

References:

1. Voloshyn O.F., Mashchenko S.O. Teoriia pryiniattia rishen: Navchalnyi posibnyk. Kyiv: Vydavnycho-polihrafichnyi tsentr «Kyivskiy universytet», 2006. 304 s.
2. Zhaldak M. I., Tryus Yu. V. Osnovy teorii i metodiv optymizatsii: navchalnyi posibnyk. Cherkasy: Brama-Ukraina, 2005. 608 s.
3. Katrenko A. V., Pasichnyk V. V., Pasko V. P. Teoriia pryiniattia rishen: pidruchnyk. Kyiv: Vydavnycha hrupa BHV, 2009. 448 s. : il.
4. Bodrov V. I., Lazareva T. Ja., Martem'janov Ju. F. Matematicheskie metody prinjatija reshenij. Tambov: TGTU, 2004. 124 s.
5. Burshtejn F.V., Korol'ov Je.S. Mnogokriterial'nye zadachi prinjatija reshenij pri neopredelennosti i riske. V kn.: Teoreticheskaja kibernetika. Tbilisi, 1980. S. 156–162.
6. Emelichev V.A., Perepelica V.A. Slozhnost' diskretnyh mnogokriterial'nyh zadach. Diskretnaja matematika, 1994. Vyp. 1, 6. S. 3–33.
7. Zajchenko Ju.P. Issledovanie operacij. Nechetkaja optimizacija: Ucheb. posobie. Kyiv: VSh, 1991. 198 s.
8. Kini R.L., Rajfa H. Prinjatje reshenij pri mnogih kriterijah: predpochtenija i zameshhenija. M.: Radio i svjaz', 1981.
9. Perepelica V.A., Sergienko I.V. Issledovanie odnogo klassa celochislennyh mnogokriterial'nyh zadach // Zhurn. vychisl. matematiki i mat. fiziki, 1988. 28, № 3. S. 400–419
10. Roshhin V.A., Semenova N.V., Sergienko I.V. Voprosy reshenija i issledovanija odnogo klassa zadach netochnogo celochislennogo programmirovanija. Kibernetika. 1989. № 2. S. 42–47.
11. Shtojer R. Mnogokriterial'naja optimizacija. Teorija, vychislenija i prilozhenija. M.: Radio i svjaz', 1992.

Solving multicriteria optimization problems with the help of computer

Ishchuk A. A.

Abstract. Solving multicriteria optimization problems of individual sections of mathematical programming in practically acceptable time is only possible with appropriately selected or specially designed software via computer. Author analyzes two methods for solving multicriteria optimization problems. The choice of method is determined by the formulation of a particular optimal problem and the used mathematical model of the optimization object. It is noted that using of information - communication technology makes the process of solving optimization problems sufficiently effective and eliminates the time-consuming calculations.

Keywords: optimization theory; multicriteria optimization problem; objective function; mathematical model.

DOI 10.31392/NPU-nc.series 2.2019.21(28).11

УДК 004.85:004.451

Ю.П. Біляй

кандидат педагогічних наук, доцент

Національний педагогічний університет імені М.П. Драгоманова

КОМП'ЮТЕРНО-ОРІЄНТОВАНІ СИСТЕМИ У STEAM-ОСВІТІ

Анотація. Освіта – фундамент для успішного розвитку країни. Після настання цифрової революції, коли більшість професій неможливі без використання комп'ютерів, інформаційних технологій, мережі Інтернет. У процес навчання потрібно гармонійно вбудовувати вивчення сучасних технологій без втрати фундаментальних принципів інших дисциплін. Таке поєднання можливе з інтеграцією у дисципліни природничого циклу елементів програмування та інформаційних технологій. В статті розглянуто деякі теоретичні відомості для тих, хто починає вивчати схемотехніку. А також продемонстровано приклади використання веб сервісу Tinkercad для створення проєктів на базі плати Arduino без наявності самої плати.

Ключові слова: ардуіно, схемотехніка, програмування.

STEM-освіта – це сукупність чи послідовність курсів та програм за якими здійснюється навчання. За такими програмами учні отримують всебічний розвиток. Для навчання потрібні різні, більш технічно складні навички, зокрема із застосуванням математичних знань і наукових понять.

