

УДК 004.9+537+62 (076)

*Рекомендовано до друку рішенням Вченої ради
Черкаського державного бізнес-коледжу
Протокол № від ____ 2024 р.*

Укладач: Орел А.С.

Основи робототехніки
Практикум
Черкаси, 2024 р. – 43 с.

Рецензент: Пасічний М.О. - кандидат фізико-математичних наук, доцент, завідувач кафедри фізики Черкаського національного університету ім. Б. Хмельницького.

Практикум містить 12 практичних робіт, кожна з яких присвячена окремій темі і роботі з певним типом цифрового чи аналогового сенсору.

До кожної практичної роботи є перелік необхідних матеріалів та детальна інструкція з її виконання. В кінці практичної роботи є завдання для самостійного виконання.

Розробка призначена для студентів закладів фахової передвищої освіти, учнів загальноосвітніх шкіл, ліцеїв, гімназій.

Затверджено на засіданні циклової комісії
природничо-математичних та гуманітарних дисциплін
Протокол № 9 від 26.04.2024 року

© А.С. Орел

ПРО УКЛАДАЧА

Орел Андрій Сергійович – викладач Черкаського Державного бізнес-коледжу з 2020 року, спеціаліст другої категорії, закінчив магістратуру Черкаського Національного Університету імені Б. Хмельницького у 2020 р., спеціальність «Середня освіта. Фізика», вчитель фізики, астрономії та інформатики, викладач фізики. Є автором навчально-методичної розробки «ФІЗИКА. Збірник тестових завдань. Частина 1»(2022 рік)

Список рекомендованих джерел

1. Клейборн А. Я можу бути робототехніком. Посібник. Київ : Vivat, 2021. 64 с.
2. Мульганов К., Пустільник С. Робототехніка. STEM-освіта. Київ : Шкільний світ, 2020. 104 с.
3. Програма технічного конструювання. Програми з позашкільної освіти науково-технічний напрям (інформаційно-технічний профіль). Київ С.15-32
4. Морзе Н.В., Варченко Л.О., Троценко М.А. Основи робототехніки: навчальний посібник. Кам'янець-Подільський : ПП Буйницький О.А., 2016. 184 с.
5. Робототехнічні системи: проектування і моделювання [Електронний ресурс]: навч. Посіб. для студ. спеціальності 126 «Інформаційні системи та технології» / М. М. Поліщук, М.М. Ткач; КПІ ім. Ігоря Сікорського. Електронні текстові дані (1 файл: 41,6 Мбайт). Київ: КПІ ім. Ігоря Сікорського, 2021. 112 с.
6. Офіційна сторінка Інституту LEGO Education (Данія) URL:<http://www.legoeducation.com>
7. Офіційна сторінка виробника LEGO (Данія) URL: <https://education.lego.com/en-us>
8. Офіційна сторінка авторів програми ТОВ «ПРОЛЕГО» (Україна, Київ). Сайт методичної підтримки вчителів курсу «Основи робототехніки» URL: <http://www.prolego.com.ua>
9. Офіційна сторінка виробника програмного забезпечення National Instruments (США) URL:<http://www.ni.com/>
10. Сайт компанії «Інноваційні Освітні Рішення» URL:<http://ies.org.ua/>

ЗМІСТ

Вступ	4
Тема 1. Цифрові порти платформи Arduino. Вхідні сигнали	5
Тема 2. Цифрові порти платформи Arduino. Вхідні сигнали	8
Тема 3. Аналогові порти платформи Arduino	11
Тема 4. Відображення даних. Дисплеї	13
Тема 5. Вхідні дані. Матриця кнопок	16
Тема 6. Широтно-імпульсна модуляція	20
Тема 7. Цифрові сенсори. Дальномір	23
Тема 8. Цифрові сенсори. Термометр	26
Тема 9. Керування двигунами. Серводвигун	30
Тема 10. Керування двигунами. Комутація	33
Тема 11. Керування кольорами. Кольоровий простір	36
Тема 12. Керування часом. Таймери	39
Список використаних джерел	42

ВСТУП

З ростом інтересу до STEM-освіти (Science, Technology, Engineering, Mathematics) робототехніка стала не лише актуальною, але й захоплюючою галуззю, що визначає майбутнє. Навички програмування, електроніки та механіки набувають нового виміру, завдяки можливостям створення роботів, які здатні спрощувати наше життя, вирішувати важливі завдання та розвивати сучасні технології.

У цьому посібнику студенти зануритися у світ робототехніки, починаючи від основ та прогресуючи до складніших проєктів. Ми розглянемо основні концепції, методи програмування та роботи з цифровими та аналоговими пристроями. Навчимося використовувати датчики та сенсори. Розв'яжемо проблеми які виникають під час створення робототехнічних систем.

Завдяки цьому посібнику студенти отримають практичні навички, які допоможуть втілити в життя власні ідеї та проєкти.

- Зробити секундомір, для цього потрібно в схему додати кнопку *рисунок 12.3* для зупинки відліку *рисунок 12.4*.

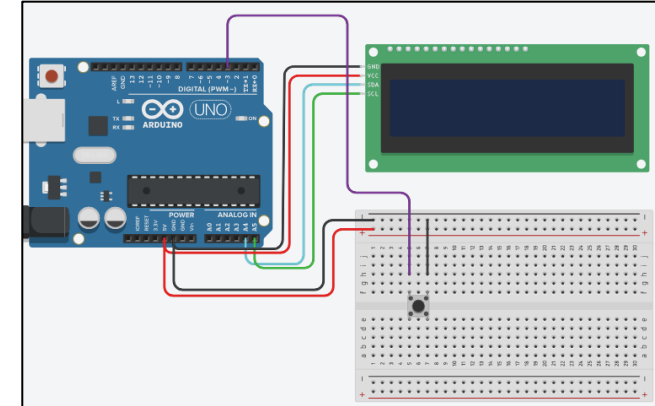


Рисунок 12.4 «Схема підключення дисплея та кнопки»

Джерело: розробка автора

- Додати в схему ще одну кнопку для продовження відліку часу після зупинки.
- Згідно проведеної роботи підготувати письмовий звіт.

```

1 #include <LiquidCrystal_I2C.h>
2 LiquidCrystal_I2C lcd(39,16,2);
3 bool flag = true;
4 void setup()
5 {
6     lcd.init();
7     lcd.backlight();
8     pinMode(3, INPUT_PULLUP);
9     attachInterrupt(1, btnStop, FALLING);
10 }
11 void btnStop() {
12     flag = false;
13 }
14 void loop()
15 {
16     if (flag){
17         lcd.clear();
18         lcd.setCursor(0,0);
19         lcd.print(millis()/1000);
20         lcd.print(":");
21         lcd.print((millis()%1000)/10);
22         delay(10);
23     }

```

Рисунок 12.4 «Програмний код»

