

Міністерство освіти і науки України

Національний університет водного господарства та природокористування

Навчально-науковий інститут енергетики, автоматики та водного господарства

Кафедра автоматизації, електротехнічних та комп'ютерно-інтегрованих
технологій

04-03-345М

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт з навчальної дисципліни
«Мікропроцесорні системи управління та захисту в енергетиці»
(частина 1) для здобувачів вищої освіти другого (магістерського)
рівня за освітньо-професійною програмою «Електроенергетика,
електротехніка та електромеханіка» спеціальності 141
«Електроенергетика, електротехніка та електромеханіка» денної та
заочної форм навчання

Рекомендовано науково-
методичною радою з якості
ННІ ЕАВГ
Протокол № 1 від 26.09.2023 р.

Рівне – 2023

Методичні вказівки до виконання лабораторних робіт з навчальної дисципліни «Мікропроцесорні системи управління та захисту в енергетиці» (частина 1) для здобувачів вищої освіти другого (магістерського) рівня за освітньо-професійною програмою «Електроенергетика, електротехніка та електромеханіка» спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» денної та заочної форм навчання [Електронне видання] / Василюк С. В., Василюк К. С., Ільчук В. В., Вознюк Н. В. – Рівне : НУВГП, 2023. – 59 с.

Укладачі:

- Василюк С. В.** професор кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій, доктор технічних наук, професор
- Василюк К. С.** старший викладач кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій
- Ільчук В. В.** старший лаборант кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій
- Вознюк Н. В.** лаборант кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій

Відповідальний за випуск:

Древецький В. В., завідувач кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій, доктор технічних наук, професор

Керівник групи забезпечення спеціальності 141 «Електроенергетика, електротехніка та електромеханіка»:

Василюк С. В., професор кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій, доктор технічних наук, професор

© С. В. Василюк,
К. С. Василюк,
В. В. Ільчук,
Н. В. Вознюк, 2023
© НУВГП, 2023

ЗМІСТ

Вступ.....	4
Лабораторна робота №1. Програмування графічного дисплея.....	5
Лабораторна робота №2. Розроблення програмного забезпечення для мікропроцесорного пристрою визначення порядку чергування фаз.....	20
Лабораторна робота №3. Програмування мікропроцесорного лічильника для технічного обліку електроенергії, що споживається освітленням цеху.....	29
Лабораторна робота №4. Програмування мікропроцесорного струмового захисту силового приєднання з розпізнаванням аварійного режиму.....	36
Лабораторна робота №5. Програмування мікропроцесорного реле мінімального опору для дистанційного релейного захисту.....	43
Література.....	59

ВСТУП

Метою навчальної дисципліни «Метою освітньої компоненти є формування у студентів навичок та вмінь побудови та експлуатації мікропроцесорних систем управління та захисту електроенергетичних об'єктів.

Завдання: набути знань щодо неперервних та дискретних систем автоматичного управління та релейного захисту; вивчити методи оброблення дискретизованих сигналів мікропроцесорними пристроями в системах управління та захисту; навчитися користуватися серійним мікропроцесорним терміналом релейного захисту.

В результаті вивчення даного курсу студент повинен знати: особливості функціонування неперервних та цифрових систем управління та захисту електроенергетичних об'єктів, математичний опис об'єктів керування та захисту, математичний опис цифрових пристроїв керування та давачів, улаштування цифрових систем управління, особливості побудови цифрових пристроїв релейного захисту, апаратне забезпечення цифрових систем управління та захисту, реалізацію цифрових алгоритмів управління, типові алгоритми цифрового оброблення сигналів мікропроцесорними пристроями в системах управління та захисту, принципи та особливості побудова цифрових органів релейного захисту та автоматики.

Тематика лабораторних робіт тісно пов'язана з лекціями, тому під час підготовки до лабораторної роботи необхідно ґрунтовно вивчити відповідний теоретичний матеріал. Крім того, готуючись до лабораторної роботи, студент має ознайомитися з методичними вказівками. Лабораторні роботи виконуються з використанням плат ARDUINO та моделей мікропроцесорних пристроїв у PROTEUS, програмних продуктів IDE Arduino, Proteus, SinaProg.

За результатами виконання кожної лабораторної роботи студент готує звіт згідно з вимогами, які наведені в кінці вказівок до кожної роботи. Студент має представити та захистити викладачеві звіт на наступному лабораторному занятті. Під час підготовки до захисту слід орієнтуватися на перелік контрольних запитань.

ЛАБОРАТОРНА РОБОТА №1

Програмування графічного дисплея

Мета: оволодіти навичками виведення графічних зображень на дисплей.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Сучасні мікропроцесорні пристрої релейного захисту обладнують графічним дисплеєм, що полегшує сприйняття інформації обслуговуючим персоналом та спрощує налаштування релейного захисту. Наприклад, таким дисплеєм обладнано пристрій релейного захисту та автоматики з функціями вимірювань SIPROTEC 5 фірми Siemens, рис. 1.1. На графічному дисплеї зазвичай відображається однолінійна схема силових кіл об'єкта, що захищається, та поточні значення робочих параметрів (струм, напруга, частота, потужність тощо).



Рисунок 1.1 – Пристрій релейного захисту та автоматики з функціями вимірювань SIPROTEC 5 фірми Siemens

Рідкокристалічний дисплей з активною матрицею (TFT LCD, англ. thin-film transistor – тонкоплівковий транзистор) – різновид рідкокристалічного дисплея, в якому використовується активна матриця, керована тонкоплівковими транзисторами. В такому дисплеї управління пікселями здійснюється за допомогою активної матриці з полікремнієвих тонкоплівкових транзисторів (P-Si TFT). Характеризуються невеликим часом затримки (в основному 30 мс), що дозволяє проектувати зміну зображення практично в реальному часі, і з гарним коефіцієнтом контрастності (100:1). Завдяки таким характеристикам відтворення відеосигналів відбувається з невеликою втратою

якості. Активні матриці характеризуються більшою яскравістю, насиченістю і чіткістю проєктованого зображення в порівнянні з пасивними матрицями.

В лабораторній роботі пропонується розглянути TFT LCD дисплей з діагоналлю 1,8 дюйма, роздільною здатністю 128x160 точок та глибиною кольору 262 тис. кольорів на пиксель, рис. 1.2. Дисплей побудовано на базі контролера ST7735. Характерною особливістю є робота через послідовний інтерфейс SPI. Розміри плати 50 x 37 мм.



Рисунок 1.2 – TFT LCD дисплей на базі контролера ST7735

Керування дисплеєм здійснюється за допомогою наступних виводів:

\overline{CS} – вибір мікросхеми;

SCK – тактовий сигнал;

D/\overline{C} – вибір типу даних, що записуються до дисплея: дані чи команди;

SDA (MOSI) – лінія передачі даних до дисплея.

\overline{RES} – апаратне скидання.

Детальну інформацію щодо дисплея можна знайти за посиланнями:

<https://cutt.ly/qVR24nZ> , <https://cutt.ly/kVR9qSb> .

Для плат Arduino існують бібліотеки, використання яких спрощує виведення графічної інформації:

Adafruit_GFX.h <https://cutt.ly/jVR2LQ8>

Adafruit_ST7735.h <https://cutt.ly/PVR2S03>

Adafruit_GFX – це бібліотека для Arduino, що має у своєму наборі синтаксис та графічні функції для LCD та OLED дисплеїв. Вона дозволяє легко адаптувати скетч Ардуїно для роботи з дисплеєм. Бібліотека Adafruit_GFX

завжди працює в парі зі спеціалізованою бібліотекою для дисплея. В даному випадку для TFT LCD дисплея з чіпом ST7735 необхідно встановлювати дві бібліотеки: Adafruit_GFX і Adafruit_ST7735. Також передбачається використання бібліотеки SPI.h для передачі даних за протоколом SPI.