STEM (S – science, T – technology, E – engineering, M – mathematics). Акронім STEM використовується для позначення популярного напрямку в освіті, що охоплює природничі науки

(Science), технології (Technology), технічну творчість (Engineering) та математику (Mathematics). Це напрям в освіті, за якого в навчальних програмах посилюється природничо-науковий компонент та інноваційні технології. Технології використовують навіть у вивченні мистецьких дисциплін. У STEM-освіті активно розвивається креативний напрямок, до якого включені творчі та художні дисципліни (промисловий дизайн, архітектура та індустріальна естетика і т.д.). Розвиток освіти лише у науковому напрямку без паралельного розвитку мистецьких дисциплін (Arts) може призвести до того, що учні позбудуться навичок креативності. Тому сьогодні використовують розширену назву STEAM [3].

Для реалізації творчих проєктів з використання знань з галузі точних наук можна використовувати платформу Arduino. Arduino – це відкрита платформа, яка складається з двох основних частин: самої плати (часто званої мікроконтролер) і програмного забезпечення (спеціальної оболонки для програмування плати) або IDE (Integrated Development Environment). Програмне забезпечення функціонує на персональному комп'ютері, а через відповідні функції можна здійснювати запис написаного коду на плату. За рахунок спрощеної системи програмування, синтаксис якої базується на широко використовуваній мові C++, використовувати Arduino можуть не лише фахівці з галузей програмування, фізики, математики та робототехніки. Проте процес освоєння і створення проєктів буде набагато легший і ефективніший, якщо учні мають базові знання з схемотехніки і електротехніки. Розглянемо деякі з них:

Закон Ома – один з головних законів електрики $I = \frac{U}{R}$;

Потужність – міра швидкості зміни електричної енергії в іншу форму $P = I \cdot U$;

Знаючи закон Ома, потужність можна обчислити $P = I^2 \cdot R = \frac{U^2}{R}$.

В процесі роботи частина енергії трансформується в тепло. Через це гріються комп'ютери, телефони та інша техніка. $P = P_W + P_D$, де P – споживча потужність, P_W – корисна потужність, P_D – потужність, що переходить в нагрівання.

Коротке замикання. З'єднання плюса і мінуса напряму, за законом Ома, означає дуже велику силу струму i , як наслідок дуже велику потужність нагрівання, що в результаті спричиняє загорання. Такий процес називається коротким замиканням. Ніколи не потрібно допускати його появу, ні за яких обставин.

Послідовне з'єднання. В разі послідовного з'єднання елементів сила струму на кожному елементі однакова, а напруга падає в кожному наступному елементі.

Паралельне з'єднання. В разі паралельного з'єднання напруга навколо кожного елемента однакова. Сила струму залежить від опору кожного окремого елемента.

Широтно-імпульсна модуляція. Через мікроконтролери зазвичай неможливо видавати довільну напругу. Через них можна видати або напругу живлення (наприклад, 5В), або землю (тобто 0В). Але, використовуючи рівень напруги, можна управляти багатьма пристроями: наприклад, яскравість світлодіода або швидкість обертання мотора. Для подання неповної напруги використовується ШІМ (Широтно-Імпульсна Модуляція, англ. Pulse Width Modulation або просто PWM).

Вихід мікроконтролера перемикається між землею і V_{CC} тисячі разів за секунду, або, як ще кажуть, з частотою в тисячі герц. Наприклад оком не помічається мерехтіння більше 50Гц, тому здається, що світлодіод не мерехтить, а горить в півсили. Аналогічно, розігнаний мотор не може зупинити вал за мілісекунди, тому через ШІМ-сигнал можна зменшити частоту обертів.

Дільник напруги. Через послідовно під'єднані резистори можна розділити напругу, що надходить на них, у певній пропорції. Сила струму, що протікає через резистори, однакова, тому що вони з'єднані послідовно, і за законом Ома може бути розрахована як:

$$I = \frac{V_{CC}}{R_1 + R_2}$$

За тим же законом Ома можна обчислити напругу V_{out} , яка падає на резисторі R_2 :

$$V_{out} = U_2 = I \cdot R_2 = \frac{R_2 \cdot V_{CC}}{R_1 + R_2}$$

З отриманої формули видно, що чим більше R_2 щодо R_1 , тим більша напруга падає на ньому.