Джерело: розробка автора

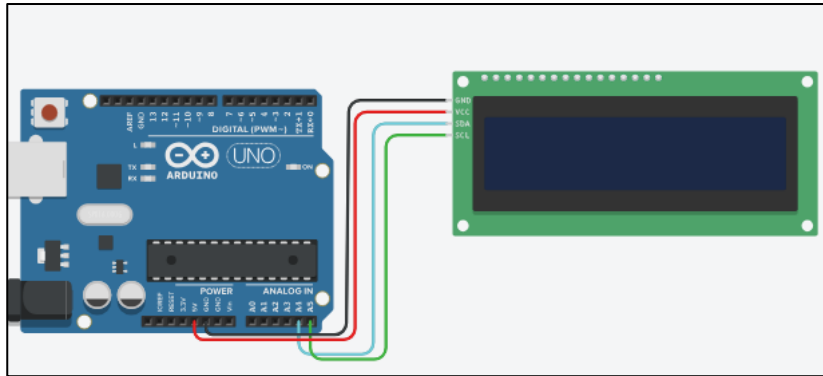


Рисунок 12.1 «Схема підключення дисплея»

Джерело: розробка автора

Завдання до виконання

1. Запрограмувати схему згідно *рисунку 12.2* таким чином, щоб на дисплей виводився час з моменту запуску контролера.

```

1  #include <LiquidCrystal_I2C.h>
2  LiquidCrystal_I2C lcd(39,16,2);
3  void setup()
4  {
5      lcd.init();
6      lcd.backlight();
7  }
8  void loop()
9  {
10     lcd.clear();
11     lcd.setCursor(0,0);
12     lcd.print(millis()/1000);
13     lcd.print(":");
14     lcd.print((millis()%1000)/10);
15     delay(10);
16 }
    
```

Рисунок 12.2 «Програмний код»

Джерело: розробка автора

ТЕМА 1 ЦИФРОВІ ПОРТИ ПЛАТФОРМИ ARDUINO. ВИХІДНІ СИГНАЛИ

Практична робота №1. Світлодіод

Мета роботи:

- ознайомитись з основами роботи на платформі Arduino;
- повторити принцип роботи світлодіода та резистора;
- навчитись керувати цифровими портами платформи.

Матеріально-технічне забезпечення:

- комп'ютер;
- платформа Arduino;
- світлодіоди;
- резистори;
- макетна плата;
- провідники.

Порядок виконання роботи

1. Повторити теоретичний матеріал за темою [3].
2. Повторити схему в подану на *рисунку 1.1* в середовищі Tinkercad.
3. Перевірити чи не має в схемі критичних помилок, які можуть призвести до згорання елементів схеми.
4. Повторити схему в подану на *рисунку 1.1* за допомогою набору робототехніки.
5. Запрограмувати мікроконтролер в середовищі Arduino IDE відповідно до завдання.
6. Сформулювати звіт.

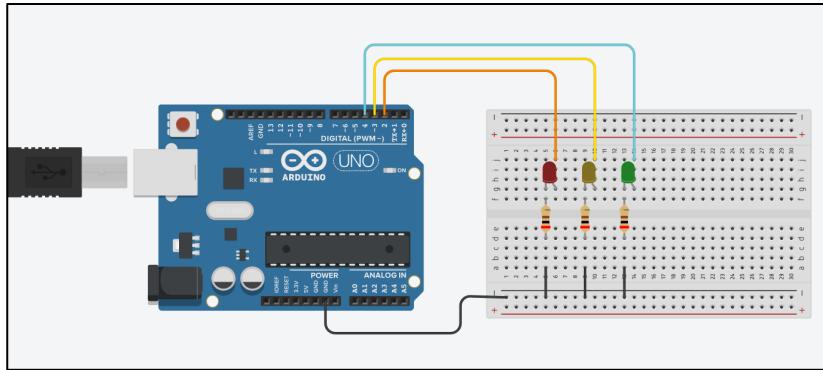


Рисунок 1.1 «Схема підключення тактової кнопки»

Джерело: розробка автора

Завдання до виконання

1. Запрограмувати схему таким чином щоб по черзі на одну секунду вмикався кожен світлодіод згідно *рисунку 1.2*.

```

1 void setup()
2 {
3   pinMode(2, OUTPUT);
4   pinMode(3, OUTPUT);
5   pinMode(4, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(2, HIGH);
11  digitalWrite(3, LOW);
12  digitalWrite(4, LOW);
13  delay(1000);
14
15  digitalWrite(2, LOW);
16  digitalWrite(3, HIGH);
17  digitalWrite(4, LOW);
18  delay(1000);
19
20  digitalWrite(2, LOW);
21  digitalWrite(3, LOW);
22  digitalWrite(4, HIGH);
23  delay(1000);
24 }
    
```

Рисунок 1.2 «Програмний код»

Джерело: розробка автора

ТЕМА 12 КЕРУВАННЯ ЧАСОМ. ТАЙМЕРИ

Практична робота №12. Таймер

Мета роботи:

- ознайомитись з основами роботи на платформі Arduino;
- повторити принцип роботи світлодіодів;
- навчитись керувати потрібним світлодіодом;
- навчитись задавати певні кольори за допомогою кольорової схеми.

Матеріально-технічне забезпечення:

- комп'ютер;
- платформа Arduino;
- світлодіод;
- макетна плата;
- провідники.

Порядок виконання роботи

1. Повторити теоретичний матеріал за темою [4,5,6].
2. Повторити схему в подану на *рисунку 12.1* в середовищі Tinkercad.
3. Перевірити чи не має в схемі критичних помилок, які можуть призвести до згорання елементів схеми.
4. Повторити схему в подану на *рисунку 12.1* за допомогою набору робототехніки.
5. Запрограмувати мікроконтролер в середовищі Arduino IDE відповідно до завдання.
6. Сформулювати звіт.

- Використовуючи сторонню бібліотеку відобразити по черзі всі доступні кольори.

```

1  #include <FastLED.h>
2  #define REDPIN    3
3  #define GREENPIN  5
4  #define BLUEPIN   6
5
6  void showAnalogRGB( const CRGB& rgb)
7  {
8      analogWrite(REDPIN,  rgb.r );
9      analogWrite(GREENPIN, rgb.g );
10     analogWrite(BLUEPIN,  rgb.b );
11 }
12
13 void colorBars()
14 {
15     showAnalogRGB( CRGB::Red );   delay(500);
16     showAnalogRGB( CRGB::Green ); delay(500);
17     showAnalogRGB( CRGB::Blue );  delay(500);
18     showAnalogRGB( CRGB::Black ); delay(500);
19 }
20
21 void setup() {
22     pinMode(REDPIN,    OUTPUT);
23     pinMode(GREENPIN,  OUTPUT);
24     pinMode(BLUEPIN,   OUTPUT);
25     colorBars();
26 }
27
28 void loop()
29 {
30     static uint8_t hue;
31     hue = hue + 1;
32     showAnalogRGB( CHSV( hue, 255, 255) );
33     delay(20);
34 }

```

Рисунок 11.3 «Програмний код»

Джерело: розробка автора

- Додати в схему тактову кнопку, запрограмувати таким чином, щоб при натисканні на неї, змінювався колір світлодіоду на випадковий у всій його палітрі.
- Згідно проведеної роботи підготувати письмовий звіт.