В програмі на мові C for Arduino для підключення даних бібліотек необхідно скористатися директивами:

```
#include <Adafruit_GFX.h>
#include <Adafruit_ST7735.h>
#include <SPI.h>
```

Ініціалізація дисплея

Об'єктна змінна, що відповідає дисплею, створюється командою:

```
//tft - об'єктна змінна, що відповідає дисплею
Adafruit_ST7735 tft = Adafruit_ST7735(10, 9);
//10 - порт Arduino, до якого підключено вивід /CS дисплея
//9 - порт Arduino, до якого підключено вивід DC дисплея
```

Дисплей ініціалізується за допомогою команди:

```
tft.initR(INITR_BLACKTAB); // ініціалізація контролера ST7735S у складі дисплея з чорним корпусом
```

Система координат

Пікселі – це блоки, з яких складаються цифрові картинки. Піксель має фіксований розмір висоти та довжини (квадрат), розмір пікселя залежить від роздільної здатності дисплея. Кожен піксель позиціонується на екрані за горизонтальною X віссю і вертикальною Y віссю, рис. 1.3.

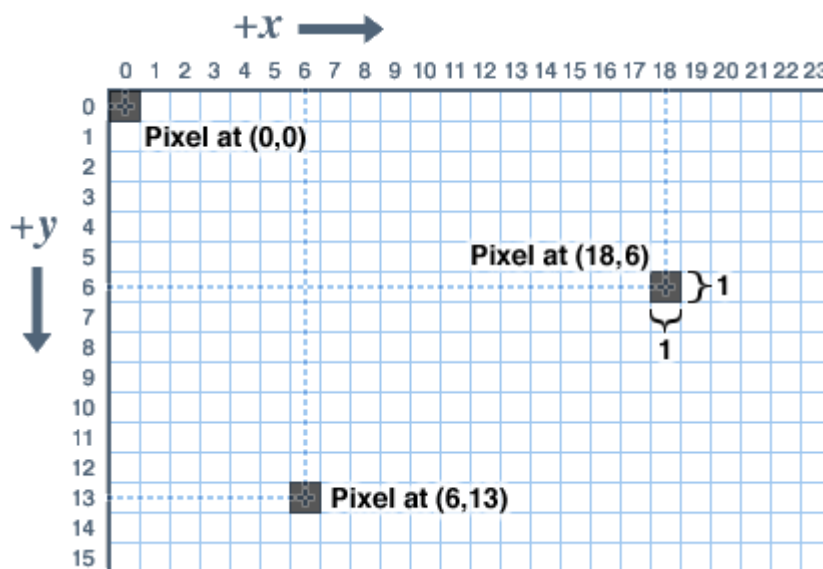


Рисунок 1.3 – Координатна сітка дисплея

Система координат починається в лівому верхньому куті екрана з точки $X=0$, $Y=0$ і продовжується з позитивним збільшенням осі X вправо, осі Y вниз. Залежно від орієнтації дисплея "Портрет", "Пейзаж" або іншого специфічного розташування, можна вказати який із чотирьох кутів буде позначений як лівий верхній (початкова точка осей).

Для визначення ширини w та висоти h екрана (кількість пікселів) можна скористатися виразами:

```
w=tft.width();
h=tft.height();
```

Кольори

Для кольорових дисплеїв кольори представлені в беззнаковому 16-бітному форматі. Деякі з дисплеїв можуть передавати кольори з великою кількістю бітів, але бібліотека оперує лише 16 бітними значеннями, це спрощує роботу Ардуїно з різними дисплеями. Колір складається з трьох основних кольорів – червоний, зелений, синій, які упаковані в одному 16-бітному значенні, рис. 1.4. Старші 5 біт для червоного, середні 6 біт для зеленого, молодші 5 біт для синього. Людське око більш чутливе до зеленого кольору, тому для нього виділено на 1-біт більше.

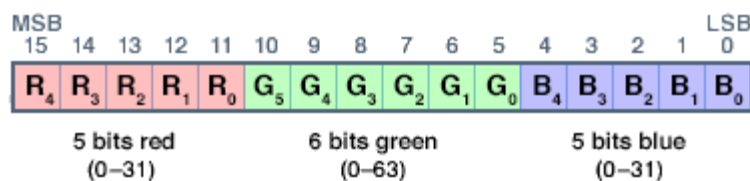


Рисунок 1.4 – Двобайтовий формат представлення кольору

До складу бібліотеки Adafruit_ST7735 входять стандартні кольори, назва яких починається з «ST7735_», наприклад: ST7735_RED, ST7735_YELLOW тощо. Також можна самостійно задати необхідний колір. Наприклад:

```
#define BLACK 0x0000
#define BLUE 0x001F
#define RED 0xF800
#define GREEN 0x07E0
#define CYAN 0x07FF
#define MAGENTA 0xF81F
#define YELLOW 0xFFE0
#define WHITE 0xFFFF
```

Загальні графічні функції

У бібліотеці для кожного дисплея є свої конструкції та функції ініціалізації. Вони описані в окремих інструкціях кожного типу дисплея. Також їх можна знайти в заголовному файлі бібліотеки дисплея. Далі описано загальні графічні функції, які працюють однаково незалежно від типу дисплея.

Очищення та заливка екрана

Функція fillScreen() заливає дисплей вибраним кольором, стираючи все, що є на екрані:

```
fillScreen(uint16_t color);
```

Наприклад:

```
tft.fillScreen(ST7735_YELLOW);
```

Піксель (кранка)

Зміни кольору пікселя на екрані необхідно вказати його координати і колір:

```
drawPixel(uint16_t x, uint16_t y, uint16_t color);
```

Наприклад, якщо дисплею відповідає об'єктна змінна tft, для зміни кольору пікселя необхідно виконати команду:

```
tft.drawPixel (10,20,ST7735_RED);
```

```
//10 – x координата розташування пікселя на екрані,
```

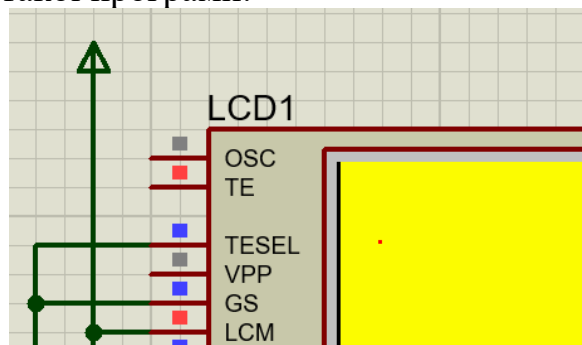
```
//15 – y координата розташування пікселя на екрані,
```

```
// ST7735_RED – колір пікселя.
```

Програма, що ініціалізує дисплей і виводить червону точку на жовтому фоні має вигляд:

```
1 #include <Adafruit_GFX.h>
2 #include <Adafruit_ST7735.h>
3 #include <SPI.h>
4
5 Adafruit_ST7735 tft = Adafruit_ST7735(10, 9);
6
7 void setup() {
8     tft.initR(INITR_BLACKTAB);
9     tft.fillScreen(ST7735_YELLOW);
10    tft.drawPixel(10,20,ST7735_RED);
11 }
12
13 void loop() {
14
15 }
```

Результат виконання такої програми:



Лінія

Для креслення лінії використовуються початкова (x_0, y_0) та кінцева (x_1, y_1) точка лінії, рис. 1.5.

```
drawLine(uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1, uint16_t color);
```