Зчитування резистивних сенсорів. Якщо замість R_2 використовувати не постійний резистор, а датчик зі змінним опором, V_{out} буде залежати від вимірюваного значення. Використовуючи мікроконтролер, можна вимірювати напругу. Таким чином, можна використовувати властивості дільника напруги для отримання показань від сенсора. На основі цього принципу побудований процес зчитування показників різних датчиків.

Arduino використовуються в якості головного пристрою управління для роботів, 3D принтерів, в системах автоматизованого поливу, світлодіодних кубах, грілках, в системах «розумних будинків» і

т.д. Схеми на основі Arduino застосовуються практично в будь-якому проекті, де необхідна автоматизація.

На сьогодні випускаються різні моделі плат на базі Arduino. Кожна з них призначена для різних завдань. Деякі плати принципово відрізняються одна від одної, але в більшості з них є такі однакові вузли.

Кожна плата Arduino повинна приєднуватися до джерела живлення. Наприклад джерелом живлення для Arduino Uno може бути USB кабель від персонального комп'ютера, або від окремого адаптера, який під'єднується до передбаченого на платі роз'єму. USB також використовується для завантаження програми (скетчу) на плату. Слід пам'ятати, що не можна використовувати джерело живлення з напругою на виході більше 20 вольт. Це може привести до того, що плата перегорить. Рекомендована напруга живлення для Arduino – від 6 до 12 вольт. Роз'єми-піни на платі Arduino – це передбачені роз'єми, до яких потрібно підключати провідники від периферійних пристроїв (дуже часто для прототипів використовують макетні плати і дроти із з'єднувачами на кінцях).

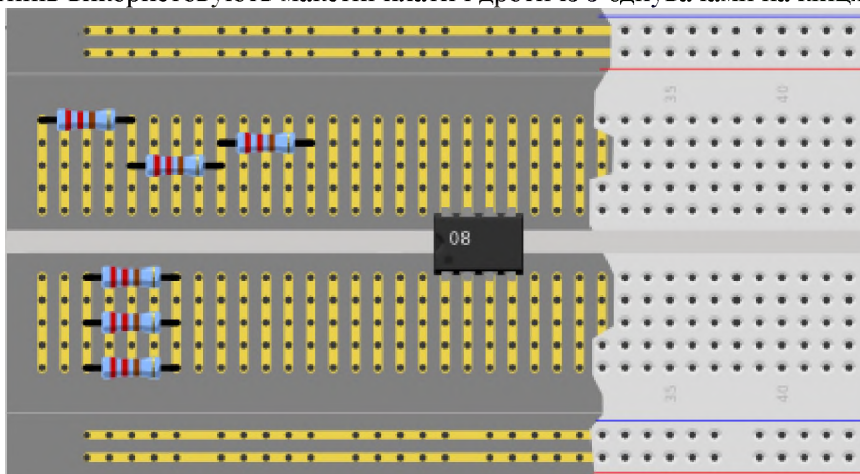


Рис. 5. Схема доріжок макетної плати

Розглянемо стандартну схему доріжок макетної плати. Довгі доріжки з боків зазвичай використовуються для під'єднання живлення та заземлення. Розрив посередині використовується для під'єднання компонентів з двома рядами ніг, наприклад тригери. Короткі доріжки можна використовувати як для послідовного так і для паралельного з'єднання.

На Arduino кілька типів пінів, кожен з яких підписаний відповідно до виконуваної функції.

GND: скорочення від «Ground» – «Земля». На платах кілька пінів GND, кожен з яких може використовуватися для заземлення електричного кола.

5V і 3.3V: – піни, через які на виході забезпечується живлення в 5 вольт і 3.3 вольт відповідно. Для більшості компонентів, які під'єднуються до Arduino, потрібне живлення 5 або 3.3 вольт.

Analog: на частині плати, яка підписана 'Analog In' (від A0 до A5 на Arduino Uno), розташовані аналогові входи. За допомогою цих пінів можна зчитувати сигнали від аналогових датчиків (наприклад, датчик температури) і перетворювати їх в цифрові значення.

Digital: навпроти аналогових пінів знаходяться цифрові піни (від 0 до 13 на Arduino Uno). Ці піни використовуються для цифрових вхідних (input) сигналів (наприклад, натискання кнопки) і для генерації цифрових вихідних (output) сигналів (наприклад, живлення світлодіода).

