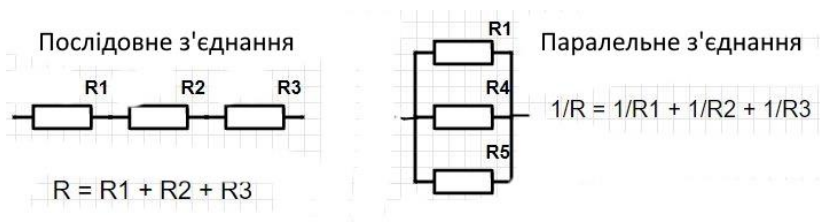
а). Послідовне з'єднання



б). Паралельне з'єднання

Рис. 3.14. Типи електричних з'єднань компонентів

Формули розрахунку електричного опору в послідовному та паралельному з'єднанні:



### Закони послідовного з'єднання резисторів:

- сила струму однакова на всіх ділянках кола;
- загальна напруга кола дорівнює сумі напруг окремих ділянок;
- загальний опір кола дорівнює сумі опорів окремих ділянок кола.

### Закони паралельного з'єднання резисторів:

- сила струмів на окремих ділянках дорівнює загальному струму кола;
- напруга на всіх ділянках кола є однаковою;
- величина, обернена загальному опору кола, дорівнює сумі величин обернених опорів.

## 3.5. Використання мультиметра для виміру параметрів кола

Для вимірювання параметрів електричного кола (напруга, сила струму, опір) можна використовувати мультиметр. Сьогодні існує багато типів мультиметрів, призначених для роботи з електричними та електронними приладами. Для роботи з виробами з електронного текстилю достатньо використовувати широко розповсюджений мультиметр DT33 (рис. 3.16).



Рис. 3.16. Мультиметр DT33B

структура  
рис. 3.17.



- ЗНАЙОМСТВО З ПРИБОРОМ ПЕРЕДНЯ ПАНЕЛЬ
- 1. ЖК дисплей.
  - 2. Кнопка HOLD, утримує дані на дисплеї
  - 3. Кнопка включення підсвічування дисплея.
  - 4. Поворотний перемикач

Рис. 3.17 Функціональна структура мультиметра DT33B

Умовні позначення мультиметру представлені у таблиці:

	Постійна або змінна напруга
	Змінна напруга
	Постійна напруга
	Земля
	Подвійна ізоляція
	Індикатор розрядженої батареї
	Діод
	Запобіжник
	Зумер
	Зверніться до інструкції
CE	Відповідність європейському стандарту безпеки

Перед початком вимірювань потрібно ознайомитися з правилами проведення вимірювань:

- Уважно огляньте прилад перед початком вимірювань. Переконайтеся, що прилад знаходиться у справному стані та не має зовнішніх пошкоджень корпусу. Не використовуйте прилад за наявності яких-небудь ознак несправностей: пошкоджень на корпусі приладу,

пошкодженої ізоляції терміналів на лицьовій панелі та ін.

- Огляньте вимірювальні щупи і переконайтеся, що їхня ізоляція не порушена. Якщо щупи несправні, замініть їх на нові з відповідними технічними параметрами.
- Щоб уникнути пошкодження приладу забороняється змінювати положення поворотного перемикача функцій під час проведення вимірювань.
- При проведенні різних вимірів слідкуйте за правильністю вибору положення поворотного перемикача функцій.
- При роботі з вимірювальними щупами намагайтеся не торкатися їхніх металевих частин.
- Перед вимірюванням опору, сили струму та тестуванням діодів і електричних кіл на обрив вимкніть живлення кола, що тестується, і розрядіть усі високовольтні конденсатори.
- Перед вимірюванням сили струму переконайтеся у справності плавких запобіжників приладу та вимкніть живлення кола, що тестується.
- При першій появі на дисплеї індикатора розрядженої батареї замініть стару батарею на нову.
- Для очищення приладу використовуйте вологу матерію. Не використовуйте миючі засоби, що містять розчинники та хімікати.
- Ці прилади призначені для використання всередині приміщення.
- Видаліть батарейки, якщо мультиметр не використовуватиметься протягом тривалого відрізка часу.
- Регулярно перевіряйте цілісність батарейки, якщо вона потече, хімікати можуть пошкодити схему приладу.

Порядок проведення вимірювань такий:

а). Вимірювання постійної напруги

Діапазони вимірювання постійної напруги: 200 мВ, 2000 мВ, 20 В, 200 В, 500 В.

Для проведення вимірювань постійної напруги:

1. Приєднайте чорний вимірювальний щуп до виводу COM і червоний щуп до виводу VWmA.
2. Установіть поворотний перемикач функцій в положення V, залежно від діапазону вимірювань.
3. Приєднайте щупи паралельно до джерела напруги, що тестується. Зніміть показання на дисплеї.

Примітка. Після завершення вимірювань постійної напруги відключіть вимірювальні щупи від навантаження і від входних гнізд приладу.

#### б). Вимірювання сили постійного струму

Модель DT-33B має діапазон вимірювань 200 мкА, 200 мА і 10 А. Моделі DT-33C / DT-33D мають діапазон вимірювань 2000 мкА, 20 мА, 200 мА і 10 А.

Для вимірювання сили струму:

1. Вимкніть живлення кола, що тестується. Розрядіть усі високовольтні конденсатори.
2. Приєднайте червоний щуп до виводу VWmA або 10 А, а чорний щуп до виводу COM.
3. Установіть поворотний перемикач функцій у положення V, залежно від діапазону вимірювань.
4. Розблокуйте коло, що тестується. Приєднайте червоний вимірювальний щуп до позитивно зарядженої ділянки кола, а чорний - до негативно зарядженої.
5. Увімкніть живлення кола. Зніміть показання на дисплеї.

Примітки:

Якщо при проведенні вимірювань перегорить запобіжник, це може



пошкодити прилад і привести до небезпечної ситуації, викликати загрозу для життя. Слідкуйте за правильністю вибору терміналів вимірювань, робочих функцій, а також діапазону вимірювань. Уникайте використання щупів послідовно до навантаження при вимірюванні сили струму.

#### в). Вимірювання опору

Модель DT-33B має наступні діапазони вимірювань: 200 Ом, 2000 Ом, 20 кОм, 200 кОм і 20 мОм. Модель DT-33D має наступні діапазони вимірювань: 200 Ом, 2000 Ом, 20 кОм, 200 кОм, 20 мОм і 200 мОм.

Для вимірювання опору:


1. Приєднайте червоний щуп до виводу VHzmA і чорний щуп до виводу COM на лицьовій панелі приладу.
2. Установіть поворотний перемикач функцій на необхідний діапазон вимірювань.
3. Приєднайте вимірювальні щупи паралельно резистору. Зніміть показання на дисплеї.

#### Примітки:

Перед проведенням вимірів переконайтеся, що живлення відключене і видалені батарейки з вимірюваних пристроїв і приладів. Перед проведенням вимірів опору всі конденсатори повинні бути повністю розряджені.

Після завершення вимірювань опору відключіть щупи від кола, що тестується, і від входних гнізд приладу.

#### г). Тестування світлодіодів

1. Підключіть червоний щуп до терміналу VHzmA, а чорний щуп до терміналу COM на лицьовій панелі приладу.
2. Установіть поворотний перемикач функцій у положення 
3. Для перевірки справності світлодіоду підключіть червоний щуп до анода (+), а чорний - до катода (-). Якщо світлодіод загориться, він справний.

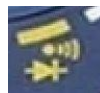
Примітки:

Щоб уникнути пошкодження приладу, перед тестуванням світлодіоду потрібно переконатися, що живлення кола, що тестується, відключено. Після завершення вимірювання діодів відключіть вимірювальні щупи від напівпровідника і від входних гнізд приладу.

д). Перевірка цілісності кола (прозвон)

Дуже часто виникає задача перевірити цілісність електричного кола, тобто чи немає його розриву. Для цього потрібно зробити такі кроки:

1. Приєднайте червоний щуп до виводу VHzmA, а чорний щуп до виводу СОМ на лицьовій панелі приладу.