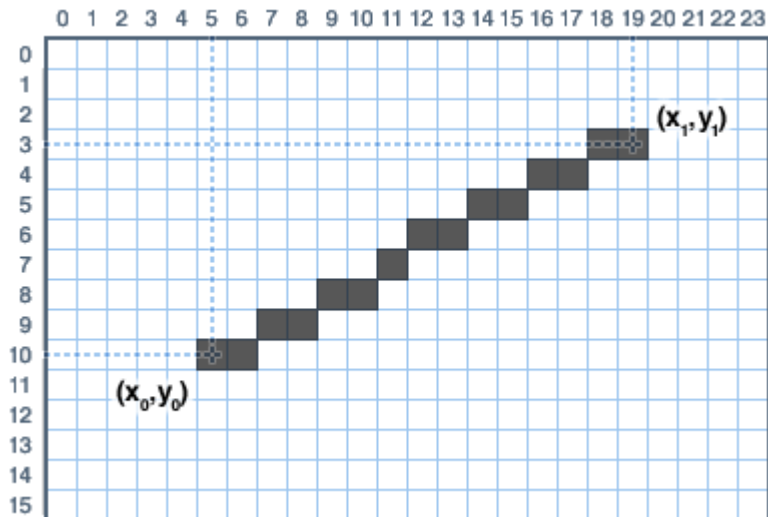


Рисунок 1.5 – Координати лінії

Наприклад, для побудови зеленої лінії, початок якої відповідає початку координата, а кінець – точці $(20, 50)$ необхідно скористатися командою:

```
tft.drawLine(5, 10, 19, 3, ST7735_GREEN);
```

Для горизонтальних та вертикальних ліній існує більш спрощена функція. У ній вказується початкові координати лінії, її довжина в пікселях та колір:

```
drawFastVLine(uint16_t x0, uint16_t y0, uint16_t length, uint16_t color);  
drawFastHLine(uint16_t x0, uint16_t y0, uint16_t length, uint16_t color);
```

Прямокутники

Для креслення прямокутників використовуються координати верхнього лівого кута прямокутника x_0, y_0 , ширина w , висота h , колір, рис. 1.6.

`drawRect()` – креслить лінію за межами сторін прямокутника:

```
drawRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t color);
```

`fillRect()` – креслить прямокутник, площа якого залита вказаним кольором:

```
fillRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t color);
```

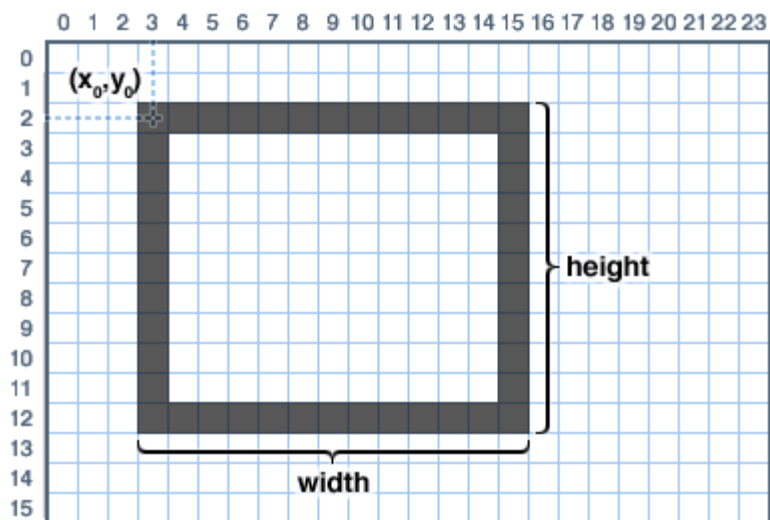


Рисунок 1.6 – Параметри прямокутника

Коло

Функція виведення кола використовує такі значення: центральні координати кола x_0, y_0 , радіус r , колір, рис. 1.7.

`drawCircle()` – креслить коло:

`drawCircle(uint16_t x0, uint16_t y0, uint16_t r, uint16_t color);`

`fillCircle()` – креслить коло, площа якого залита вибраним кольором.

`fillCircle(uint16_t x0, uint16_t y0, uint16_t r, uint16_t color);`

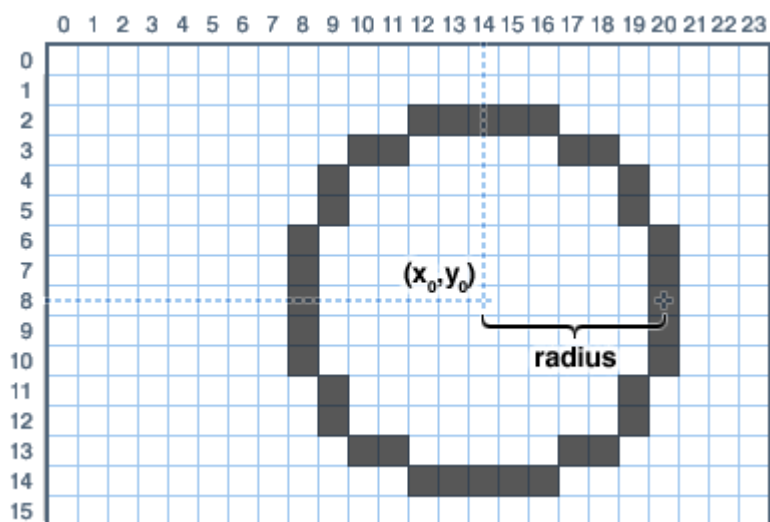


Рисунок 1.7 – Параметри кола

Прямокутник із заокругленими кутами

Функція, що креслить прямокутник із заокругленими кутами, використовує: початкові координати лівого верхнього кута прямокутника x_0, y_0 , ширину w , висоту h , радіус r заокруглення кута в пікселях та колір, рис. 1.8.

`drawRoundRect()` – креслить прямокутник із округленими кутами:

`drawRoundRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t radius, uint16_t color);`

`fillRoundRect()` – креслить прямокутник із округленими кутами, площа якого залита вибраним кольором:

`fillRoundRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t radius, uint16_t color);`

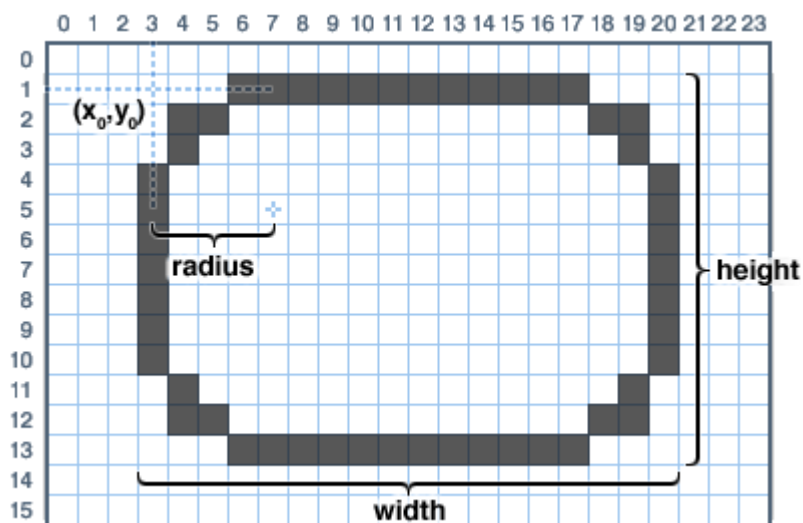


Рисунок 1.8 – Параметри прямокутника із заокругленими кутами

Трикутник

Функція креслення трикутника використовує такі параметри: початкові координати першого кута (x_0, y_0), координати другого кута (x_1, y_1), координати третього кута (x_2, y_2), колір, рис. 1.9.