PWM: знак (~) поряд з деякими цифровими пінами (3, 5, 6, 9, 10, і 11 на UNO) означає, що ці піни можна використовувати як в звичайному цифровому режимі, так і в режимі ШІМ-модуляції.

Кнопка скидання (Reset Button) – кнопка скидання (reset). В разі натискання на неї контакт скидання замикається з землею і код, завантажений на Arduino, починає виконуватись заново.

Індикатор живлення (Power LED) – світлодіод, підписаний "on". Цей світлодіод повинен засвітитися під час підключення плати до джерела живлення. Світлодіоди TX і RX – скорочення від transmit (передавання), RX – від receive (приймання). Ці умовні позначення часто зустрічаються в електроніці для позначення контактів, які відповідають за серійний обмін даними. На Arduino ці контакти зустрічаються два рази на цифрових пінах 0 і 1 і в якості світлодіодів TX і RX. Завдяки сигналам цих світлодіодів можна візуально відслідковувати, що відбувається – передавання або приймання даних.

Розглянемо деякі основні компоненти, які використовуються для побудови проектів.

Конденсатор. Конденсатор (англ. capacitor) – система з двох чи більше електродів (обкладок), які розділені діелектриком, товщина якого менша у порівнянні з розміром обкладок. В такій системі є взаємна електрична ємність і тому в ній можна зберігати електричний заряд. Конденсатор є пасивним

електронним компонентом і широко застосовується в електронних схемах для блокування постійного струму з пропусканням змінного струму. Конденсатор – це маленький акумулятор, який швидко заряджається і швидко розряджається.

Резистор. Резистори належать до електричних компонентів, що застосовуються в схемах електротехніки та електроніки для обмеження сили струму та розподілу напруги. Резистори – найпоширеніші пасивні компоненти електронної апаратури, що використовуються як навантаження, поглиначі та дільники напруги в колах живлення, як елементи фільтрів, шунти, в колах формування імпульсів і т.д.

За послідовного з'єднання резисторів їх еквівалентною схемою буде резистор з опором, рівним сумі опорів окремих резисторів:

$$R = R_1 + R_2 + \dots + R_n = \sum_{i=1}^N R_i$$

За *паралельного* з'єднання резисторів обернена величина еквівалентного опору (провідність) дорівнює сумі обернених величин усіх опорів (провідностей).

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} + \dots + \frac{1}{R_N} = \sum_{i=1}^N \frac{1}{R_i}$$

Діод – електронний прилад з двома електродами, через який пропускається електричний струм лише в одному напрямі. Розрахуємо який опір резистора R потрібно взяти, щоб отримати оптимальний результат. Для прикладу, нехай характеристики світлодіода, який потрібно під'єднати такі: $V_F = 2.3\text{В}$, $I = 20\text{мА}$, напруга джерела живлення $V_{CC} = 5\text{В}$. Знайдемо оптимальний опір R і мінімально допустиму потужність резистора P_R . Спочатку обчислимо, яку напругу слід подати на резистор: $U_R = V_{CC} - V_F = 5\text{В} - 2.3\text{В} = 2.7\text{В}$. За законом Ома знайдемо значення опору, за якого буде забезпечено таке падіння: $R = \frac{U_R}{I} = \frac{2.7\text{В}}{0.02\text{А}} = 135\text{Ом}$. Таким чином в разі опору більше за 135Ом яскравість буде нижчою за заявлену у специфікації виробу, в разі опору меншому за 135Ом термін роботи світлодіода буде меншим. Тепер знайдемо потужність, яку за такого процесу, через резистор доведеться розсіювати: $P_R = I^2 \cdot R = 0.02\text{А}^2 \cdot 135\text{Ом} = 0.054\text{Вт}$. Це означає, що за потужності резистора, меншій за 54 мВт – резистор перегорить.

Тактова кнопка – простий механізм, за допомогою якого можна замкнути електричне коло поки кнопка натиснута.

Для програмування плат код записується у дві основні функції.

Функція `setup()` викликається на початку скетчу. Вона використовується для ініціалізації змінних, налаштування режимів роботи пінів (на введення або на виведення). Функція `setup()` виконується один раз після подання живлення або перезавантаження плати Arduino.