2. Установіть поворотний перемикач функцій в положення

3. Для перевірки відсутності розриву підключіть червоний щуп до однієї ланки кола, а чорний — до іншої. Якщо мультиметр видасть звук (прозвон), - коло не має розривів.

## Розділ 4. Технологічні основи шиття електронного текстилю

### 4.1. Основні технологічні операції

Електричне з'єднання елементів розумного текстилю може відбуватися за допомогою дротів у ізоляції або струмопровідних ниток. Друге, звичайно ж, краще, оскільки виріб буде більш акуратним, з'явиться можливість його прання. Але тут є свої підводні камені. Якщо використовуються струмопровідні нитки для створення текстилю з великою кількістю світлодіодів, датчиків, інших елементів, то електрична схема ускладнюється. Особливо багато ниток потрібно, якщо проєкт містить різні групи світлодіодів по-різному запрограмованих. Тому розглянемо основні технологічні прийоми з'єднання елементів проєкту струмопровідними

нитками. Найпростіший із них — миготіння світлодіоду із живленням від батарейки напругою 3В, яка розташована у тримачі (рис. 4.1).

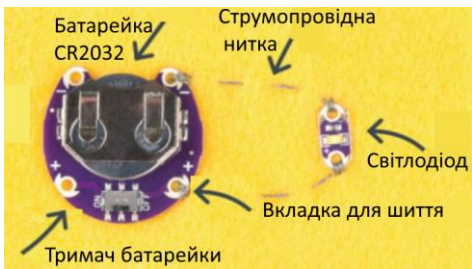
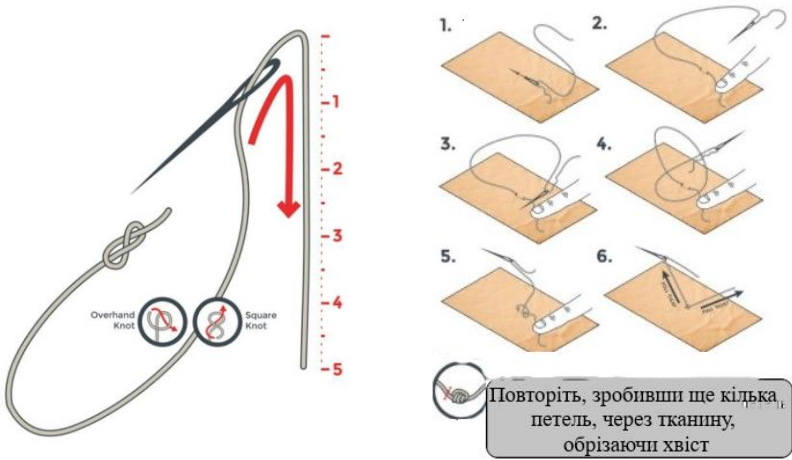


Рис. 4.1. Проєкт “Миготіння світлодіоду”

Тримач батарейки, як і всі інші елементи платформи LilyPad, має великі отвори з провідними срібними прокладками, які називаються пришивними вкладками. Ці вкладки призначені для того, щоб дати місце для проходження голки та нитки кілька разів. Перш ніж почати зшивати схему, слід визначити шви, які треба з'єднати, та зорієнтувати їх так, щоб вони були легко доступні в дизайні. Рекомендується використовувати маленьку крапку гарячого клею (бажано) або клею для тканини, щоб прикріпити кожен шматочок LilyPad до тканини, щоб вона не рухалася під час шиття. Слідкуйте за тим, щоб випадково не закрити отвори на швах.

Перш ніж приступити до шиття вашого проєкту, потрібно зав'язати вузол на довгому кінці нитки, щоб уникнути повного протягування її через тканину (рис. 4.2.). Можна зав'язати простий вузол або квадратний вузол. У наступних розділах буде пояснено кілька

інших  
методів  
вузлів.



а)

б)

Рис. 4.2. Створення вузла на голці (а) та перші кроки шиття (б)

Можливо краще використати так званий вузол Килтера - це спосіб зав'язати швидкий, надійний вузол на нитці. Після деякої практики цей вузол можна зав'язати дуже швидко (рис. 4.3).

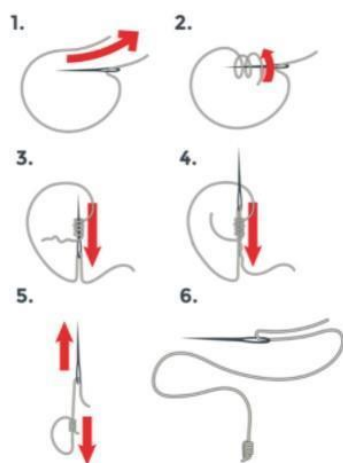


Рис. 4.3. Вузол Килтера

Для того, щоб зробити міцний електричний і фізичний зв'язок між ниткою і швом, важливо робити три-чотири петлі кожного разу, коли треба з'єднати свою нитку з порожньою вкладкою для зшивання і натягувати нитку щільно з кожним проходом. Треба міцно натягнути петлю, перш ніж продовжувати строчку (рис. 4.4).



Рис. 4.4. Затягування петлі

Після того, як будуть прострочені петлі навколо вкладки для зшивання, **біговий стібок** дозволить з'єднати частини LilyPad струмопровідною ниткою.

Далі представлено послідовність дій пришивання компонента проєкту (рис. 4.5).

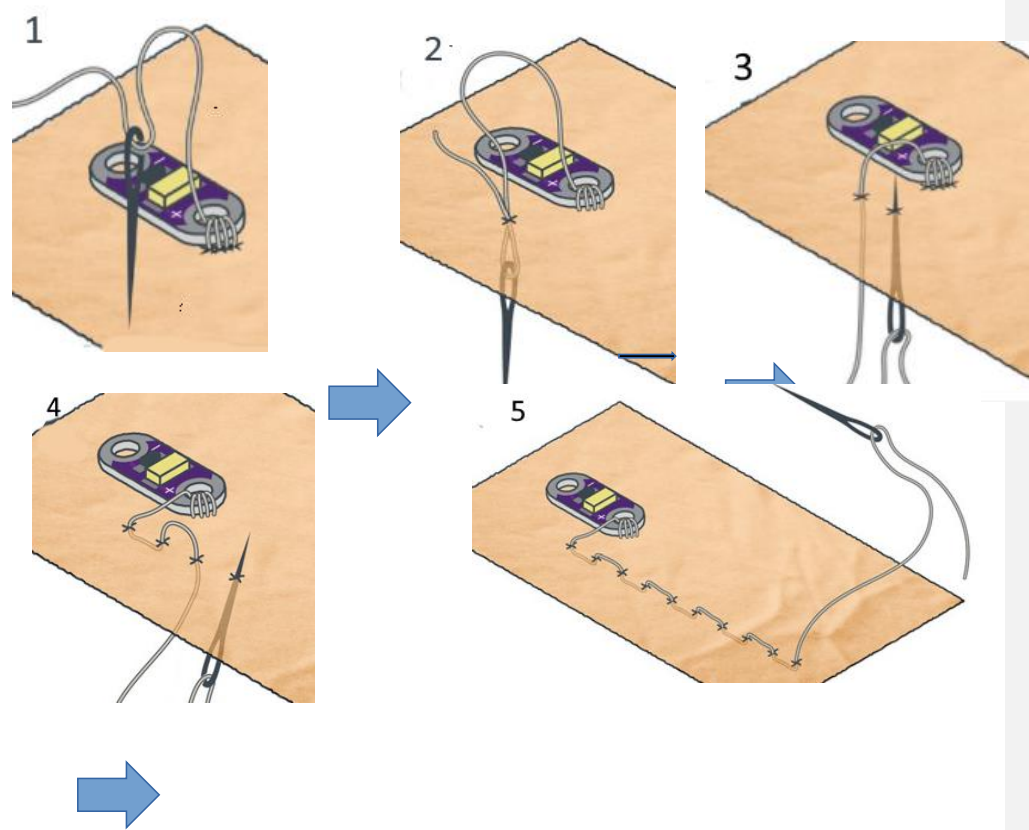
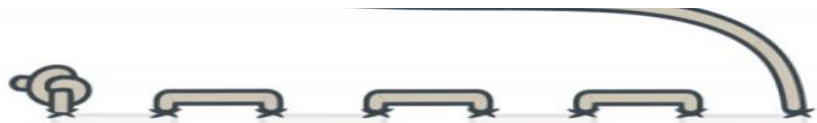


Рис. 4.5. Послідовність дій пришивання компонента проєкту

В основному поточному стібку зшивання буде навіть на обох сторонах тканини (рис. 4.6а). Щоб приховати стібки так, щоб їх не було видно, роблять довший стібок на тильній стороні проєкту і дуже маленький стібок на лицьовій стороні тканини. Цей метод називається



"прихованим стібком" (рис. 4.6.).