`drawTriangle()` – промальовує трикутник:

`drawTriangle(uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1, uint16_t x2, uint16_t y2, uint16_t color);`

`fillTriangle()` – промальовує трикутник площа якого залита вибраним кольором:

`fillTriangle(uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1, uint16_t x2, uint16_t y2, uint16_t color);`

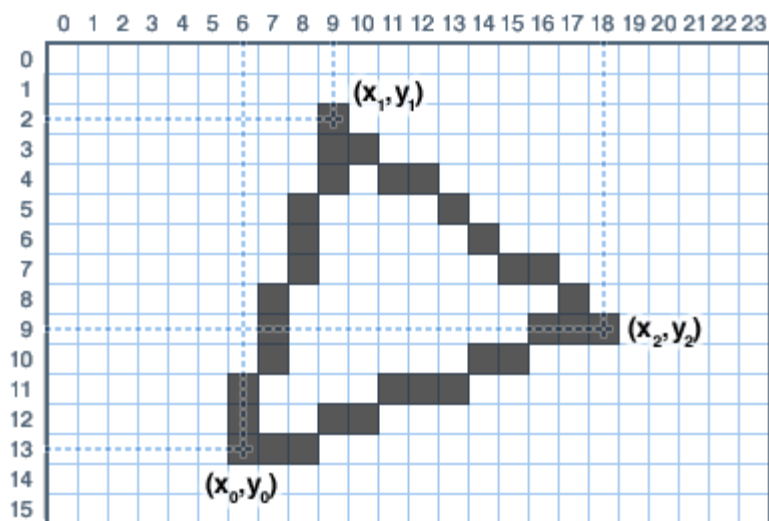


Рисунок 1.9 – Параметри трикутника

Відображення символу

Для виведення єдиного символу використовується функція `drawChar`. Ви можете помістити один символ у будь-яке місце екрану та вказати цьому символу потрібний колір, рис. 1.10. Основний розмір символу 5x8 пікселів. Але розміри можна змінювати пропорційно. Наприклад, задавши символ розмір `size=2`, розмір символу стане 10x16 пікселів.

```
drawChar(uint16_t x, uint16_t y, char c, uint16_t color, uint16_t bg, uint8_t size);
```

Параметри функції `drawChar`:

`x` – абсциса символу,

`y` – ордината символу,

`c` – символ, що виводиться, повинен бути в одинарних лапках,

`color` – колір символу,

`bg` – колір фону символу,

`size` – розмір символу.

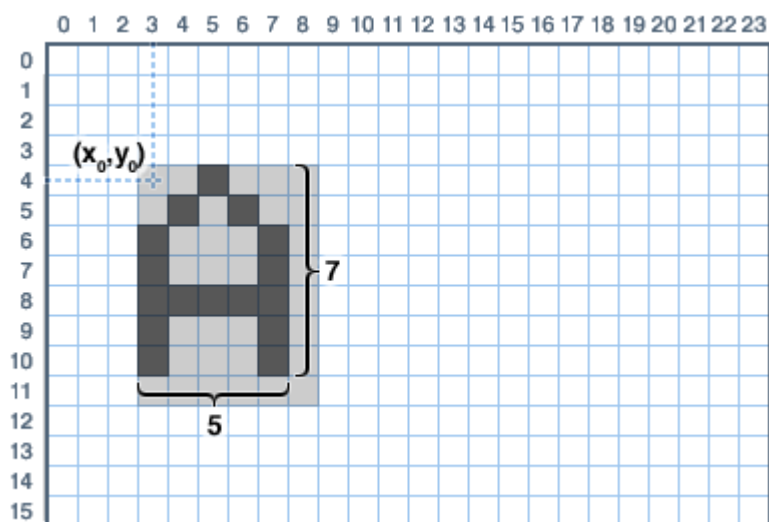
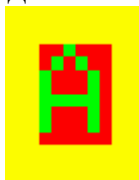


Рисунок 1.10 – Параметри символу

Наприклад, для виведення зеленої літери А на червоному фоні використовується команда:

```
tft.drawChar (100, 100, 'A', ST7735_GREEN, ST7735_RED, 2);
```

Результатом виконання команди є:



Виведення рядка символів (текста)

Виведення рядка символів (текст) відрізняється від виведення одного символу. Для того, щоб встановити параметри для рядка, потрібно використовувати кілька функцій. Задавши координати, колір та розміри в

різних функціях, рядок виводиться функцією `print()`, яка використовує задані раніше параметри. Нижче наведено набір функцій для виведення рядків.

```
setCursor(uint16_t x0, uint16_t y0);
setTextColor(uint16_t color);
setTextColor(uint16_t color, uint16_t backgroundColor);
setTextSize(uint8_t size);
setTextWrap(boolean w);
```

Насамперед, необхідно задати координати верхнього лівого кута рядка за допомогою функції `setCursor(x, y)`. Після цього задається колір (`color`) тексту за допомогою функції `setTextColor(color, backgroundColor)`. Параметр `backgroundColor` визначає колір фону, на якому виводиться текст, і не є обов'язковим. За допомогою функції `setTextSize(size)` задається розмір символів у рядку. За замовчуванням значення `size=1`, допустимі значення 2 та 3.

За замовчуванням, довгі рядки виводяться зліва направо. Перенесення тексту на новий рядок забезпечує функція `setTextWrap(boolean w)`. При `w=1` відбувається перенесення тексту на новий рядок, якщо текст довший за розмір екрану. При `w=0` довгий рядок не буде переноситися і вийде за межі екрана, така можливість дозволяє реалізувати рухомий рядок.

Для виведення рядка символів на екран використовується функція `print()` або `println()`. Особливістю другої функції є переведення курсора на новий рядок.

Приклад виведення текстового рядка:

```
// Встановлюємо курсор (x=5, y=3)
```

```
tft.setCursor(5, 3);
```

```
// Задаємо колір тексту (синій) та колір фону (білий)
```

```
tft.setTextColor(ST7735_BLUE, ST7735_WHITE);
```

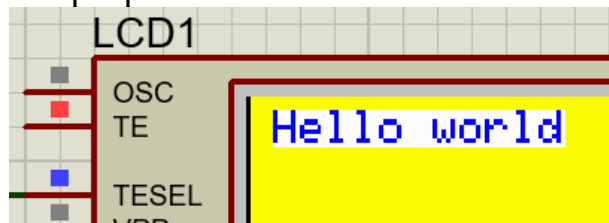
```
// Вказуємо розмір символів у рядку (2)
```

```
tft.setTextSize(1);
```

// Виводимо текст на екран, текст буде виведений із зазначеними вище налаштуваннями

```
tft.print("Hello world");
```

Результат роботи програми:



Поворот екрану

Існує можливість змінювати орієнтацію виведеної графіки та тексту на екрані. Ця функція не змінює положення вже виведених елементів, але змінить розташування системи координат наступного виведення. Функція `setRotation()` може застосувати лише один раз, у розділі `setup()`. Можна повернути екран

лише на 0, 90, 180 або 270 градусів. Інші кути вимагають більш потужного обладнання, це унеможлиблює їх обчислення на Arduino.

`setRotation(uint8_t rotation);`

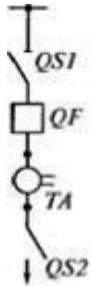
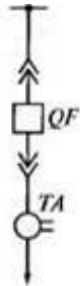



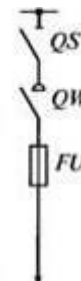
де `rotation` – кут повороту екрана в градусах: 0 = 0°, 1 = 90°, 2 = 180°, 3 = 270°.

ЗАВДАННЯ

Написати програму для плати ArduinoUNO, що виводить на дисплей ST7735 схему з'єднання головних кіл комірки відповідно до табл. 1.1.

Таблиця 1.1

Завдання для виконання

Варіант	1	2	3
Назва схеми комірки	Комірка КСО з шинним роз'єднувачем, вимикачем, трансформаторами струму, лінійним роз'єднувачем	Комірка КРУ з викатним вимикачем	Комірка КРУ з запобіжниками
Схема з'єднання головних кіл комірки			
Варіант	4	5	6
Назва схеми	Комірка КСО з вимикачем навантаження та запобіжниками	Комірка КСО з вимикачем навантаження і запобіжником	Комірка КСО з вимикачем навантаження і шинним роз'єднувачем
Схема з'єднання головних кіл комірки			

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Виконати необхідне креслення схеми з'єднання головних кіл комірки в піксельній системі координат дисплея, рис. 1.11.