- Запрограмувати схему таким чином, щоб після вмикання всі світлодіоди горіли 3 секунди, а потім по черзі по 1 секундні кожен *рисуюнок 1.3*.

```

1  void setup()
2  {
3      pinMode(2,  OUTPUT);
4      pinMode(3,  OUTPUT);
5      pinMode(4,  OUTPUT);
6
7      for(int i = 2;i<5;i++){
8          digitalWrite(i, HIGH);
9      }
10     delay(1000);
11     for(int i = 2;i<5;i++){
12         digitalWrite(i, LOW);
13     }
14 }
15
16 void loop()
17 {
18     for(int i = 2;i<5;i++){
19         digitalWrite(i, HIGH);
20         delay(1000);
21         digitalWrite(i, LOW);
22     }
23 }

```

Рисунок 1.3 «Програмний код»

Джерело: розробка автора

- Запрограмувати схему, таким чином, щоб зелений світлодіод вмикався на пів секунди потім вимикався і через 1/10 секунди знову вмикався на пів секунди, а потім вже вмикались інші світлодіоди по одній секундні.

ТЕМА 2 ЦИФРОВІ ПОРТИ ПЛАТФОРМИ ARDUINO. ВХІДНІ СИГНАЛИ

Практична робота №2. Тактова кнопка

Мета роботи:

- ознайомитись з основами роботи на платформі Arduino;
- повторити принцип роботи тактової кнопки;
- навчитись керувати цифровими портами платформи.

Матеріально-технічне забезпечення:

- комп'ютер;
- платформа Arduino;
- тактова кнопка;
- світлодіод;
- макетна плата;
- провідники.

Порядок виконання роботи

1. Повторити теоретичний матеріал за темою [3,4].
2. Повторити схему в подану на *рисунку 2.1* в середовищі Tinkercad.
3. Перевірити чи не має в схемі критичних помилок, які можуть призвести до виходу з ладу елементів схеми.
4. Повторити схему в подану на *рисунку 2.1* за допомогою набору робототехніки.
5. Запрограмувати мікроконтролер в середовищі Arduino IDE відповідно до завдання.
6. Сформулювати звіт.

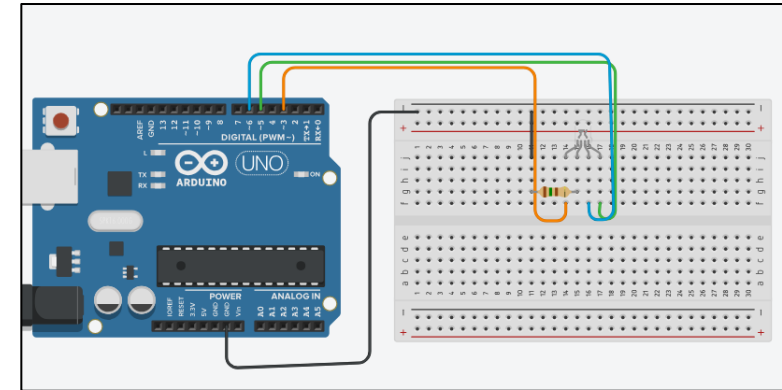


Рисунок 11.1 «Схема світлодіода»

Джерело: розробка автора

Завдання до виконання

1. Запрограмувати схему згідно *рисунку 11.2* таким чином, світлодіод по черзі плавно вмикався основними кольорами червоним, зеленим та блакитним.

```

1  #define REDPIN 3
2  #define GREENPIN 5
3  #define BLUEPIN 6
4
5  void render(int color){
6      for (int i = 0; i<=255;i++){
7          analogWrite(color, i );
8          delay(20);
9      }
10     analogWrite(color, 0 );
11 }
12
13 void setup() {
14     pinMode(REDPIN, OUTPUT);
15     pinMode(GREENPIN, OUTPUT);
16     pinMode(BLUEPIN, OUTPUT);
17 }
18
19 void loop()
20 {   int colorPin[] = {REDPIN, GREENPIN, BLUEPIN};
21     render(colorPin[0]);
22     render(colorPin[1]);
23     render(colorPin[2]);
24 }
    
```

Рисунок 11.2 «Програмний код»

Джерело: розробка автора

ТЕМА 11 КЕРУВАННЯ КОЛЬОРАМИ. КОЛЬОРОВИЙ ПРОСТІР

Практична робота №11. RGB світлодіод

Мета роботи:

- ознайомитись з основами роботи на платформі Arduino;
- повторити принцип роботи світлодіодів;
- навчитись керувати потрібним світлодіодом;
- навчитись задавати певні кольори за допомогою кольорової схеми.

Матеріально-технічне забезпечення:

- комп'ютер;
- платформа Arduino;
- світлодіод;
- макетна плата;
- провідники.

Порядок виконання роботи

1. Повторити теоретичний матеріал за темою [5,6].
2. Повторити схему в подану на *рисунку 11.1* в середовищі Tinkercad.
3. Перевірити чи не має в схемі критичних помилок, які можуть призвести до згорання елементів схеми.
4. Повторити схему в подану на *рисунку 11.1* за допомогою набору робототехніки.
5. Запрограмувати мікроконтролер в середовищі Arduino IDE відповідно до завдання.
6. Сформулювати звіт.

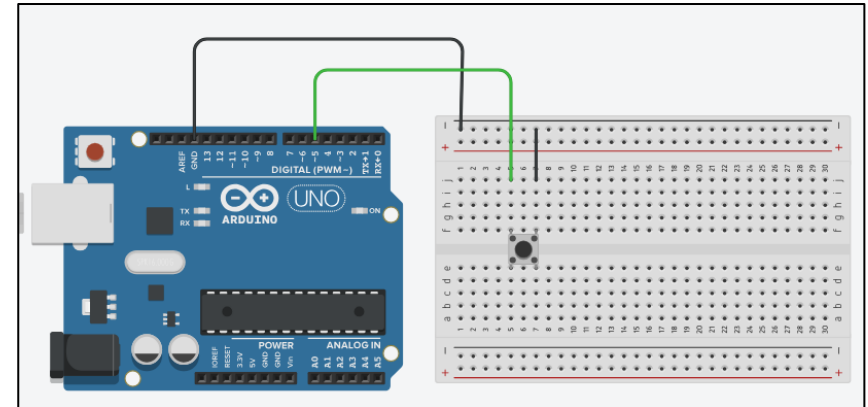


Рисунок 2.1 «Схема підключення тактової кнопки»

Джерело: розробка автора

Завдання до виконання

1. Запрограмувати схему згідно *рисунку 2.2* таким чином, щоб в полі монітора послідовного порту при нажатій кнопці показувалося число 1, а при відпущеній 0.

```
void setup()
{
    pinMode(5, INPUT_PULLUP);
    Serial.begin(9600);
}

void loop()
{
    Serial.println(!digitalRead(5));
}
```

Рисунок 2.2 «Програмний код»

Джерело: розробка автора

2. Запрограмувати схему згідно *рисунку 2.3* таким чином, щоб кнопка працювала як перемикач. Один клік по кнопці задавалось значення 1, другий клік – значення 0.

```
const int buttonPin = 5;
bool flag = false;
bool num = 1;

void setup() {
  Serial.begin(9600);
  pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
  bool btnState = digitalRead(buttonPin);
  if (btnState && !flag) {
    flag = true;
    num = !num;
    Serial.println(num);
  }
  if (!btnState && flag) {
    flag = false;
  }
  delay(10);
}
```

Рисунок 2.1 «Програмний код»

Джерело: розробка автора

3. Додати в схему світлодіод і керувати ним кнопкою як перемикачем. Один клік по кнопці вмикав світлодіод, а другий вимикав.
4. Згідно проведеної роботи підготувати письмовий звіт.