Після виконання функції `setup()`, циклічно виконується функція `loop()`, яка безпосередньо є основою програми для управління платою Arduino.

Процес ознайомлення і навчання роботи з платою можна почати і без використання реального пристрою, оскільки під час навчання велика ймовірність пошкодити плату через брак досвіду. Таким чином доцільніше використати спеціальні програмні сервіси для емулявання роботи пристроїв.

Емулятор – це повноцінний аналог, за допомогою якого можна замінити оригінал. Одним із таких сервісів є онлайн емулятором Tinkercad Circuits Arduino. Перевагою даного сервісу над іншими аналогічними програмними продуктами є те, що можна не тільки створювати електричні схеми, а й запускати емулятор електричного кола та підключати до проекту віртуальну плату Arduino, в яку можна завантажити працюючі скетчі. Tinkercad – це безкоштовний, простий для опанування сервіс, з використанням якого можна розпочинати навчання електротехніки та робототехніки.

Розглянемо основні послуги, що включені до даного сервісу:

- емулятор датчиків та інструментів зовнішнього впливу, за допомогою яких можна змінювати показники датчиків, стежачи за тим, які дані при цьому надходять до системи;
- вбудований редактор коду Arduino з монітором порту і можливістю покрокового налагодження;
- готові для розгортання проекти Arduino зі схемами і кодом;
- візуальний редактор коду Arduino;
- можливість інтеграції з рештою функцій Tinkercad і швидкого створення для проєктованого пристрою корпусу та інших конструктивних елементів – створена модель може бути відразу ж завантажена на 3D-принтер;
- вбудовані підручники і величезне співтовариство з колекцією готових проєктів.

Також в даному сервісі можна програмувати коди програм, використовуючи як класичну мову C++, так і створювати код з допомогою візуальних блоків типу Scratch (відповідний код мовою C++ буде згенерований автоматично).

Розглянемо простий проект для демонстрації основ роботи з сервісом.

Нехай потрібно створити можливість встановлювати яскравість світіння світлодіода за допомогою потенціометра, а також за допомогою кнопки організувати перемикання між трьома світлодіодами різних кольорів. Схему під'єднання подано на рис. 2.

Для реалізації потрібен такий набір компонентів:

Таблиця 1. Список компонентів

<i>Компонент</i>	<i>Кількість</i>
Arduino Uno R3	1
Резистор 150 Ом	3
Резистор 10 Ком	1
Червоний/синій/зелений світлодіод	1/1/1
Кнопка	1
Поляризований конденсатор 100 нФ	1
Тригер Шмідта	1
Потенціометр 250 КОм	1