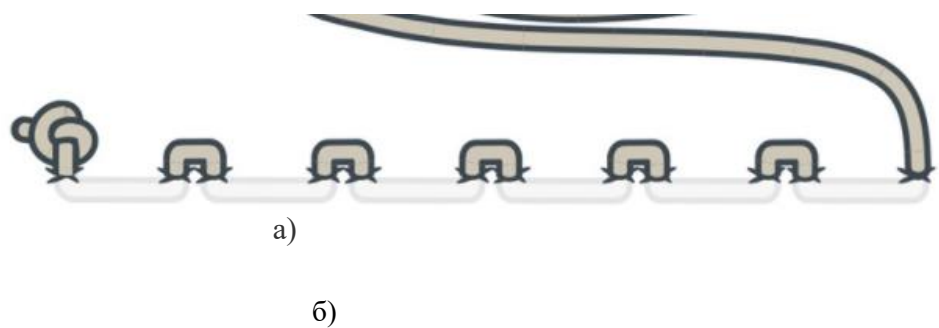


Рис. 4.6. Поточний (а) та прихований стібки (б)

Впродовж шиття час від часу треба перевертати тканину, щоб переконатися, що струмопровідна нитка не заплутується. Зрозуміло, що потрібен певний час, терпіння та досвід шиття струмопровідними нитками, перш ніж воно стане комфортним та легким. Якщо нитка обривається, її можна пришити до струмопровідної нитки, щоб продовжити електричне з'єднання.

І поточні, і приховані стібки забезпечують з'єднання компонентів LilyPad, використовуючи довжиною однієї струмопровідної нитки. Щоб з'єднати дві частини LilyPad, продовжуйте шити, пройшовши три-чотири петлі навколо вкладки для зшивання (рис. 4.7).

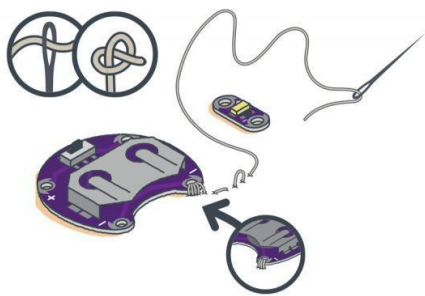


Рис. 4.7. Шиття двох компонентів

Щоб з'єднати більше двох частин LilyPad, замість того, щоб обрізати

Formatted: Font: (Default) Times New Roman, Font color: Black, Highlight

Formatted: Font: (Default) Times New Roman, Highlight

Formatted: Font: (Default) Times New Roman, Font color: Black, Highlight

нитку і починати заново, продовжуйте шити до наступної частини, зробіть три-чотири петлі і повторіть за необхідністю. Не треба використовувати нову нитку, якщо частини мають спільне з'єднання (рис. 4.8).



Рис. 4.8. З'єднання декількох компонентів

Після завершення з'єднання компонентів необхідно зробити фінішний вузол. Хвости ниток можуть викликати короткі замикання, тому обов'язково треба обрізати нитку (рис. 4.9).



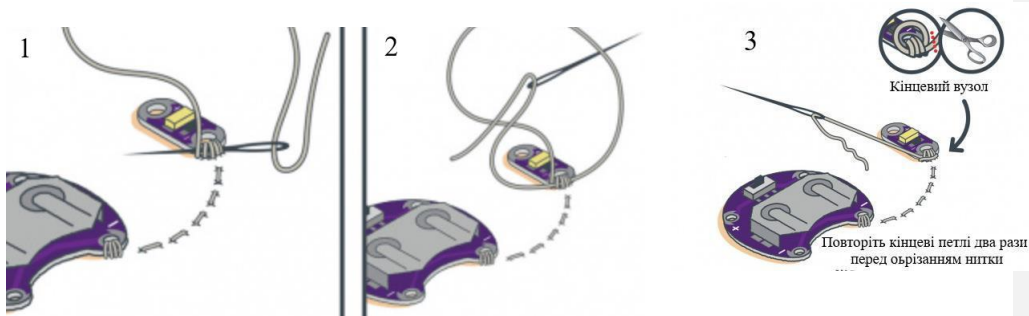


Рис. 4.9. Кінцевий вузол та останні петлі

#### 4.2. Поради щодо шиття струмопровідними нитками

Для початківців без сумніву будуть у нагоді такі поради щодо шиття струмопровідними нитками, яких слід дотримуватися.

1. Перед початком роботи накресліть схему електричного кола на папері. Відзначте плюсові та мінусові виводи. Продумайте, як прокласти нитяні доріжки від компонентів до плати з найменшою кількістю перетинів.
2. Маючи електронні компоненти розумного одягу на тканині перед роботою, можна примітати їх звичайними нитками. Використовуйте для цього п'яльці, якщо хочете, щоб робота виглядала акуратно, тканина не стовбурчилася і не збивалася. Можна перекреслити електричну схему на тканину крейдою, милом, олівцем.

3. Не потрібно пришивати компоненти занадто близько, щоб уникнути зайвих електричних контактів і замикання.

4. Зазвичай, виникають вузлики з виворітного боку тканини. Треба пришивати елементи проєкту так само, як пришивають гудзики, де вузлики заховують між гудзиком і тканиною: треба покласти вузлик всередину "вушка" компонента смарт-текстилю. Можна також починати шиття струмопровідними нитками з петлі тамбурного шва - це створить додаткове кріплення.

5. Якщо у виробі використовується велика кількість по-різному запрограмованих світлодіодів та інших компонентів, то перед шиттям треба приділити особливу увагу їхньому розташуванню та орієнтації на тканині. Плюсів виводи елементів, що підключаються до одного контакту, краще розташувати якомога ближче. Нитяні доріжки, що йдуть на плату, повинні бути рівними і перетинатися якомога меншу кількість разів (буде краще, якщо вони взагалі не перетинаються). Якщо ж перетину не уникнути, прослідкуйте, щоб стяжки, що накладаються один на одного, проходили під і над тканиною. У крайньому випадку можна використовувати ізолюючі матеріали для уникнення зайвих контактів (скотч, тканину).

6. Завжди слід пам'ятати, що струмопровідна нитка - це не просто нитка для шиття, а справжнісінький провідник, тобто по її тонким металізованим волокнам може і має йти струм. Тому пришивати компоненти смарт-текстилю необхідно при відключеному живленні: плата не повинна бути приєднана ні до usb-порту комп'ютера, ні до джерела живлення. Інакше нитки можуть лягти один на одного, викликати коротке замикання й навіть іскріння та дим.

7. При закріпленні струмопровідних ниток вузлики повинні бути особливо акуратними: по-перше, вони не повинні торкатися ниток і вузлів, що проходять поруч (див. Правило № 5); по-друге, слід зав'язувати їх так, щоб нитка не рвалася (не затягуйте занадто сильно), але і не "розмочалювалася" (уникайте занадто слабкого затягування вузлика). Краще зав'язувати вузлик, розділивши кінчик нитки на дві частини - тоді він буде меншим і акуратнішим. Ізолювати та зафіксувати край ниток допоможе безбарвний лак для нігтів або клей.

8. Стяжки не повинні створювати складку на тканині, їх величина повинна бути співмірна зі структурою волокна самої тканини, а також з величиною електронних компонентів, що використовуються, та відстанню між ними. Так як і сама нитка має певну товщину, рекомендований розмір стяжка - 4-6 мм.

9. При кріпленні довгастого елемента смарт-текстилю струмопровідною ниткою за "вушко" (кнопки, датчики, світлодіоди), потрібно зробити мінімум три стяжка в різних напрямках - це дозволить не тільки добре зафіксувати елемент на тканині, але і створити міцний контакт внутрішньої металізованої поверхні "вушка" й нитки.