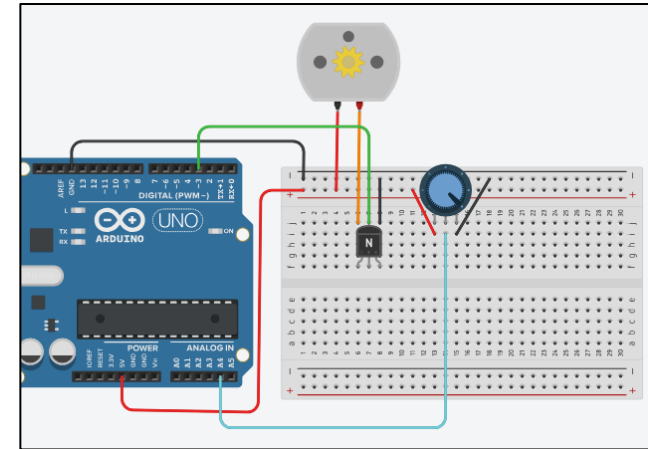


Рисунок 10.3 «Схема підключення потенціометру»
Джерело: розробка автора

```
#define Pot A4
#define Mot 3

void setup()
{
  pinMode(Mot, OUTPUT);
  pinMode(Pot, INPUT);
}

void loop()
{
  int value = analogRead(Pot);
  analogWrite(Mot, map(value, 0, 1024, 0, 256));
}
```

Рисунок 10.4 «Програмний код»

Джерело: розробка автора

3. Доповнити схему однією кнопкою, яка б вмикала моторчик та вимикала його..
4. Згідно проведеної роботи підготувати письмовий звіт

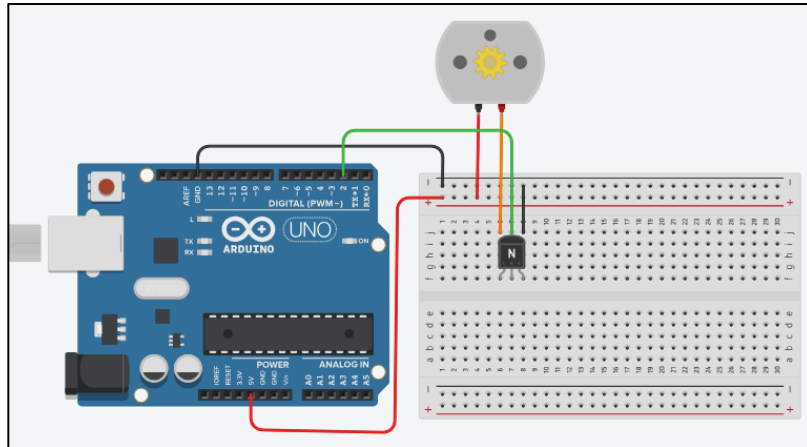


Рисунок 10.1 «Схема підключення транзистора»

Джерело: розробка автора

Завдання до виконання

1. Запрограмувати схему згідно *рисунку 10.2* таким чином, щоб запустити моторчик.

```
#define Mot 3

void setup()
{
  pinMode(Mot, OUTPUT);
}

void loop()
{
  digitalWrite(Mot, HIGH);
}
```

Рисунок 10.2 «Програмний код»

Джерело: розробка автора

2. Додати в схему потенціометр *рисунку 10.3*, та керувати обертами моторчика з його допомогою *рисунку 10.4*.

ТЕМА 3 АНАЛОГОВІ ПОРТИ ПЛАТФОРМИ ARDUINO

Практична робота №3. Потенціометр

Мета роботи:

- ознайомитись з основами роботи на платформі Arduino;
- повторити принцип роботи потенціометру;
- навчитись зчитувати значення з аналогових входів платформи.

Матеріально-технічне забезпечення:

- комп'ютер;
- платформа Arduino;
- потенціометр;
- світлодіод;
- макетна плата;
- провідники.

Порядок виконання роботи

1. Повторити теоретичний матеріал за темою [3,6].
2. Повторити схему в подану на *рисунку 3.1* в середовищі Tinkercad.
3. Перевірити чи не має в схемі критичних помилок, які можуть призвести до згорання елементів схеми.
4. Повторити схему в подану на *рисунку 3.1* за допомогою набору робототехніки.
5. Запрограмувати мікроконтролер в середовищі Arduino IDE відповідно до завдання.
6. Сформулювати звіт.

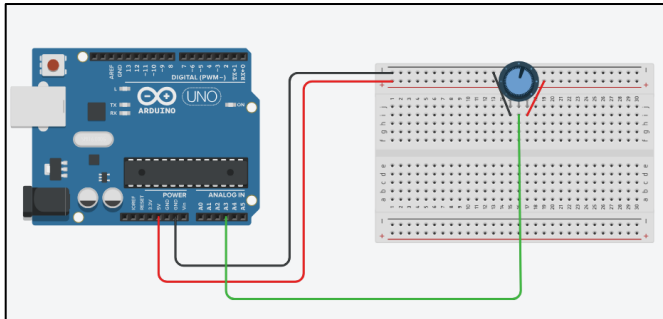


Рисунок 3.1 «Схема підключення потенціометра»

Джерело: розробка автора

Завдання до виконання

1. Запрограмувати схему згідно *рисунку 3.2* таким чином, щоб в полі монітора послідовного порту виводилось значення від 0 до 1023, взятого з потенціометру.

```
void setup()
{
  pinMode(A3, INPUT);
  Serial.begin(9600);
}

void loop()
{
  Serial.println(analogRead(A3));
  delay(50);
}
```

Рисунок 3.2 «Програмний код»

Джерело: розробка автора

2. За допомогою функції `map` зробити так щоб весь діапазон потенціометру був від 0 до 255.
3. Додати в схему світлодіод і керувати ним за допомогою потенціометру, тобто вмикати його коли значення потенціометра більше половини.
4. Згідно проведеної роботи підготувати письмовий звіт

ТЕМА 10 КЕРУВАННЯ ДВИГУНАМИ. КОМУТАЦІЯ

Практична робота №10. Транзисторне керування

Мета роботи:

- ознайомитись з основами роботи на платформі Arduino;
- повторити принцип роботи транзистору;
- навчитись керувати мотором;
- навчитись зчитувати значення з аналогових входів платформи.