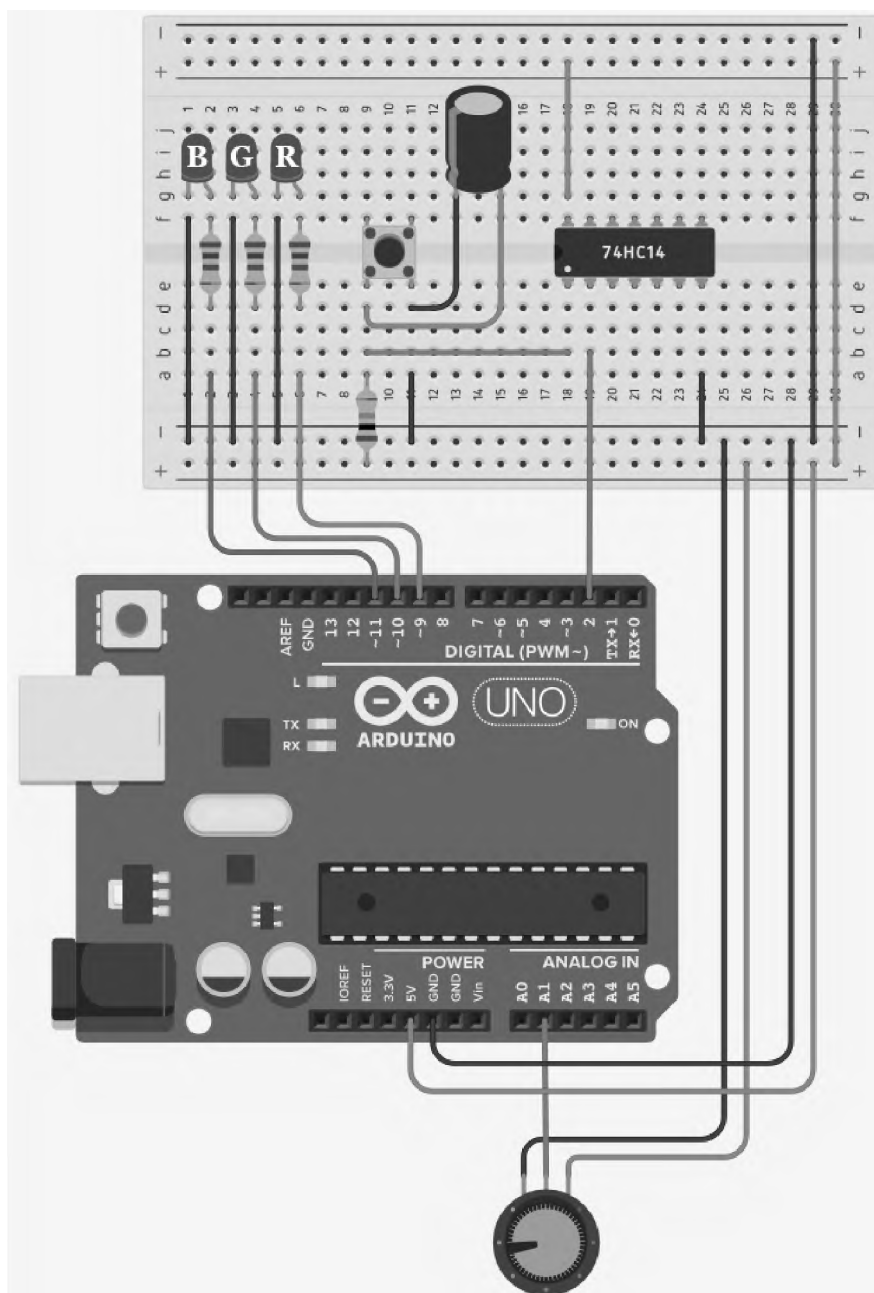


Рис. 6. Приклад проекту з використанням кнопки

Світлодіоди, як згадувалось раніше, повинні під'єднуватись з використанням резисторів. Потенціометр є звичайним дільником напруги, дані з якого, використовуючи вбудовану функцію `map()`, потрібно перетворити у діапазон вихідного сигналу (від 0 до 255). Оскільки використання тактової кнопки пов'язане з деякими фізичними процесами, де потрібна стабілізація сигналу, то підключення потрібно здійснювати з використанням конденсатора та інвертуючого тригера Шмітта.

Код програми, який потрібно завантажити на плату.

```
int buttonInt = 0;
int redLED = 9;
int greenLED = 10;
int blueLED = 11;
int nullLED = 6;
volatile int selectedLED = redLED;
int distPin = 1;
void setup()
{
  pinMode(redLED, OUTPUT);
  pinMode(greenLED, OUTPUT);
  pinMode(blueLED, OUTPUT);
  pinMode(nullLED, OUTPUT);

  attachInterrupt(buttonInt, swap, RISING);
  Serial.begin(9600);
}
void swap()
{
  if (selectedLED == redLED)
    selectedLED = greenLED;
  else if (selectedLED == greenLED)
    selectedLED = blueLED;
  else if (selectedLED == blueLED)
    selectedLED = nullLED;
  else
    selectedLED = redLED;
  Serial.println(selectedLED);
}
void loop()
{
  int dist = analogRead(distPin);
  int brightness = map(dist, 0, 1023, 0, 255);
  analogWrite(selectedLED, brightness);
}
```

З проектом можна ознайомитись за посиланням <https://www.tinkercad.com/things/ITKKkTfXvNc-buttonrgbpottrigger>.

Розглянемо ще одну задачу з підключенням мотора з живленням 9 В та LCD-дисплеєм. Швидкість обертання відображається на дисплеї у вигляді напису «RPM is 000» (Rotes per minute – обертів за хвилину), також швидкість можна змінювати за допомогою потенціометра.