10. Перевірка виробу на працездатність відбувається з підключеним джерелом живлення. Не можна згинати або звертати тканину через можливий контакт струмопровідних ниток між собою та з іншими елементами. Переворот тканини здійснюється при піднятті її за сусідні краї й акуратному опусканні на робочу поверхню.

Також є правила носіння розумного одягу, який використовує струмопровідні нитки. Текстиль, начинений світлодіодами та іншими компонентами, перед безпосереднім одяганням слід знеструмити. Одягнувши виріб на себе, необхідно розправити його. Тільки після цього можна подавати живлення на плату. У зворотному порядку все відбувається, коли такий розумний одяг знімається: спочатку відключаємо живлення, а потім роздягаємось. Попросіть допомоги, якщо боїтеся пошкодити розумний текстиль, знімаючи його, адже можна розтягнути і порвати нитки або дроту.

#### **4.3. Перевірка працездатності частини готового проєкту**

Насамперед треба уважно слідкувати за будь-якими вільними нитками або хвостовими вузлами у проєкті. Якщо певний шматок струмопровідної нитки з позитивної (+) сторони електричного кола випадково торкнеться негативної (-) сторони, це може спричинити коротке замикання. Коротке замикання з'єднує акумулятор (батарейку) із самим собою та оминає іншу частину проєкту, викликаючи небажану кількість струму від батареї. Зшивання безпосередньо над стібками в іншій частині схеми також може викликати коротке замикання.

Одне з найпоширеніших коротких замикань в електронному шитті трапляється, коли вільний хвіст нитки біля мінусової вкладки на тримачі батареї торкається батареї (рис. 4.10). Завжди треба перевіряти якість зшивання перед включенням проєкту.



Рис 4.10. Коротке замикання кола

Важливо, щоб шви не перетиналися один з одним і не торкалися інших частин кола. Джерела живлення, які використовуються в цих проєктах, якщо замикаються, не здатні спалити щось або вразити когось, хоча вони можуть нагріватися. Найчастіші причини коротких замикань представлені на рис. 4.11.

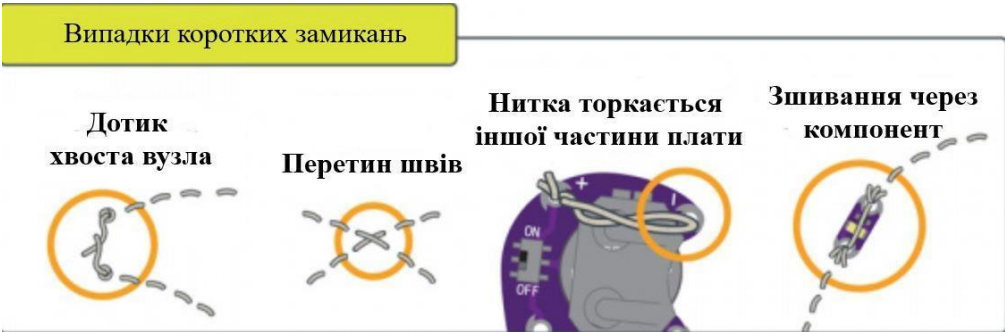


Рис. 4.11. Причини коротких замикань

Після того, як всі частини з'єднані провідною ниткою, готовій схемі потрібно джерело живлення. Для цього треба вставити монетну батарею позитивною (+) стороною вгору в тримач батареї. На зображенні нижче показано, як помістити батарею у тримач для монетних батарей CR2032 (рис. 4.12).

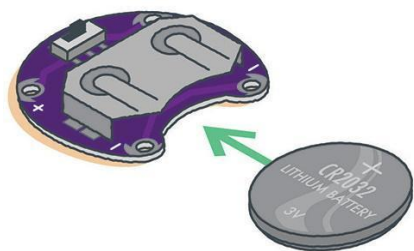


Рис. 4.12. Вставка батареї в тримач

Увага! Завжди виймайте акумулятор, якщо вам потрібно продовжити роботу над проєктом, щоб уникнути пошкодження компонентів!

Після завершення провідникових з'єднань треба увімкнути схему і подивитися, що відбувається. Якщо коло не працює, можливо, є коротке замикання, нещільне з'єднання, невірне підключення компонентів або навіть щось таке просте, як розряджена батарея.

Важливо звернути увагу на можливість вмикання/вимикання тримача батарейки - при вмиканні струм тече по провідній нитці до компонентів проєкта (рис. 4.13).

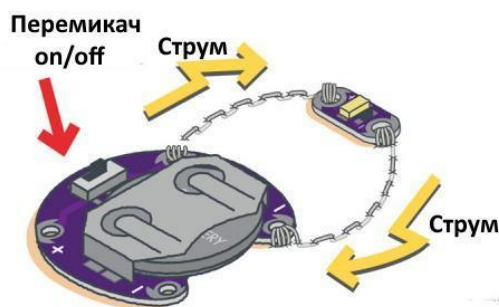


Рис. 4.13. Перемикач струму

Під час роботи над проєктами електронного шиття можна зіткнутися з такими проблемами, як вільні з'єднання, які перешкоджають горінню світлодіодів або призводять до несправності кола. Можлива також ситуація, коли схема працює нестабільно. У цьому випадку треба перевірити наявність

неналежних з'єднань. Якщо струмопровідна нитка не має щільного з'єднання з пришивними вкладками LilyPad, струм не зможе протікати послідовно. Іноді струмопровідна нитка може відірватися від шва та вимкнути коло при необережному русі тіла. Щоб виправити це, за можливості треба міцно натягнути шви пінцетом або голкою. Можна також пришити верхню частину існуючої нитки, щоб створити більший натяг і міцно утримувати нитку на шві (рис. 4.14).

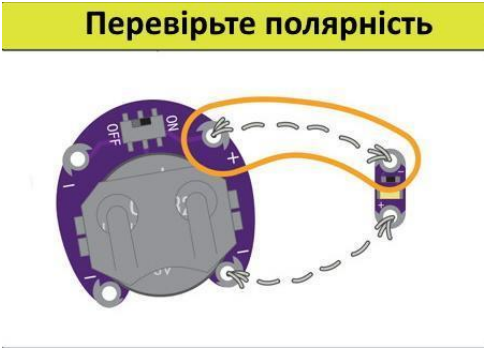


Рис. 4.14. Перевірка полярності з'єднання

Інші проблеми, на які слід звернути увагу:

- Переконайтеся, що вимикач на тримачі батареї знаходиться в положенні ON.
- Перевірте, чи розряджений акумулятор. Зробити це можна за допомогою мультиметра. Спробуйте встановити запасну батарею.
- Якщо ви дотримуетесь шаблону проекту, двічі перевірте, чи компоненти з'єднані разом у правильній конфігурації.
- Якщо у вас все ще виникають проблеми, треба скористатися мультиметром, щоб перевірити цілісність схеми або наявність інших проблем зі схемою.

**Важлива порада з ізоляції**

Щоб уникнути випадкових коротких замикань після зшивання та випробування проекту, рекомендується покривати нитку тонким шаром клею для тканини, фарби для тканини або додаткового шару тканини. Це особливо важливо для проектів, які

**Розділ 5. Приклади проектів на платформі Arduino LilyPad**

5.1. Простий проєкт без програмування — “Ілюмінована шпилька”

Цей проєкт є прикладом базової схеми - електричного кола, по якому проходить електричний струм від джерела живлення до компонента (або компонентів), а потім назад до джерела живлення. Для цього проєкту використовується світлодіод (рис. 5.1).