Матеріально-технічне забезпечення:

- комп'ютер;
- платформа Arduino;
- потенціометр;
- мотор;
- транзистор;
- макетна плата;
- дисплей;
- провідники.

Порядок виконання роботи

1. Повторити теоретичний матеріал за темою [3,4].
2. Повторити схему в подану на *рисунку 10.1* в середовищі Tinkercad.
3. Перевірити чи не має в схемі критичних помилок, які можуть призвести до згорання елементів схеми.
4. Повторити схему в подану на *рисунку 10.1* за допомогою набору робототехніки.
5. Запрограмувати мікроконтролер в середовищі Arduino IDE відповідно до завдання.
6. Сформулювати звіт.

- Додати в схему потенціометр, який повинен керувати кутом повороту серводвигуну.

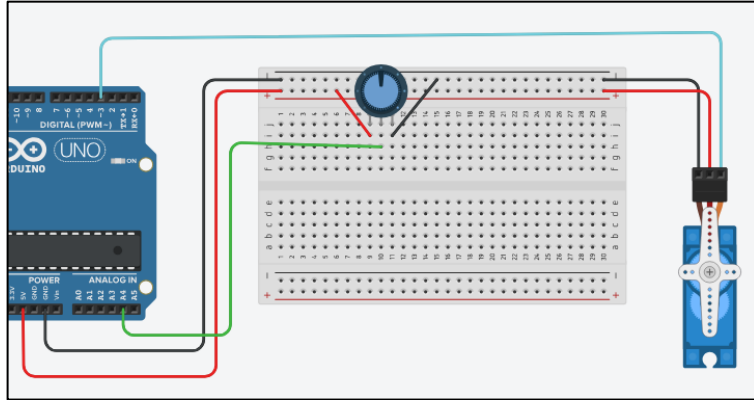


Рисунок 9.3 «Схема підключення серводвигуна і потенціометра»

- Дописати код так щоб на монітор послідовного порту виводилось значення кута на який повернутий серводвигун.
- Згідно проведеної роботи підготувати письмовий звіт

ТЕМА 4 ВІДОБРАЖЕННЯ ДАНИХ. ДИСПЛЕЇ

Практична робота №4. Дисплей 1602

Мета роботи:

- ознайомитись з основами роботи на платформі Arduino;
- повторити принцип роботи тактової кнопки;
- навчитись керувати дисплеєм;
- навчитись зчитувати значення з аналогових входів платформи.

Матеріально-технічне забезпечення:

- комп'ютер;
- платформа Arduino;
- потенціометр;
- світлодіод;
- макетна плата;
- дисплей;
- провідники.

Порядок виконання роботи

- Повторити теоретичний матеріал за темою [3,5].
- Повторити схему в подану на *рисунку 4.1* в середовищі Tinkercad.
- Перевірити чи не має в схемі критичних помилок, які можуть призвести до згорання елементів схеми.
- Повторити схему в подану на *рисунку 4.1* за допомогою набору робототехніки.
- Запрограмувати мікроконтролер в середовищі Arduino IDE відповідно до завдання.
- Сформулювати звіт.

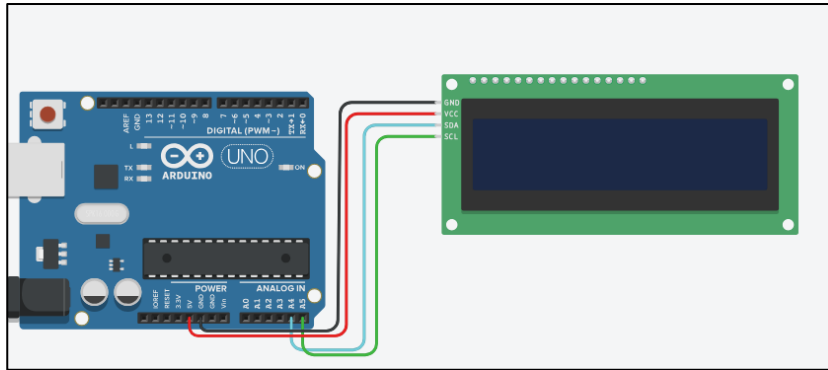


Рисунок 4.1 «Схема підключення дисплея»

Джерело: розробка автора

Завдання до виконання

1. Запрограмувати схему згідно *рисунку 4.2* таким чином, щоб на дисплей в першій рядок виводилось ваше прізвище та ім'я, а в другий рядок номер групи.

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(39,16,2);

void setup()
{
  lcd.init();
  lcd.backlight();
}

void loop()
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Orel Andriy ");
  lcd.setCursor(0,1);
  lcd.print("1P-23");
  delay(50);
}
```

Рисунок 4.2 «Програмний код»

Джерело: розробка автора

2. Додати в схему потенціометр, та виводити на дисплей взятє з нього значення від 0 до 1023.

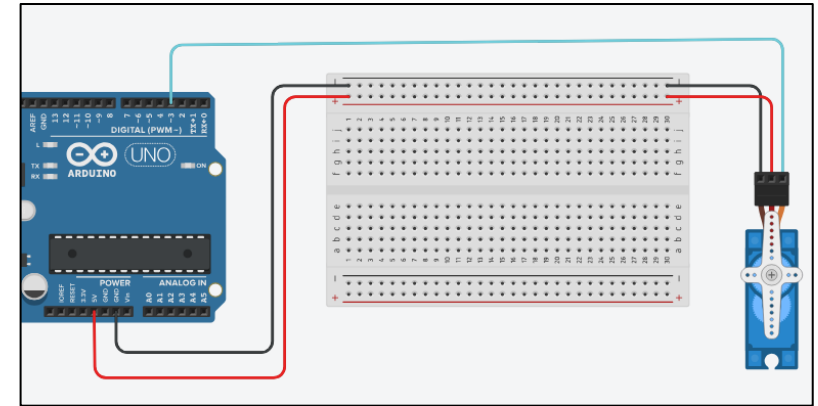


Рисунок 9.1 «Схема підключення серводвигуна»

Джерело: розробка автора

Завдання до виконання

1. Запрограмувати схему згідно *рисунку 9.2* таким чином, щоб серводвигун спочатку повернувся на 0° градусів потім, через 2 секунди на 180° градусів.