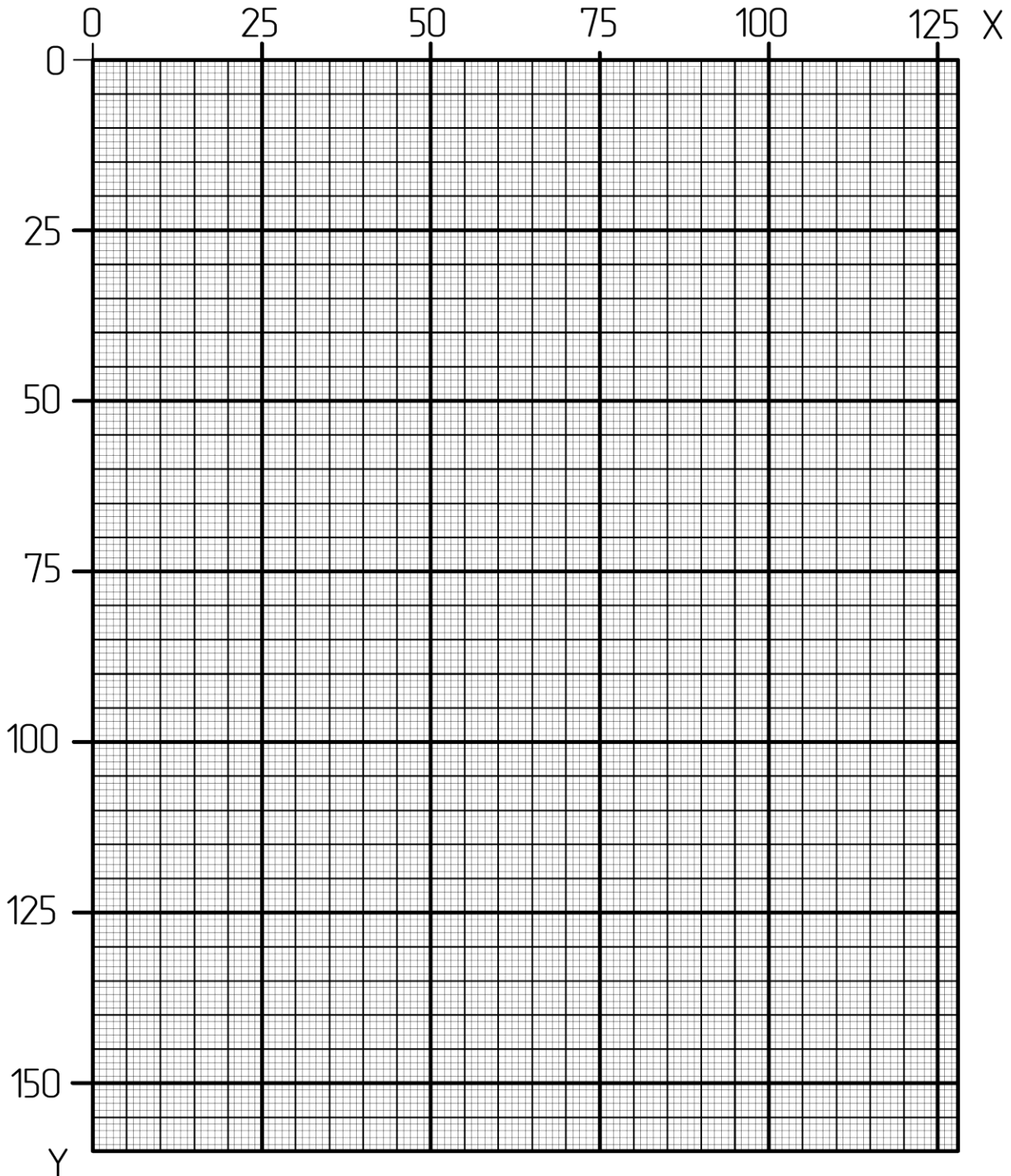


Рисунок 1.11 – Піксельна система координат дисплея 128x160

2. Визначити координати примітивних графічних елементів (лінія, коло, прямокутник тощо), з яких складається креслення, рис. 1.12.

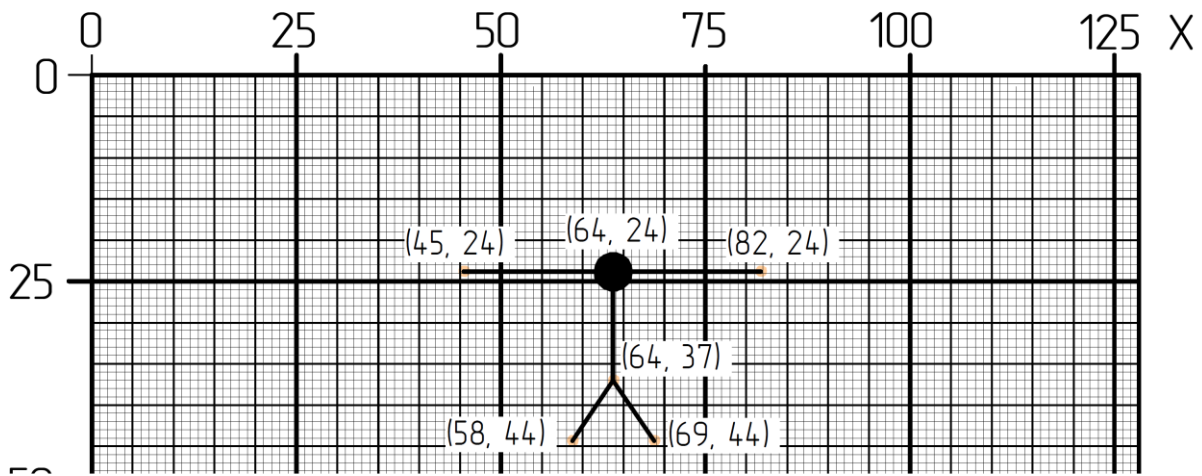


Рисунок 1.12 – Визначення координат графічних примітивів

3. Скласти в симуляторі Proteus модель відповідно до рис. 1.13.