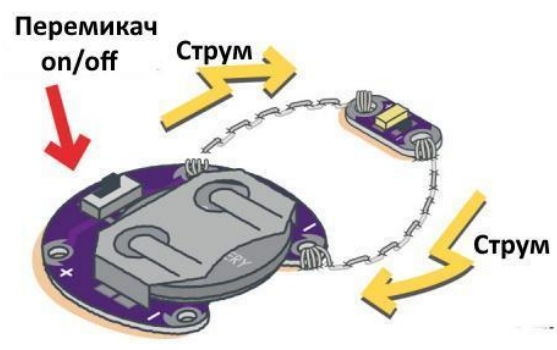


Рис. 5.1. Загальний вигляд проєкту «Ілюмінована шпилька»

Для цього проєкту взагалі не потрібна плата Arduino LilyPad, світлодіоди загоряються безпосередньо від батарейки. Для виготовлення проєкту потрібні:

Тримач для монетної батареї	Ручка або маркер
Монетна батарея 3 В ( CR2032)	Тканина для малювання дизайну
Світлодіод	Ножиці
Струмопровідна нитка	Клей
Голка	
Білий фетр (не менше 7 кв. см.)	
Шпилька	

Крок 1. Підготовча робота

Зображення на шпильці, що підсвічується, можна взяти із шаблонів, або придумати самим (рис. 5.2).



Рис. 5.2. Шаблони для проєкта «Люмінована шпилька»

Насамперед треба правильно розташувати складові проєкту — джерело живлення та світлодіод. Металеві вкладки (“вушка”) тримач батареї для зшивання мають маркування: або позитивне (+), або негативне (-). Багато електронних компонентів також мають полярність, тобто електричний струм може протікати через них тільки в одному напрямку. Якщо приєднати неправильно, вони або не працюватимуть, або зламаються. Батарея також має позитивну і негативну сторони. Завжди треба уважно перевіряти маркування на LilyPad-компонентах, щоб переконатися, що вони правильно орієнтовані, перш ніж зшити коло (рис. 5.3а). У цьому проєкті світлодіод встановлюється обличчям до тканини, щоб просвічувати її з іншої сторони (рис. 5.3б).

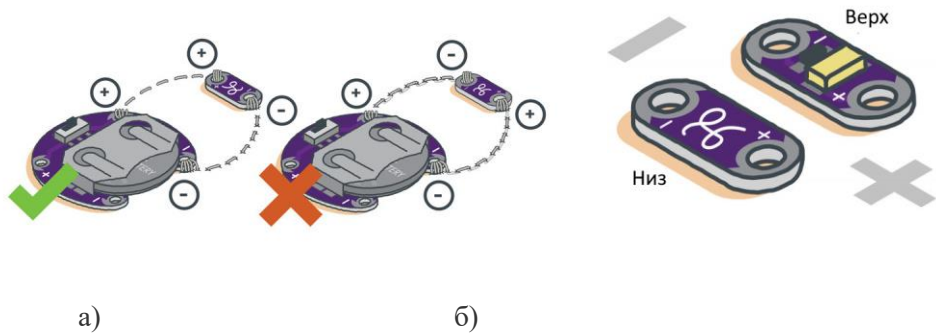


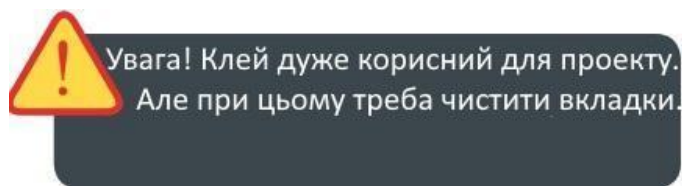
Рис. 5.3. Полярність компонентів

**Крок 2. Створення електричного кола**

Помістіть тримач батареї з перемикачем ON/OFF у ліву сторону на



нижні дві прошивальні вкладки близько до нижнього краю фетру. Рекомендується капнути невелику краплю гарячого клею в центр тримача, щоб прикріпити його до фетру. Приклеювати тримач батареї треба так, щоб залишилося місце для розміщення світлодіода на фетрі. Плануючи розміщення світлодіода, зверніть увагу, щоб він був трохи вище центру або в напрямку до верхньої половини тканини і не торкався та не перекривав тримач батареї.



Увага! Клей дуже корисний для проекту.  
Але при цьому треба чистити вкладки.

Лише для цього проекту світлодіод має бути зверненим до фетру, що забезпечує просвічування з іншого боку. На задній частині світлодіода є *℄* курсив, який має бути зверненим до вас (рис. 5.4). Для решти проєктів у цьому посібнику світлодіод встановлюється об'єктивом вгору (подалі від фетру).



Рис. 5.4. Зворотний бік світлодіода

**Крок 3. Зшиваємо разом.**

Відріжте довгий шматок струмопровідної нитки, протягніть його в голку і зав'яжіть на кінці вузол. Настав час приєднати світлодіод до тримача батареї за допомогою провідної нитки. Одна лінія зшивання з'єднає позитивну (+) сторону тримача батареї з позитивним кінцем світлодіода. Друга лінія зшивання з'єднає негативні (-) сторони плати завершить схему (рис. 5.5).

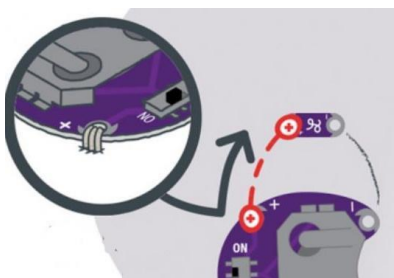


Рис. 5.5. Зшивання тримача і світлодіода

**Крок 4. Обрізання нитки**

Завершіть свою першу лінію зшивання, зав’язавши фінішний вузол на вкладці зшивання та обрізавши зайву нитку (рис. 5.6).

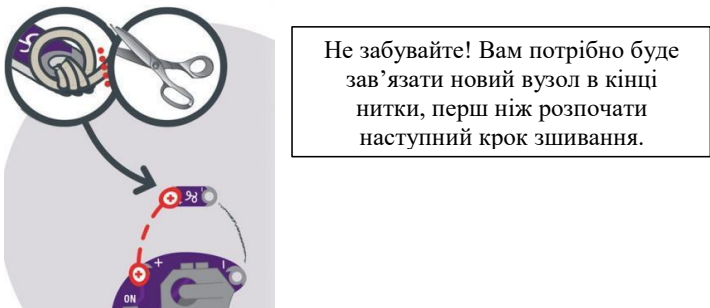


Рис. 5.6. Обрізання зайвої нитки

**Крок 5. Зшивання негативних полюсів**

Повторіть процес із новим відрізком нитки, щоб з'єднати негативну сторону тримача батареї з негативним кінцем світлодіода. Будьте обережні, щоб стежки не торкалися нитки, яка використовувалась для позитивних з'єднань, оскільки це призведе до короткого замикання. Перед тестуванням обріжте хвости ниток. Тепер схема завершена!.

Після завершення роботи потрібно виийняти батарею із тримача. Далі потрібно використати гарячий клейовий пістолет або нитку, щоб прикріпити тканину до фетрового кола, щоб світлодіод просвічував. Тепер треба перевернути проєкт і прикріпити клеєм шпильку, щоб завершити свій виріб, який можна носити.

**5.2. Проєкт без програмування “Підсвічена маска”**

У цьому проєкті зробимо новорічну ілюміновану маску, створену із фетру, джерела живлення (батарейка CR2032), трьох світлодіодів та струмопровідної нитки. На відміну від попереднього проєкту тут світлодіоди з'єднані паралельно.

Для створення цього проєкту використовуються такі матеріали:

Тримач для монетної батареї	Ручка або маркер
Монетна батарея 3В ( CR2032)	Тканина для малювання дизайну
3 світлодіоди	Ножиці
Струмопровідна нитка	Клей
Голка	Еластичний шнур
Білий фетр (не менше 7 кв. см.)	Фетр (з одного аркуша “9х12” вийде одна маска; спробуйте змішати кольори для більш святкової маски)
Шпилька	Пістолет для гарячого клею (з додатковим клеєм)

Створення проєкту “Підсвічена маска” передбачає таку послідовність дій.

**Крок 1. Підготовчий етап.** Створення шаблону, враховуючи розміри власного обличчя. Існує дві частини шаблону: верхній шар і нижній шар. Треба піднести шаблон верхнього шару до обличчя, щоб перевірити його придатність, тобто внести будь-які зміни у форму очей або носа, перш ніж наносити шаблон на фетр. За допомогою ручки або маркера треба намалювати шаблони на фетрі. Обидва шаблони розміщують на одному шматочку фетру. Електронні компоненти розміщуються на верхньому шарі, тоді як нижній шар буде використовуватися для надання масці додаткової підтримки (рис. 5.7).