```
#include <Servo.h>
#define ServoPin 3

Servo myServo;
int value = 0;

void setup()
{
  myServo.attach(ServoPin);
}

void loop()
{
  value = 0;
  myServo.write(value);
  delay(2000);
  value = 180;
  myServo.write(value);
  delay(2000);
}
```

Рисунок 9.2 «Програмний код»

Джерело: розробка автора

ТЕМА 9 КЕРУВАННЯ ДВИГУНАМИ. СЕРВОДВИГУН

Практична робота №9. Серводвигун

Мета роботи:

- ознайомитись з основами роботи на платформі Arduino;
- повторити принцип роботи тактової кнопки;
- навчитись зчитувати значення з аналогових входів платформи.

Матеріально-технічне забезпечення:

- комп'ютер;
- платформа Arduino;
- потенціометр;
- світлодіод;
- макетна плата;
- серводвигун;
- провідники.

Порядок виконання роботи

1. Повторити теоретичний матеріал за темою [3,5].
2. Повторити схему в подану на *рисунку 9.1* в середовищі Tinkercad.
3. Перевірити чи не має в схемі критичних помилок, які можуть призвести до згорання елементів схеми.
4. Повторити схему в подану на *рисунку 9.1* за допомогою набору робототехніки.
5. Запрограмувати мікроконтролер в середовищі Arduino IDE відповідно до завдання.
6. Сформулювати звіт.

3. Скласти схему згідно *рисунку 4.3*. Запрограмувати схему таким чином щоб на дисплеї відображалось кількість натискань на кнопку *рисунку 4.4*.

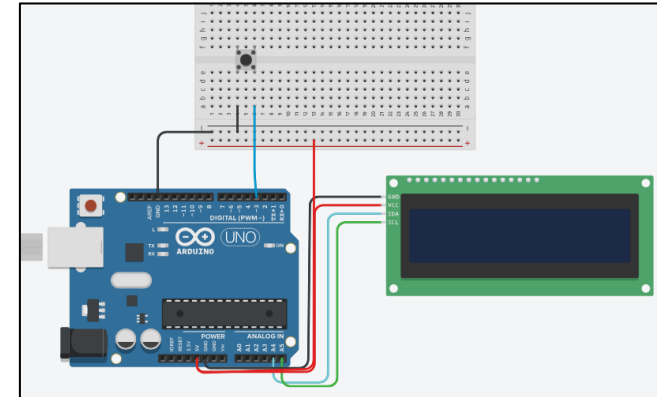


Рисунок 4.3 «Схема підключення дисплея та кнопки»

Джерело: розробка автора

```
#include <LiquidCrystal_I2C.h>
#define Btn1 3

LiquidCrystal_I2C lcd(39,16,2);
int sum = 0;
void setup()
{
    lcd.init();
    lcd.backlight();
    pinMode(Btn1, INPUT_PULLUP);
    attachInterrupt(1, btnPlus, FALLING);
}
void loop()
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(sum);
    delay(50);
}
void btnPlus() {
    sum = sum + 1;
}
```

Рисунок 4.4 «Програмний код»

Джерело: розробка автора

4. Згідно проведеної роботи підготувати письмовий звіт

ТЕМА 5 ВХІДНІ ДАНІ. МАТРИЦЯ КНОПОК

Практична робота №5. Кодовий замок

Мета роботи:

- ознайомитись з основами роботи на платформі Arduino;
- повторити принцип роботи тактової кнопки;
- навчитись керувати дисплеєм;
- навчитись зчитувати значення з цифрових входів платформи.

Матеріально-технічне забезпечення:

- комп'ютер;
- платформа Arduino;
- матриця кнопок;
- макетна плата;
- дисплей;
- провідники.

Порядок виконання роботи

1. Повторити теоретичний матеріал за темою [4,6].
2. Повторити схему в подану на *рисунку 5.1* в середовищі Tinkercad.
3. Перевірити чи не має в схемі критичних помилок, які можуть призвести до згорання елементів схеми.
4. Повторити схему в подану на *рисунку 5.1* за допомогою набору робототехніки.
5. Запрограмувати мікроконтролер в середовищі Arduino IDE відповідно до завдання.
6. Сформулювати звіт.

```
#define trig 3
#define echo 4
#define Zoom 6
long data, S;
void setup()
{
    pinMode(trig, OUTPUT);
    pinMode(Zoom, OUTPUT);
    pinMode(echo, INPUT);
    Serial.begin(9600);
}
void loop()
{
    digitalWrite(trig, LOW);
    delayMicroseconds(2);
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);
    data = pulseIn(echo, HIGH);
    S = data / 29 / 2;
    Serial.println(S);
    delay(S);
    if (S > 5) {
        digitalWrite(Zoom, LOW);
        delay(S);
        digitalWrite(Zoom, HIGH);
    }else{
        digitalWrite(Zoom, HIGH);
    }
    delay(50);
}
```

Рисунок 8.4 «Програмний код»

Джерело: розробка автора

- Додати в схему зумер *рисунок 8.3*, та керувати частотою його вмикання за допомогою Arduino *рисунок 8.4*.

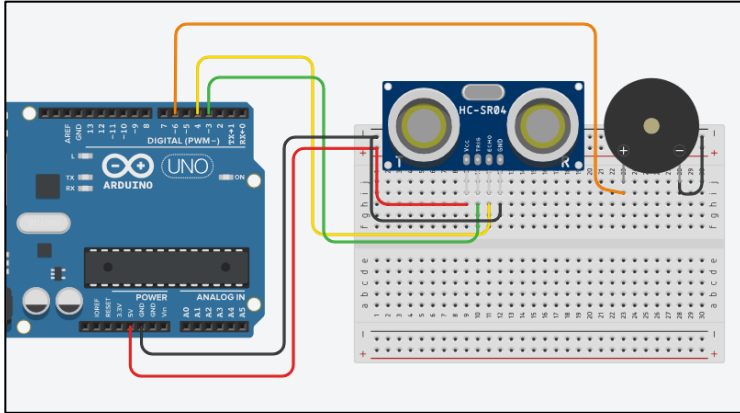


Рисунок 8.3 «Схема підключення дальноміру та зумера»

Джерело: розробка автора

- Додати в схему дисплей, який буде показувати відстань в сантиметрах та дюймах.
- Згідно проведеної роботи підготувати письмовий звіт.

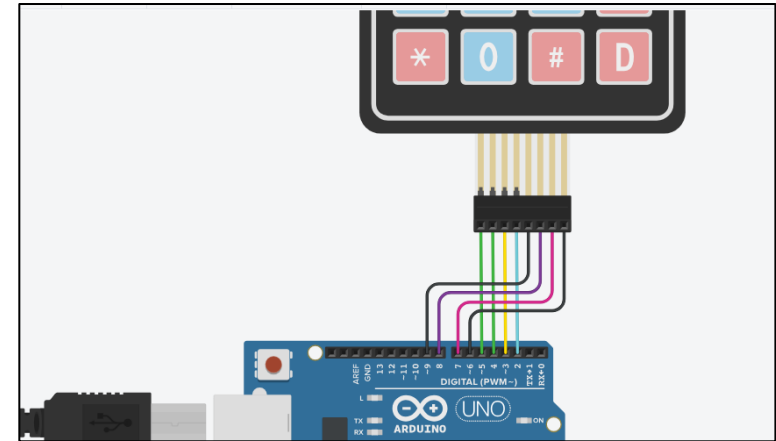


Рисунок 5.1 «Схема підключення матриці кнопок»

Джерело: розробка автора

Завдання до виконання

- Запрограмувати схему згідно *рисунок 5.2* таким чином, щоб значення відповідної клавіші виводилось на монітор послідовного порту.