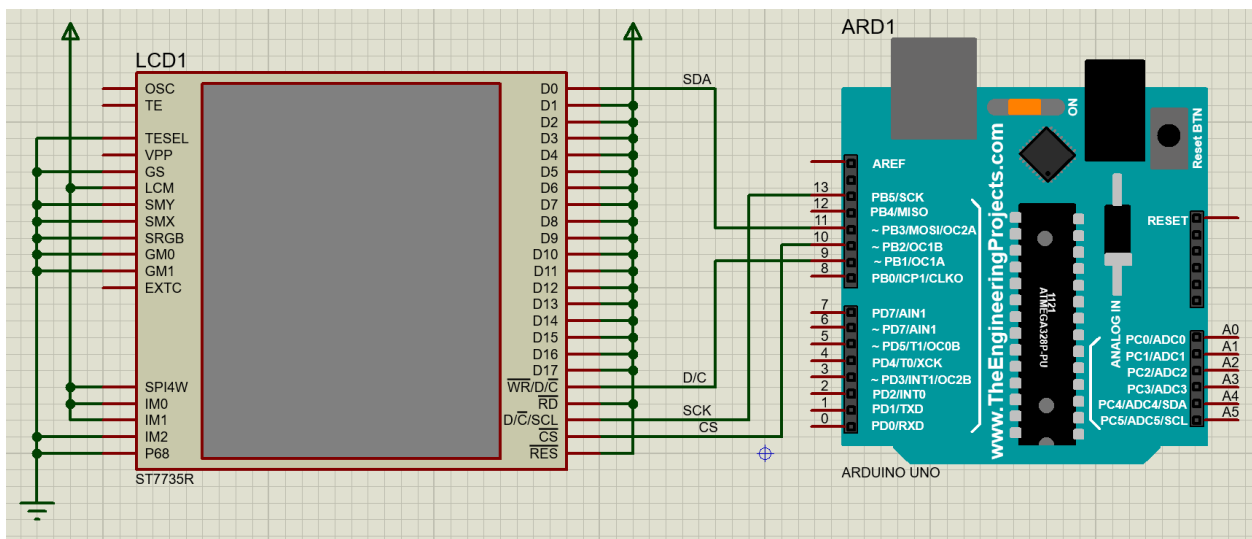


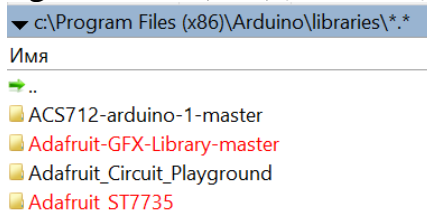
Рисунок 1.13 – Модель, що відповідає схемі приєднання TFT LCD дисплей на базі контролера ST7735 до плати ArduinoUNO

4. Завантажити бібліотеки

Adafruit_GFX.h <https://cutt.ly/jVR2LQ8>

Adafruit_ST7735.h <https://cutt.ly/PVR2S03>

Вказані бібліотеки необхідно розпакувати і папки з бібліотеками розмістити в каталозі: c:\Program Files (x86)\Arduino\libraries\ :



5. Створити новий скетч в середовищі ArduinoIDE.

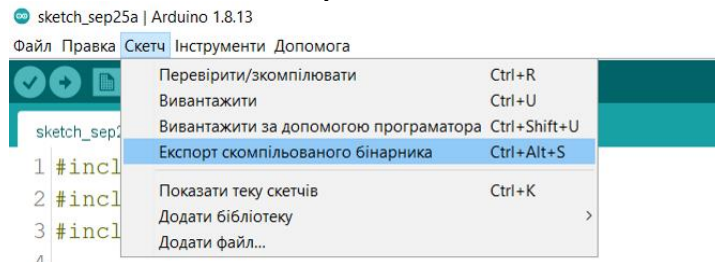
6. Написати програму для плати Arduino, що виводить задане графічне зображення на дисплей. В якості додаткового джерела інформації можна користуватися тестовою програмою:

<https://simple-circuit.com/arduino-st7735r-tft-proteus-simulation/>

Результат роботи програми:

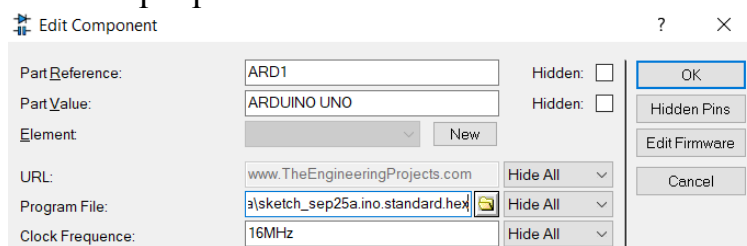
<https://www.youtube.com/watch?v=-5Yr76sXjB4>

7. Для компіляції програми в середовищі Arduino виконати команду меню Скетч–Експорт скомпільованого бінарника.

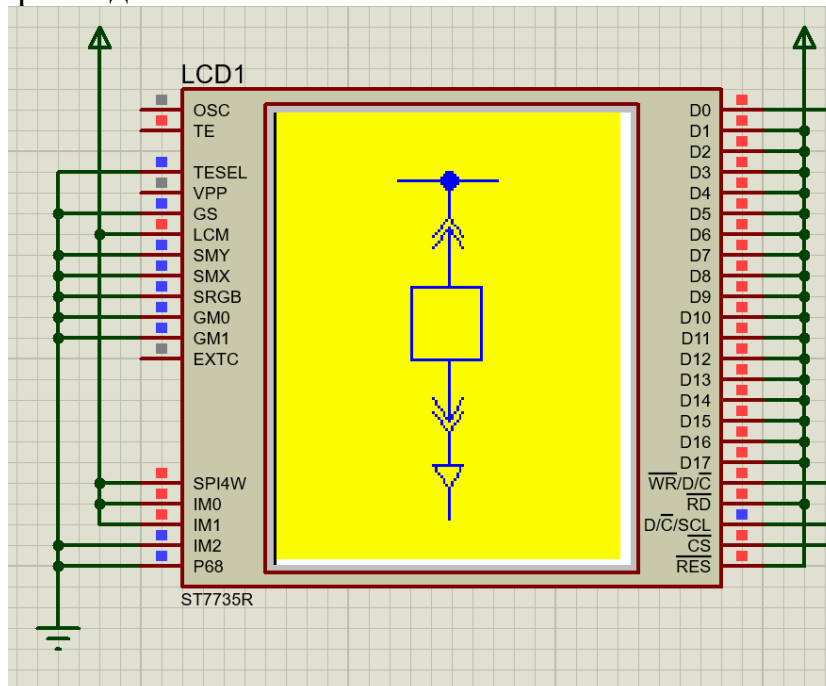


В папці проекту Arduino з'явиться файл ...ino.standard.hex з кодами скомпільованої програми.

8. В Proteus-модель мікроконтролера вказати шлях до вказаного файлу з кодами скомпільованої програми.



9. За відсутності помилок на модель дисплея відобразить необхідне креслення, наприклад:



Вміст звіту з лабораторної роботи

1. Тема, мета роботи.
2. Піксельна система координат з визначенням координат графічних примітивів.
3. Програма для виведення графічного зображення на дисплей.
4. Вигляд моделі у Proteus під час моделювання (на дисплей виводиться графічне зображення).
5. Висновки.

Контрольні запитання

1. Охарактеризуйте TFT LCD дисплеї.
2. Охарактеризуйте інтерфейс SPI.
3. Які бібліотеки можна використовувати для виведення графічної інформації?
4. Опишіть систему координат графічного дисплея.
5. Які функції використовуються для визначення розмірів дисплея?
6. Охарактеризуйте представлення кольору в кольорових дисплеях.
7. За допомогою яких функцій можна змінити колір пікселя та вивести лінію? Які параметри мають дані функції?
8. Які функції використовуються для креслення прямокутників? Які параметри мають дані функції?
9. Які функції використовуються для креслення кіл? Які параметри мають дані функції?
10. Яким чином відобразити символ або рядок символів? Які параметри мають дані функції?

ЛАБОРАТОРНА РОБОТА №2

Розроблення програмного забезпечення для мікропроцесорного пристрою визначення порядку чергування фаз

Мета: навчитися аналізувати аналогові сигнали за допомогою мікроконтролера.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Принцип визначення порядку чергування фаз

Трифазна симетрична система ЕРС являє собою сукупність трьох синусоїдних електрорушійних сил, які мають однакову амплітуду та частоту, та які зсунуті за фазою на 120 ел. градусів (рис. 2.1, а). Це визначає зсув на 120 ел. градусів і векторів, що відображають фазні ЕРС (рис. 2.1, б). При цьому фаза В відстає від фази А на 120 градусів, а фаза С відстає від фази А на 204 градусів, тобто випереджає фазу А на 120 градусів.