Для реалізації потрібен такий набір компонентів:

Таблиця 2 Список компонентів

<i>Компонент</i>	<i>Кількість</i>
Arduino Uno R3	1
Резистор 1 Ком	3
Елемент живлення 9 В	1
Мотор	1
Біполярний NPN-транзистор	1
Конденсатор 100 нФ	1
Діод	1
ЖК дисплей 16x2	1
Потенціометр 250 Ком	1

Схему під'єднання подано на рис. 3.

Код програми, який потрібно завантажити на плату.

/*

The circuit:

```

* LCD RS pin to digital pin 8
* LCD Enable pin to digital pin 9
* LCD D4 pin to digital pin 3
* LCD D5 pin to digital pin 4
* LCD D6 pin to digital pin 5
* LCD D7 pin to digital pin 6
* LCD R/W pin to ground
* LCD VSS pin to ground
* LCD VCC pin to 5V
// підключення бібліотеки для роботи з дисплеєм
#include <LiquidCrystal.h>
// ініціалізація дисплею відповідно до під'єднаних пінів
LiquidCrystal lcd(8, 9, 3, 4, 5, 6);
int motorPin = 10;
int potPin = A5;
void setup() {
    // встановлення відповідної до обраного дисплею символічної роздільної
    // здатності, 2 рядки по 16 символів:
    lcd.begin(16, 2);
    lcd.print("RPM is");
    pinMode(motorPin, OUTPUT);
    pinMode(potPin, INPUT);
}
void loop() {
    int rpm = analogRead(potPin);
    rpm = map(rpm, 0, 1023, 0, 255);
    analogWrite(motorPin, rpm);
    rpm = map(rpm, 0, 255, 0, 285);
    lcd.setCursor(0, 1);
    lcd.print(rpm);
    delay(200);
}

```

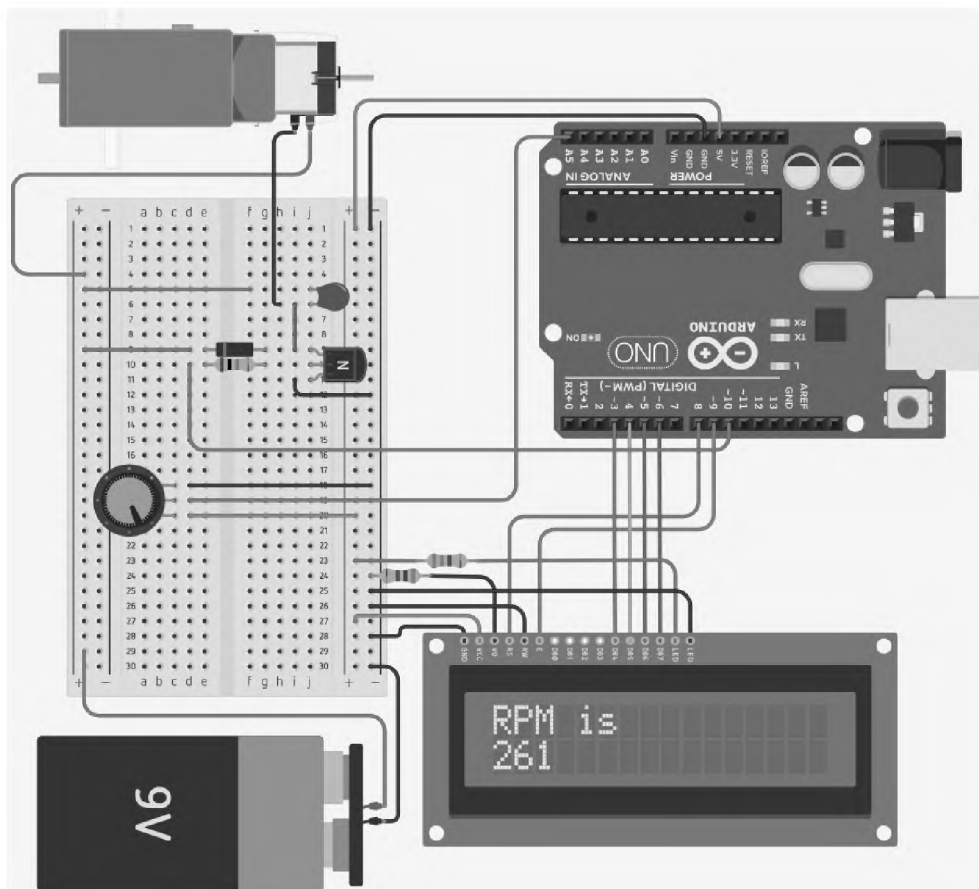


Рис. 7 Приклад проекту з використанням мотора

З проектом можна ознайомитись за посиланням <https://www.tinkercad.com/things/6ifC6VcTbxY-9v-motor-control-with-lcd>.