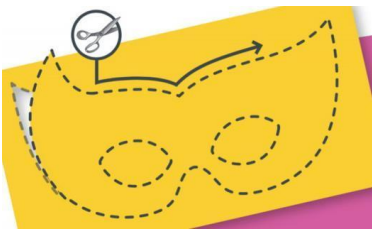


Рис. 5.7. Підготовка нижнього шару маски

**Крок 2. Підключення світлодіодів.** Щоб “запалити” маску, треба підключити світлодіоди до тримача батареї струмопровідною ниткою,

з'єднуючи їх паралельно, щоб отримати однакову кількість потужності, яка проходить через них (рис. 5.8). Кожен світлодіод у масці буде світити так само яскраво, як і інші, спільно використовуючи з'єднання з батареєю. Більшість проєктів електронного текстилю, які ми реалізуємо, використовують паралельні схеми (див. п. 3.3).

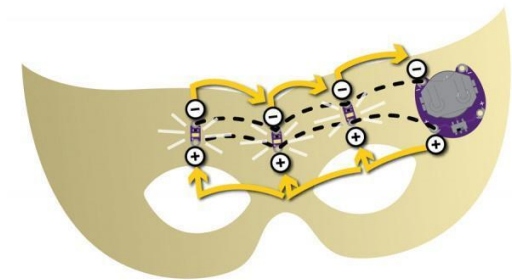


Рис. 5.8. Планування розміщення кола на масці

**Крок 3. Організація електричного кола.** Для розміщення компонентів проєкту на фетрі використовується шаблон, для чого кожен компонент закріплюється мазком клею, щоб утримувати його на місці. Треба двічі перевірити, щоб позитивні (+) виводи світлодіодів співпадали зі знаком (+) вкладки тримача батареї (рис. 5.9).



Рис. 5.9. Розміщення світлодіодів на масці

Можна використовувати ручку або маркер (але тільки не чернило), щоб намалювати лінії від одного фрагмента LilyPad до іншого, щоб надати шлях, який слід дотримуватись під час зшивання (рис. 5.10).

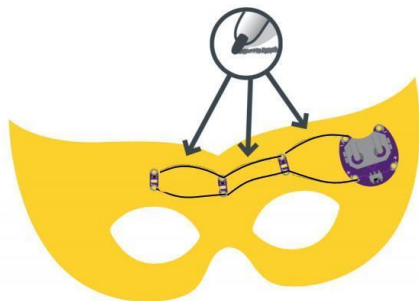


Рис. 5.10. Розмітка шляху струмопровідної нитки

**Крок 4. Зшивання разом компонентів кола. Зробіть такі дії:**

- Відріжте довгий шматок струмопровідної нитки, протягніть його в голку.
- Зав'яжіть на кінці вузол.
- Зшийте своє перше з'єднання: починаючи з позитивної зшивної вкладки тримача батареї.
- З'єднайте кожен із трьох світлодіодів біговим стібком і трьома петлями струмопровідної нитки навколо кожної позитивної (+) вкладки. Зав'яжіть і обріжте свою нитку.
- Завершіть лінію зшивання, зав'язавши фінішний вузол на останній позитивній (+) вкладці обрізавши зайву нитку (рис. 5.11).

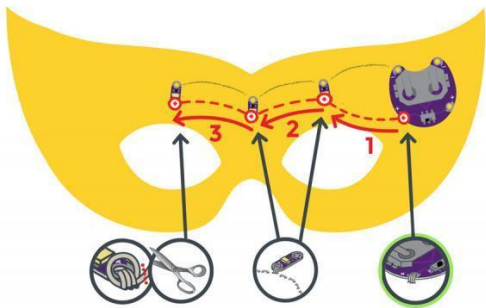


Рис. 5.11. Послідовність зшивання компонентів проєкту

**Крок 5. Повторення процесу зшивання.** З новим відрізком нитки процес повторюється для негативної частини схеми - з'єднавши негативні шви трьох світлодіодів один до одного і, нарешті, негативну вкладку на тримачі батареї. Тепер треба остаточно зав'язати фінішний вузол і обрізати зайву нитку (рис. 5.12).

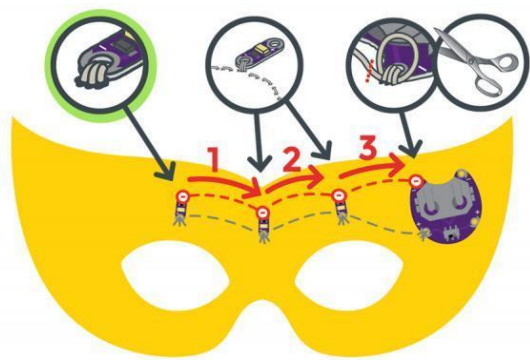


Рис. 5.12. Повторення зшивання компонентів

**Крок 6. Встановлення батареї та тестування.** Вставте монетну батарею CR2032 у тримач батареї позитивною позначкою (+) вгору. Поставте перемикач у положення ON, щоб світлодіоди загорілися.

У будь-якому проєкті з електроніки іноді трапляється, що зібрана схема не працює. Якщо світлодіоди не світяться, замініть батарею або перевірте, чи перемикач увімкнено. Перевірте свій виріб на наявність вільних ниток або кінців, які можуть торкатися інших частин кола та викликати коротке замикання.

**Завершальний крок.**

Після перевірки схеми настав час проявити творчість, додавши деякі деталі до маски. Коли батарею вийнято, приклейте другий шар фетру на задній стороні маски, щоб зробити її більш стійкою та ізолювати задню частину кола. Далі можна вирізати ножицями або перфоратором невеликі отвори по обидві сторони маски і зав'язати резинку або стрічку, щоб утримувати маску (рис. 5.13).



Рис. 5.13. Створення отворів для гумки

Нарешті можна додати блиск, фарби або інші декоративні елементи, щоб покращити зовнішній вигляд маски та приховати світлодіоди й сліди зшивання. Якщо є бажання прикрити тримач батареї на передній частині маски, можна додати будь-які прикраси. Не забудьте залишити отвір, щоб за потреби замінити батарею.

**5.3. Проєкт без програмування “Підсвічений плюш”**

У цьому проєкті будуть використані ще два компоненти, які дозволяють ручне управління процесом вмикання/вимикання світлодіодів: кнопка та перемикач (рис. 5 14).

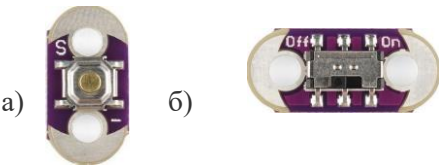


Рис. 5.14. Кнопка (а) та перемикач (б)

Для цього проєкту використовується шаблон (рис. 5.15).



Рис. 5.15. Шаблон проєкту

**Крок 1. Підготовча робота.** Треба обвести й обрізати плюшеву форму шаблону на шматку фетру. Щоб повністю приховати стібки, виріжте додаткову половинку фетру (як показано на рисунку), щоб покласти його на готовий плюш. Дві половини того, що стане вашим плюшем, з'єднані в "ногах", щоб дозволити всій вашій схемі перебувати на одній поверхні та полегшити набивання проєкту. Не розрізайте ці дві половинки.

Кнопки та вимикачі - це електронні компоненти, які керують потоком електроенергії по колу. Схема замикається при включенні вимикача або натисканні кнопки - по колу протікає електричний струм. Коли ділянка кола роз'єднується вимикачем або кнопкою, коло розривається і струм не протікає. Розглянемо пересувний перемикач LilyPad, який має позначки ON/OFF. Коли перемикач переведено у положення ON, обидві швейні вкладки на вимикачі з'єднані, що дозволяє протікати струму і розмикати коло. Якщо перевести перемикач в положення OFF, частини всередині вимикача віддаляються одна від одної і розмикають коло (рис. 5.16).