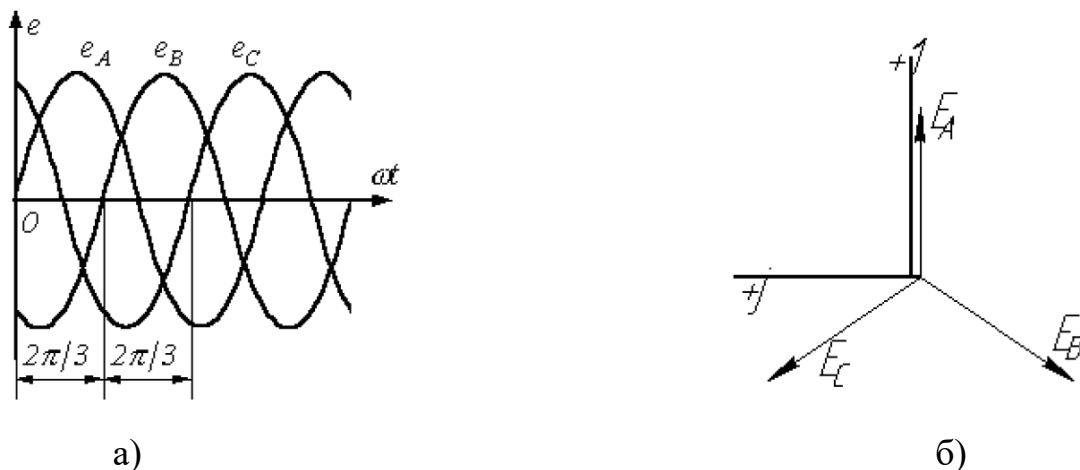


Рисунок 2.1 – Трифазна система синусоїдних електрорушійних сил (а) та відповідна векторна діаграма (б)

Припустимо, що три ідентичні котушки розташовані таким чином, щоб їх осі були зсунуті на 120° (рис. 2.2) і по котушках пропускають струми, які описуються наступними синусоїдними функціями:

$$i_1 = I_m \sin \omega t;$$

$$i_2 = I_m \sin(\omega t - 120^\circ);$$

$$i_3 = I_m \sin(\omega t + 120^\circ).$$

Тоді котушки утворять обертове електричне поле, частота обертання якого визначається частотою змінного струму, що проходить через котушки. Напрямок обертання поля визначається зсувами фаз струмів. Такий принцип покладений в основу роботи трифазного асинхронного та синхронного двигунів.

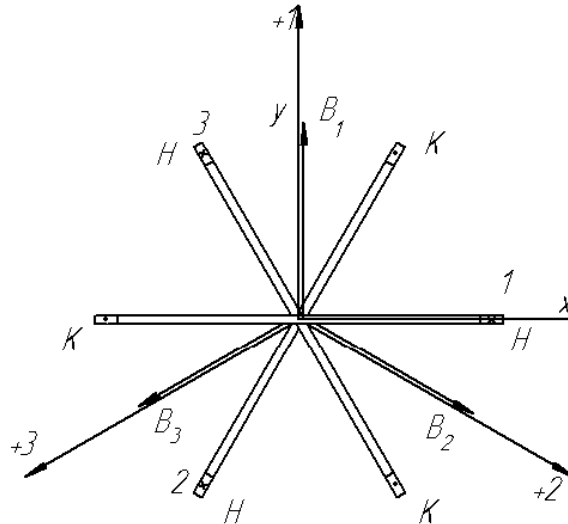


Рисунок 2.2 – Утворення обертового магнітного поля трьома котушками, що зсунуті на 120 ел. градусів

Для правильного підключення двигунів до мережі необхідно знати послідовність чергування фаз, оскільки при переключенні будь-яких двох фаз, наприклад – фаз В та С, двигун обертається в іншу сторону, що може спричинити пошкодження технологічного обладнання.

Визначення порядку чергування фаз трифазної симетричної системи ЕРС може бути здійснено за допомогою вказівника послідовності чергування фаз, який включає дві однакові лампи розжарювання HL1, HL2 та конденсатор С (рис. 2.3). Якщо використовуються лампи розжарювання потужністю P , то величину ємності обирають, виходячи з тотожності ємнісного опору конденсатора X_C та активного опору R лампи:

$$R = X_C. \quad (1)$$

Оскільки

$$R = \frac{U^2}{P}, \quad X_C = \frac{1}{\omega C}, \quad (2)$$

то

$$C = \frac{P}{\omega U^2}. \quad (3)$$

В трифазній мережі 380 В фазна напруга «зірки» дорівнює $U = 220$ В. При використанні ламп потужністю $P=15$ Вт, необхідна ємність конденсатора становить:

$$C = \frac{P}{\omega U^2} = \frac{15}{314 \cdot 220^2} \approx 1 \text{ мкФ} \quad (4)$$

При підключенні вказівника, відповідно до схеми рис. 3, до трифазної електромережі, то потенціал нульової точки буде визначатися положенням точки O' на векторній діаграмі рис. 2.4 (відповідає точці p).

Принцип дії пристрою визначення порядку чергування фаз

В основу роботи пристрою визначення порядку чергування фаз покладено принцип контролю напруги на одній з ламп. Якщо напруга буде перевищувати уставку, що відповідає яскравому горінню лампи, то на дисплеї навпроти відповідного затискача яскравої лампи, має з'явитися позначка «фаза В», біля затискача тьмяної лампи відображається позначка «фаза С». При переключенні фаз позначки міняються. Біля затискача, що відповідає конденсатору, має відображатися позначка «фаза А».

Такий принцип дії може реалізувати мікропроцесорний пристрій, структурна схема якого наведена на рис. 2.5. Пристрій включає ламповий вказівник послідовності чергування фаз (ВПФ), схема кого відповідає рис. 2.3. Напруга з однієї з ламп розжарювання подається на перетворювач напруги (ПН), який забезпечує гальванічну розв'язку та масштабування сигналу за рахунок наявності вимірювального трансформатора напруги. Сигнал, що є пропорційним миттєвим значенням напруги на визначеній лампі, подається до аналого-цифрового перетворювача АЦП, де перетворюється в цифровий код. Цей код надходить до мікроконтролера МК, який за миттєвими значеннями обчислює діюче значення напруги та порівнює його з уставкою. За результатами порівняння на дисплеї LCD, що розташований біля силових затискачів пристрою, відображається повідомлення про назви фаз.

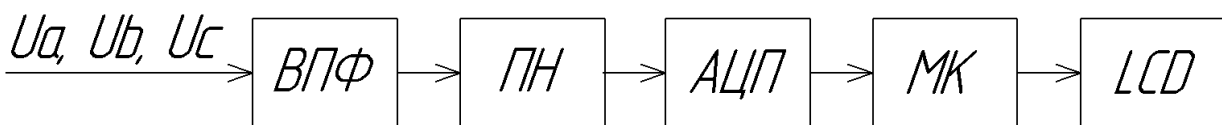


Рисунок 2.5 – Структурна схема мікропроцесорного пристрою визначення порядку чергування фаз трифазної електромережі

Принципова електрична схема пристрою

Принципова електрична схема мікропроцесорного пристрою визначення порядку чергування фаз трифазної електромережі, що відповідає структурній схемі (рис. 2.5), наведена на рис. 2.6.

Схема включає лампи розжарювання HL1, HL2 та конденсатор С, що з'єднані за схемою «зірка». Силовий затискач, що відповідає конденсатору, позначений як фаза «А». Силовий затискач лампи HL1 позначений Х1, лампи HL2 – Х2. Між затискачем Х2 та штучною нульовою точкою паралельно до лампи HL2 підключено схему DA1 вимірювача напруги. Така схема типу ZMPT101В включає вимірювальний трансформатор струму та схему на операційному підсилювачі, що масштабує вимірювальний сигнал та зсуває його у додатну напівплощину (рис. 2.7).