Використовуючи веб сервіси, вивчення елементів схемотехніки можна поєднувати з вивчення дисциплін природничого циклу разом зі створенням проектів в галузі інформаційних технологій з використанням теоретичних знань відповідних дисциплін. Проте використання веб сервісів для емуляції роботи пристроїв не може повною мірою замінити аналіз роботи цих пристроїв. Тому розпочинати навчання доцільно, використовуючи такі сервіси, а для подальшої роботи бажано використовувати реальні об'єкти. Незважаючи на якість створених емуляторів, вони ніколи не замінять реальний процес роботи, а якісне навчання можливе лише за гармонійного поєднання теорії і практики.

Список використаних джерел:

1. Tinkercad blog: веб-сайт. URL: <https://blog.tinkercad.com> (дата звернення 07.05.2019).
2. Arduino official site: веб-сайт. URL: <https://www.arduino.cc> (дата звернення 07.05.2019).
3. Інститут модернізації змісту освіти: веб-сайт. URL: <https://imzo.gov.ua/stem-osvita> (дата звернення 07.05.2019).
4. Blum J. Exploring Arduino®: Tools and Techniques for Engineering Wizardry / Jeremy Blum. Indianapolis, Indiana: John Wiley & Sons Inc, 2013. 385 с.

References:

1. Tinkercad blog: website. URL: <https://blog.tinkercad.com> (review date 07.05.2019).
2. Arduino official site: website. URL: <https://www.arduino.cc> (review date 07.05.2019).
3. Instytut modernizaciyi zmistu osvity: website. URL: <https://imzo.gov.ua/stem-osvita> (review date 07.05.2019).
4. Blum J. Exploring Arduino®: Tools and Techniques for Engineering Wizardry / Jeremy Blum. Indianapolis, Indiana: John Wiley & Sons Inc, 2013. 385 p.

Computer-oriented systems in STEAM-education

Bilii Yu.P.

Abstract. Education is the foundation for the successful development of the country. After the digital revolution, when most of the professions are impossible without the use of computers, information technology, the Internet. In the process of learning it is necessary to harmoniously add the study of modern technologies without losing the fundamental principles of other disciplines. Such a combination is possible with the integration in the fundamental disciplines of programming elements and information technology. The article contains of some theoretical information is considered for those who are beginning to study circuitry and also show examples of using the Tinkercad web service to create Arduino-based projects without the availability of the board.

Keywords: Arduino, programming, circuitry.

DOI 10.31392/NPU-nc.series 2.2019.21(28).12

УДК 378.147:004.65

В.В. Єфименко

кандидат педагогічних наук, доцент

Національний педагогічний університет імені М.П. Драгоманова

ОСОБЛИВОСТІ КУРСУ "ПРОЕКТУВАННЯ ТА ОПРАЦЮВАННЯ БАЗ ДАНИХ" ДЛЯ МАЙБУТНІХ ВЧИТЕЛІВ ІНФОРМАТИКИ

Анотація. У статті розглянуто деякі аспекти методики навчання баз даних в педагогічному університеті як необхідну складову професійної підготовки фахівців в галузі інформатики. Досліджено групу питань теоретичного змісту та запропоновано методику проведення лабораторних робіт з курсу «Проектування та опрацювання баз даних». Особливу увагу приділено вивченню питань, пов'язаних з мережевими моделями баз даних. Розглянуто приклади таких бази даних.

Ключові слова: база даних, мережева модель баз даних, система управління базами даних, технологія опрацювання даних.

Сьогодні поєднання традиційних технологій навчання з інформаційними для майбутнього фахівця дає ефективний результат формування професійно-педагогічних якостей. Майбутній вчитель інформатики зобов'язаний вміти розробляти різні дисципліни інформатичного циклу та брати активну участь з підтримки інформаційного забезпечення закладу освіти (наприклад, створення сайту навчального закладу, впровадження дистанційного навчання, обліку і т.п.). Впровадженню сучасних