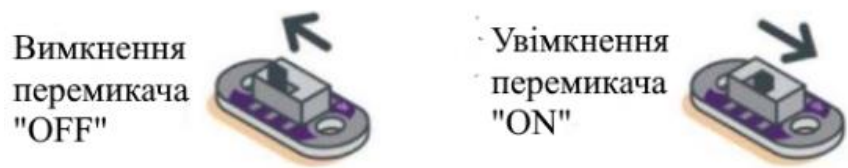


Рис. 5.16. Перемикач проєктів LilyPad

Це допомагає візуалізувати вимикачі як підйомні мости для електрики - коли міст піднятий (OFF), ніщо не може його перетнути. Коли він опущений (ON), шлях знову з'єднується і електричний струм може протікати.

Компонент “Кнопка” також є типом перемикача. При натисканні на кнопку з'єднуються дві швейні вкладки, внаслідок чого заряди починають рухатися, тобто виникає електричний струм. Якщо кнопку відпустити, кнопка повертається в початкове положення — електричний струм зникає. Ця кнопка є прикладом миттєвого перемикача: вона активна лише тоді, коли застосовується дія. Це дещо відрізняється від повзункового вимикача, який є прикладом перемикача, в якого стан залишається незмінним до чергової зміни положення повзунка.



Компоненти на фетрі розташовуються за схемою, яку представлено на рис. 5.18. При цьому треба обов’язково перевірити орієнтацію світлодіодів, перш ніж з’єднувати їх. Позитивні вкладки світлодіода (+) підключаються до кнопки або до перемикача, а негативні - до мінусової вкладки (-) на тримачі батареї. Коли ваша схема буде завершена, нанесіть тампон клею на тильну сторону кожного компонента, щоб прикріпити їх до фетру (рис. 5.17).



Рис. 5.17. Електричне коло на фетрі

Якщо у проєкті є необхідність приховати стібки, можна нанести шар фетру або прикрасити нитку після створення електричного кола. Можна також скористатися прихованим стібком.

**Крок 2. Технологія шиття.** Треба відрізати довгий шматок струмопровідної нитки, протягнути його в голку і зав’язати на кінці вузол. Шиття починається з позитивного шва на утримувачі батареї, найближчого до згину, або з «ніжок» на вирізі з фетру. Як і раніше, під час шиття треба робити три-чотири петлі навколо кожної вкладки.

Щоб з’єднати позитивну прострочку на батарейній панелі з найближчою вишивкою на вимикачі рекомендується застосувати біговий

стібок або прихований стібок. Потім треба зшити три-чотири петлі навколо затискача вимикача, зав'язати вузол та відрізати нитку (рис. 5.18).

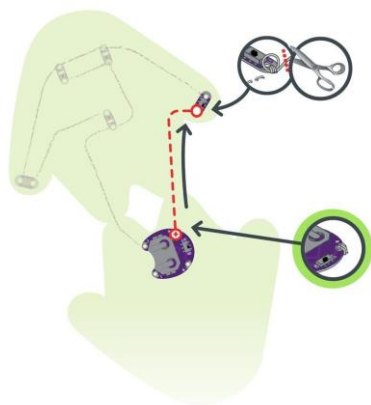


Рис. 5.18. Послідовність першого етапу шиття

**Крок 3. Зв'язування компонентів проєкту**

З новим шматочком нитки необхідно з'єднати іншу сторону вимикача з позитивними вкладками двох верхніх світлодіодів і закінчити трьома-чотирма петлями на найближчій вкладці кнопки. Зв'язати й вирізати (рис. 5.19).

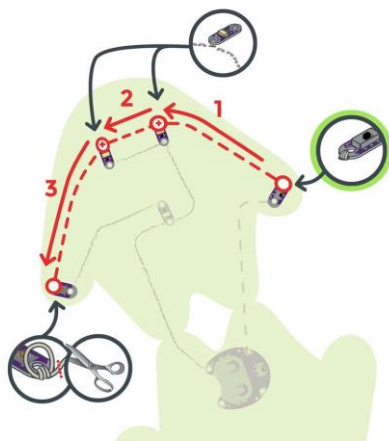


Рис. 5.19. Послідовність другого етапу зшивання компонентів

**Крок 4. Початок зворотного руху шиття.** Використовуючи новий шматок нитки треба прострочити три-чотири петлі навколо вкладки для зшивання. Далі - шити до позитивної сторони останнього світлодіода, закінчуючи трьома-чотирма петлями. Зв'язати і відрізати (рис. 5.21).

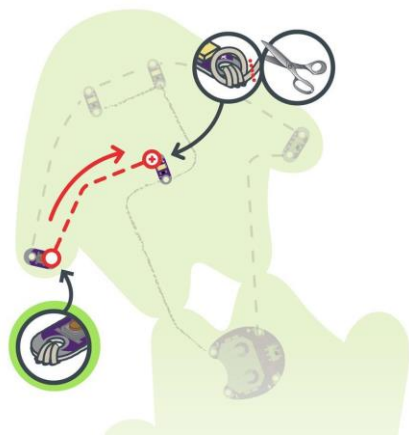


Рис. 5.21. Шиття між негативними вкладками

**Крок 6. Завершення шиття.** Нарешті, треба зшити всі негативні зв'язки. З новим шматочком нитки прострочіть три-чотири петлі на негативній (-) зшитій вкладці першого світлодіода і з'єднайте її з негативними накладками на інших світлодіодах, закінчуючи на негативній вкладці тримача батареї. Потрібно переконатися, що на кожному з'єднанні робиться три-чотири петлі (рис. 5.22).

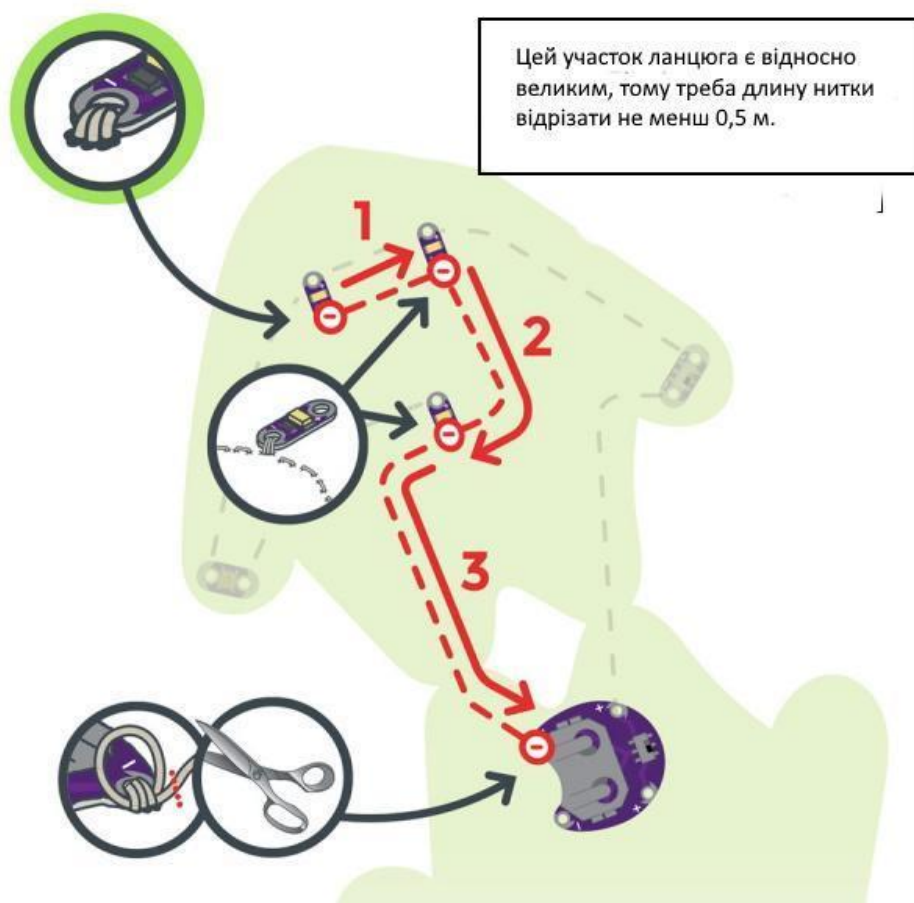


Рис. 5.22. Завершальний етап шиття

Після завершення шиття треба уважно переглянути схему, про що йшлося вище.