```
#include <Keypad.h>

const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {5, 4, 3, 2};
byte colPins[COLS] = {9, 8, 7, 6};

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup(){
  Serial.begin(9600);
}

void loop(){
  char key = keypad.getKey();

  if (key){
    Serial.println(key);
  }
}
```

Рисунок 5.2 «Програмний код»

Джерело: розробка автора

- Додати в схему дисплей *рисунок 5.3*, та виводити на дисплей значення нажатої клавіші згідно *рисунок 5.4*.

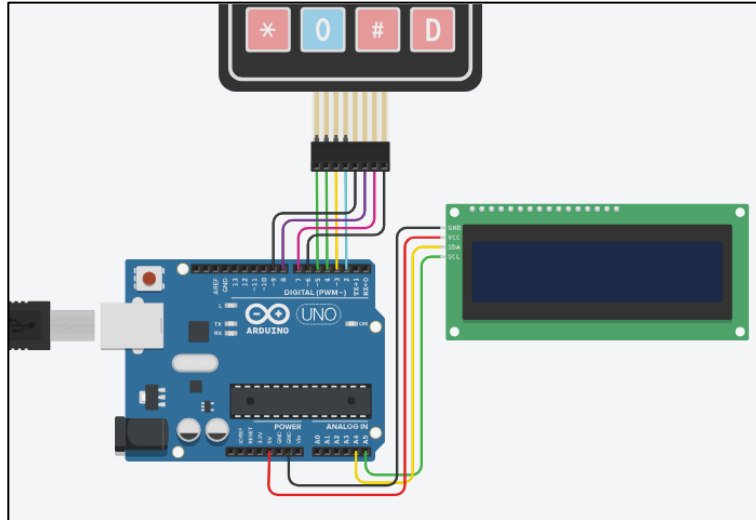


Рисунок 5.3 «Схема підключення матриці кнопок та дисплею»

Джерело: розробка автора

- Написати програму «кодовий замок», При введенні секретного 4 значного коду на дисплеї з'являлось «DONE» або «ERROR»
- Згідно проведеної роботи підготувати письмовий звіт

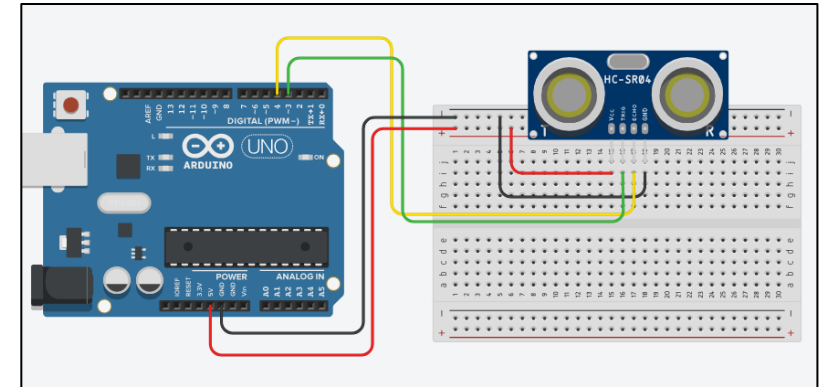


Рисунок 8.1 «Схема підключення дальноміру»

Джерело: розробка автора

Завдання до виконання

- Запрограмувати схему згідно *рисунок 8.2* таким чином, щоб отримувати відстань в сантиметрах.

```
#define trig 3
#define echo 4
long data, S;
void setup()
{
    pinMode(trig, OUTPUT);
    pinMode(echo, INPUT);
    Serial.begin(9600);
}

void loop()
{
    digitalWrite(trig, LOW);
    delayMicroseconds(2);
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);
    data = pulseIn(echo, HIGH);
    S = data / 29 / 2;
    Serial.println(S);
    delay(100);
}
```

Рисунок 8.2 «Програмний код»

Джерело: розробка автора

ТЕМА 8 ЦИФРОВІ СЕНСОРИ. ДАЛЬНОМІР

Практична робота №8. Парктронік

Мета роботи:

- ознайомитись з основами роботи на платформі Arduino;
- повторити принцип роботи дальноміру;
- навчитись керувати дальноміром;
- навчитись зчитувати значення з цифрових входів платформи.

Матеріально-технічне забезпечення:

- комп'ютер;
- платформа Arduino;
- ультразвуковий дальномір;
- зумер;
- макетна плата;
- провідники.

Порядок виконання роботи

1. Повторити теоретичний матеріал за темою [3,4].
2. Повторити схему в подану на *рисунку 8.1* в середовищі Tinkercad.
3. Перевірити чи не має в схемі критичних помилок, які можуть призвести до згорання елементів схеми.
4. Повторити схему в подану на *рисунку 8.1* за допомогою набору робототехніки.
5. Запрограмувати мікроконтролер в середовищі Arduino IDE відповідно до завдання.
6. Сформулювати звіт.