#### 5.4. Проєкт “Програмоване включення світлодіодів”

У цьому проєкті необхідне включення 5 світлодіодів за певною програмою. Якщо поглянути на плату Arduino LilyPad USB, на ній розташовано 9 вкладок, з яких 5 є цифровими, а 4 — аналоговими. Серед цифрових контактів виводи 3, 9, 10, 11 можуть працювати в режимі широтно-імпульсної модуляції (ШІМ), тобто світлодіоди, які підключені до цих контактів можуть програмно змінювати напругу (одночасно яскравість) від 0 до 3.3 В. Аналогові виводи можуть бути використані як цифрові (рис. 5.23).

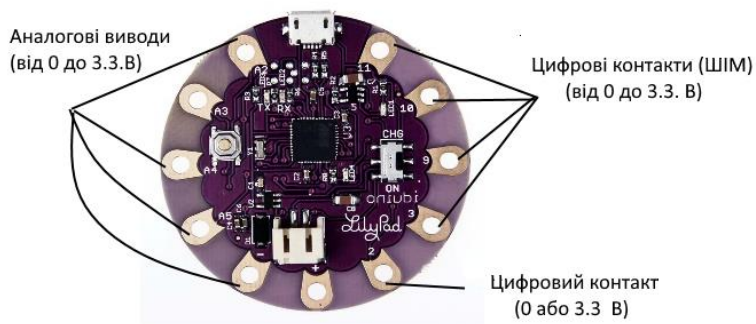


Рис. 5.23. Контакти плати Arduino LilyPad USB

Початок шиття починається із вкладки-контакту A5 (рис. 5.24).

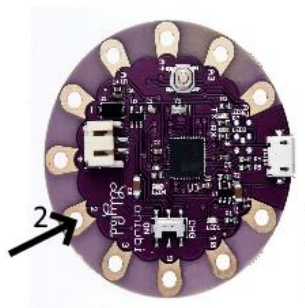


Рис. 5.24. Контакт-вкладка 2 плати LilyPad USB

Тепер треба прошити струмопровідну нитку до першого світлодіода та від цієї точки до того місця, де ви хочете розмістити перший світлодіод. Потім пришийте позитивну вкладку світлодіода (+) та електрично з'єднайтеся з ним (рис. 5.25).

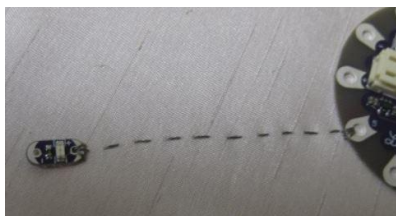


Рис. 5.25. Перше з'єднання

Після цього треба закріпити нитку та обрізати її. Можна перейти на зворотну сторону плати, щоб надійно закріпити світлодіод з обох сторін. Далі операція повторюється для кожного чергового світлодіода. Нарешті треба пришити вкладку з позначкою (-), яка маркована як «земля», і прострочити струмопровідну нитку від неї до знака (-) на найближчому світлодіоді. І далі — до відповідних вкладок-контактів кожного світлодіода, намагаючись ніколи не перетинати доріжку (+) з доріжкою (-) (рис. 5.26).

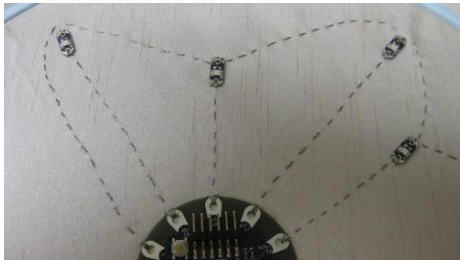


Рис. 5.26. Назва

Далі розглянемо програму ввімкнення/вимкнення світлодіодів. Оберемо найпростішу програму “Біжить вогонь”, яка вмикає та вимикає світлодіоди спочатку “зліва направо”, а потім навпаки. Можна також запрограмувати випадкове включення світлодіодів.

Розглянемо детальніше підходи до програмування проєкту. У проєкті задіяні 5 світлодіодів, відповідно, всі вони мають бути підключені до цифрових виводів Arduino LilyPad – 2, 3, 9, 10, 11. Для скорочення тексту програми більш ефективно створити масив з п’яти елементів ledpin [4], який починається з 0-вого елемента. Заповнимо елементи масиву відповідними номерами виводів:

```
Int ledPin[] = {2,3,9,10,11};
```

Тепер можна звертатися до номерів виводів у циклі від 0 до 4 так:

```
for int pin = 0; pin <5; pin ++ {  
  pinMode(pin, OUTPUT);  
}
```

Враховуючи ці пояснення, програмний код управління світлодіодами у проєкті “Біжить вогонь” виглядає так:

```
Int ledPin[] = {2,3,9,10,11};
void setup(){
for (int pin = 0; pin <= 4; ++pin)
pinMode(pin, OUTPUT);
}
void loop(){
for (int pin = 0; pin <= 4; ++pin){
digitalWrite(ledPin[pin], HIGH);
delay(1000);
digitalWrite(ledPin[pin], LOW);
delay(1000);
}
for (int pin = 4; pin >= 0; --pin){
digitalWrite(ledPin[pin], HIGH);
delay(500);
digitalWrite(ledPin[pin], LOW);
delay(500);
}
}
```

**ВИСНОВКИ**

Як можна бачити з тексту посібника, який орієнтований на технологічні питання створення розумного одягу, STEAM-підхід передбачає, по-перше, програмування компонентів платформи LilyPad, а, по-друге, - більш глибоке розуміння фізичних основ роботи цих компонентів. Водночас, починати створювати проєкти розумного одягу можна з найпростіших електричних схем, які обмежуються лише джерелом живлення та кількома світлодіодами, тобто, без програмування. Після набуття певного досвіду можна приступити до програмування режимів роботи світлодіодів, мається на увазі — частота миготінь, плавна зміна яскравості тощо, застосовуючи світлодіоди різних кольорів. На наступному етапі слід експериментувати з

іншими компонентами платформи, зокрема, зумера та різних фізичних датчиків. Нарешті, слід більше заглибитися у програмування - програмування за допомогою таких понять як функції, класи та об'єкти з використанням відповідних бібліотек. Насамперед слід освоїти програмування піксельної плати LilyPad Pixel Board.

Зрозуміло, що всі ці етапи потребують наявності належного обладнання. Сьогодні (2021 рік) воно є фінансово доступним: вартість простого набору, до якого входить мікроконтролер Arduino LilyPad 32U4 USB, світлодіод, тримач батареї живлення, струмопровідна нитка складає не більше 500 грн. Його можна придбати у спеціалізованих магазинах, зокрема, [arduino.ua](http://arduino.ua) та в інтернет-магазинах, зокрема, [prom.ua](http://prom.ua).

Виникає питання щодо формату занять. Досвід певних закладів США свідчить про велику увагу до напрямку “Електронний текстиль” у закладах середньої освіти. Цікавим є досвід упровадження цього напрямку під гаслом “Інформатика через електронний текстиль”. Враховуючи цей досвід, заняття можуть проводитися на інтегрованих уроках “інформатика + трудове навчання”. Однак, у наших умовах напрям “розумний одяг” або ширше “електронний текстиль” скоріше має розвиватися у форматі позакласної освітньої діяльності або позашкільної діяльності.

У цьому аспекті важливе значення набуває створення системи творчих груп та тренінгів, на яких буде підготовлена певна кількість вчителів, здатних розвивати цей напрям в освітніх закладах та брати участь у конкурсах та STEAM-фестивалях, на яких будуть демонструватися кращі вироби.

Варто підкреслити, що на ринку дослідів та виробництв “електронного текстилю” відбуваються суттєві зрушення. Складається враження, що акцент з автономних гаджетів переміщується в бік “розумного одягу”. І це не тільки стосовно суто прикрашення одягу шляхом втілення системи світлодіодів та піксельних плат. Йдеться про насичення одягу певними датчиками, які збирають інформацію про стан тіла та реалізують зв'язок через Wi-Fi-комунікації.

Нарешті дуже важливим аргументом на користь організації навчання зі створення виробів з електронного текстилю є досвід Сполучених Штатів Америки, де розроблено багато методичної літератури з цього напрямку та проводяться заняття на уроках в старших класах.