```

1 #include <Keypad.h>
2 #include <LiquidCrystal_I2C.h>
3 LiquidCrystal_I2C lcd(32,16,2);
4 const byte ROWS = 4; //four rows
5 const byte COLS = 4; //three columns
6 char keys[ROWS][COLS] = {
7   {'1','2','3','A'},
8   {'4','5','6','B'},
9   {'7','8','9','C'},
10  {'*','0','#','D'}};
11 };
12 String str;
13 byte rowPins[ROWS] = {5, 4, 3, 2};
14 byte colPins[COLS] = {9, 8, 7, 6};
15 Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
16 void setup(){
17   lcd.init();
18   Serial.begin(9600);
19   lcd.backlight();
20   lcd.setCursor(0,0);
21   lcd.print("Hello, world!");
22 }
23 void loop(){
24   char key = keypad.getKey();
25
26   if (key){
27     str = str+key;
28     lcd.clear();
29     Serial.println(str);
30     lcd.setCursor(0,0);
31     lcd.print(str);
32   }
33 }

```

Рисунок 5.4 «Програмний код»

Джерело: розробка автора

ТЕМА 6 ШИРОТНО-ІМПУЛЬСНА МОДУЛЯЦІЯ

Практична робота №6. Фоторезистор

Мета роботи:

- ознайомитись з основами роботи на платформі Arduino;
- повторити принцип роботи Широтно-імпульсної модуляції;
- навчитись зчитувати значення з аналогових входів платформи.

Матеріально-технічне забезпечення:

- комп'ютер;
- платформа Arduino;
- потенціометр;
- світлодіод;
- макетна плата;
- провідники.

Порядок виконання роботи

1. Повторити теоретичний матеріал за темою [3,4].
2. Повторити схему в подану на *рисунку 6.1* в середовищі Tinkercad.
3. Перевірити чи не має в схемі критичних помилок, які можуть призвести до згорання елементів схеми.
4. Повторити схему в подану на *рисунку 6.1* за допомогою набору робототехніки.
5. Запрограмувати мікроконтролер в середовищі Arduino IDE відповідно до завдання.
6. Сформулювати звіт.

2. Додати в схему світлодіод *рисунок 7.3*, який повинен моргати 4 рази в секунду коли температура більше 40 і світитись коли температура більше 60.

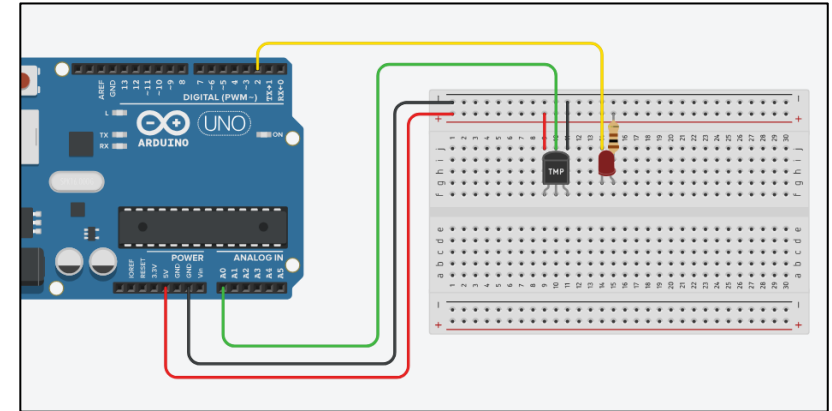


Рисунок 7.3 «Схема підключення сенсору температури і світлодіоду»

Джерело: розробка автора

3. Згідно проведеної роботи підготувати письмовий звіт

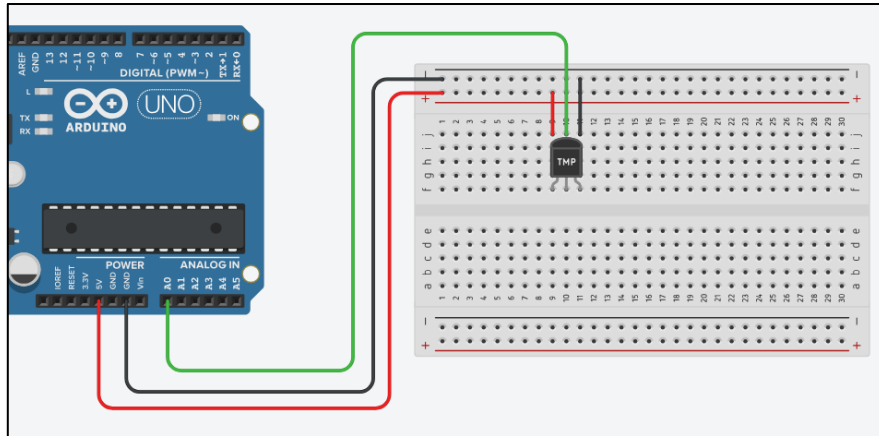


Рисунок 7.1 «Схема підключення сенсору температури»

Джерело: розробка автора

Завдання до виконання

1. Запрограмувати схему згідно *рисунку 7.2* таким чином, щоб в полі монітора послідовного порту виводилась температура в градусах Цельсія.

```
void setup()
{
  pinMode(A0, INPUT);
  Serial.begin(9600);
}

void loop()
{
  int temp = analogRead(A0);
  Serial.println((temp/1023.0)*5.0*1000/10);
}
```

Рисунок 7.2 «Програмний код»

Джерело: розробка автора

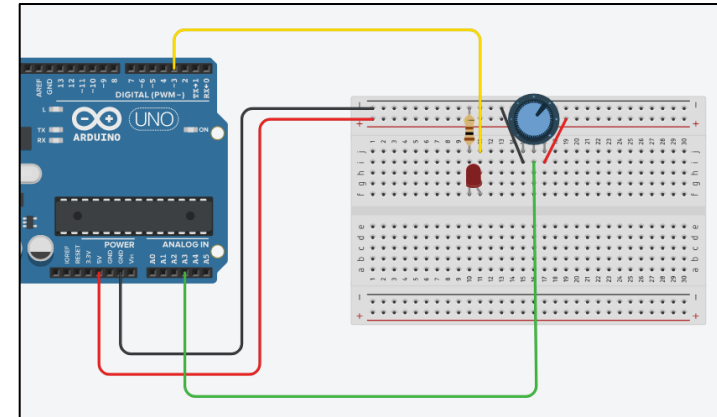


Рисунок 6.1 «Схема підключення потенціометра»

Джерело: розробка автора

Завдання до виконання

1. Запрограмувати схему згідно *рисунку 6.2* таким чином, щоб в полі монітора послідовного порту виводилось значення від 0 до 255, взятого з потенціометру.

```
void setup()
{
  pinMode(A3, INPUT);
  pinMode(3, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  int value = map(analogRead(A3), 0, 1024, 0, 255);
  analogWrite(3, value);
  Serial.println(value);
  delay(50);
}
```

Рисунок 6.2 «Програмний код»

Джерело: розробка автора

- Згідно *рисунку 6.3* замінити потенціометр на фоторезистор, запрограмувати так щоб світлодіод горів коли темно, а коли світло не горів.

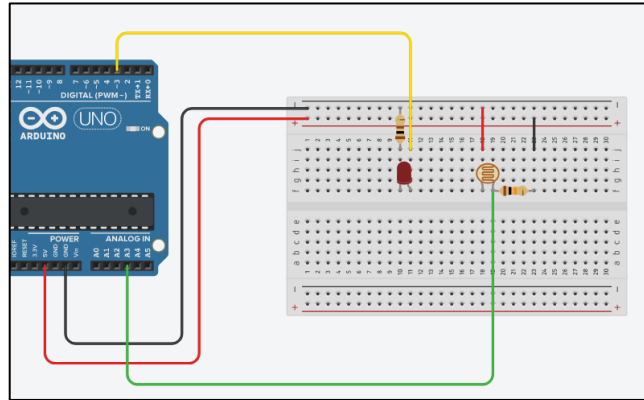


Рисунок 6.3 «Схема підключення фоторезистора»

Джерело: розробка автора

- Згідно проведеної роботи підготувати письмовий звіт

ТЕМА 7 ЦИФРОВІ СЕНСОРИ. ТЕРМОМЕТР

Практична робота №7. Термометр

Мета роботи:

- ознайомитись з основами роботи на платформі Arduino;
- повторити принцип роботи термометрів;
- навчитись зчитувати значення з аналогових входів платформи.

Матеріально-технічне забезпечення:

- комп'ютер;
- платформа Arduino;
- сенсор температури;
- світлодіод;
- макетна плата;
- провідники.

Порядок виконання роботи

- Повторити теоретичний матеріал за темою [3,6].
- Повторити схему в подану на *рисунку 7.1* в середовищі Tinkercad.
- Перевірити чи не має в схемі критичних помилок, які можуть призвести до згорання елементів схеми.
- Повторити схему в подану на *рисунку 7.1* за допомогою набору робототехніки.
- Запрограмувати мікроконтролер в середовищі Arduino IDE відповідно до завдання.
- Сформулювати звіт.