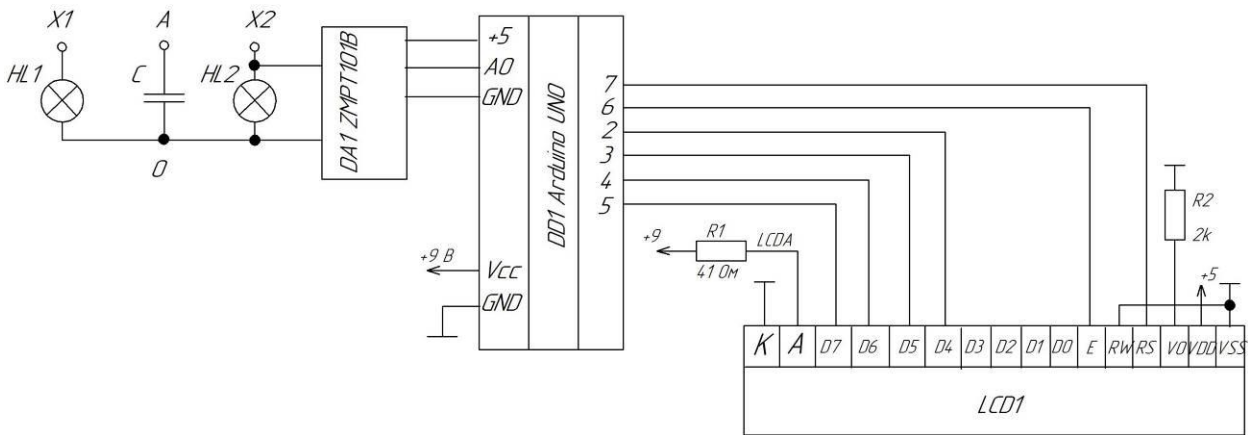
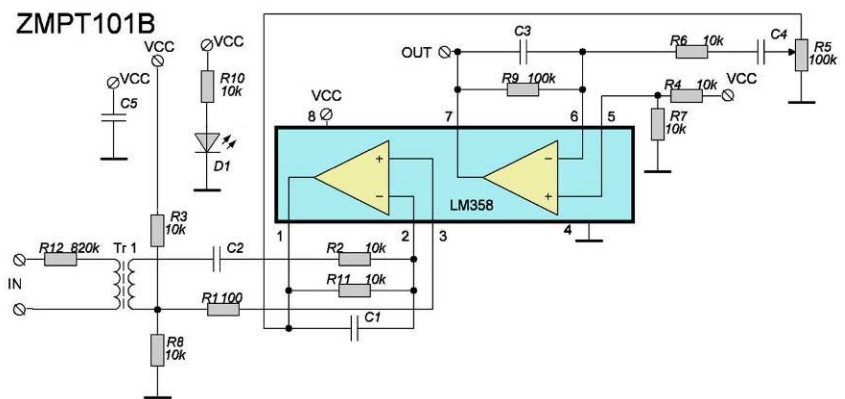
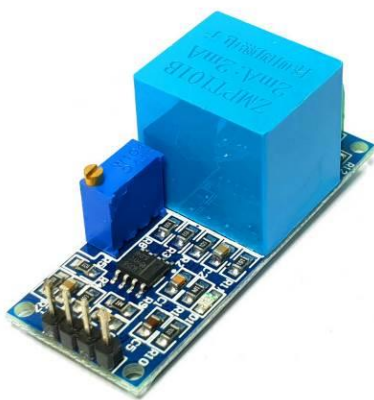


Рисунок 2.6 - Принципова електрична схема мікропроцесорного пристрою визначення порядку чергування фаз трифазної електромережі



а)

б)

Рисунок 2.7 – Давач напруги типу ZMPT101B:
а – загальний вигляд, б – принципова схема

В якості мікропроцесору використовується плата Arduino UNO, що обладнана мікроконтролером ATmega328p. Такий мікроконтролер має вбудований аналого-цифровий перетворювач з аналоговим мультиплексором на вході. Оскільки пристрій має контролювати одну аналогову величину, то в схемі використовується вхідний канал мультимплектора, що позначений A0. Вбудоване АЦП має розрядність 10 біт, що визначає 1024 рівні квантування вхідного сигналу. Вхідний сигнал може змінюватися від 0 до +5 В відносно GND.

Для відображення порядку чергування фаз використовується рідкокристалічний індикатор LCD1 типу WH1602A, що підключається до мікроконтролера за допомогою паралельного 4x розрядного інтерфейсу. Шина даних утворена виводами D4-D7. Вивід RS використовується для подачі дисплею інформації, дані чи команди передаються по шині даних. Вивід RW дозволяє записувати до дисплея або зчитувати дані, цей вивід приєднано до GND, тим самим дисплей налаштований для роботи тільки в режимі прийому даних. Вивід E – строб-сигнал, перепад сигналу з лог. «1» до лог. «0» на якому визначає момент зчитування дисплеєм стану шини даних.

На рис. 2.8 зображено пристрій для визначення порядку чергування фаз, виконаний відповідно до принципової схеми на рис. 2.6.

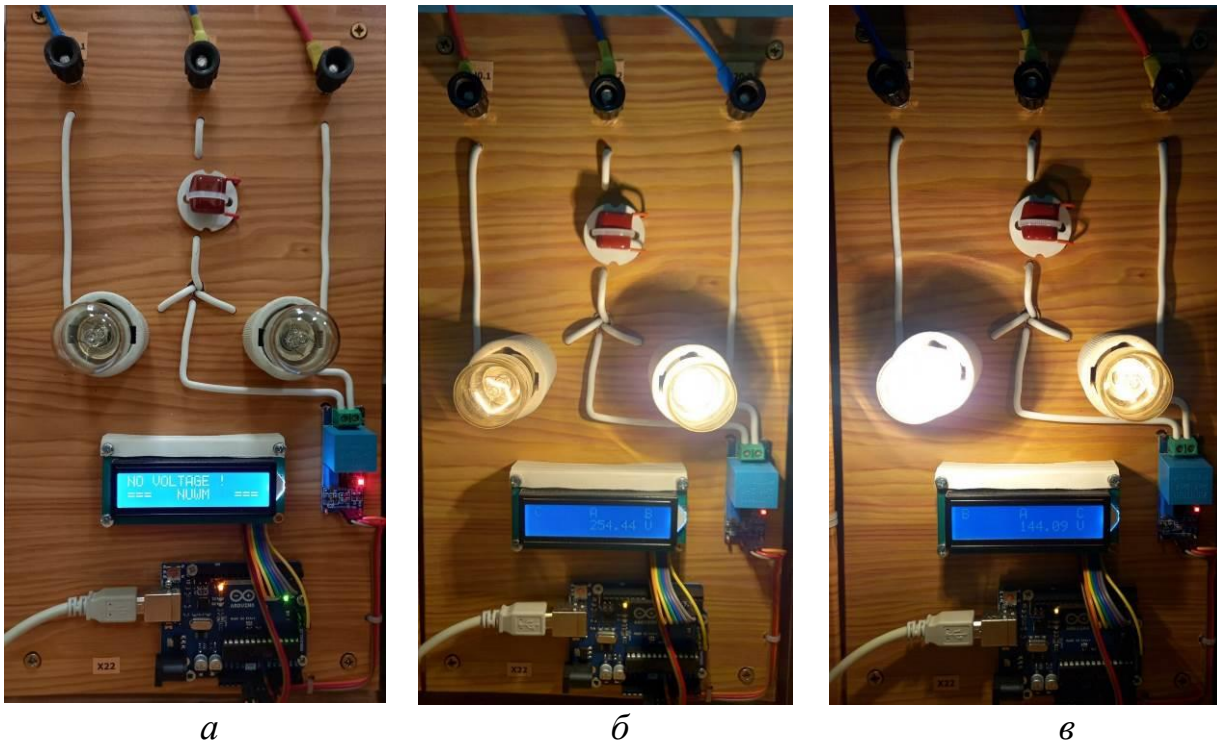


Рисунок 2.8 – Функціонування пристрою для визначення порядку чергування фаз: *а* – напруга на пристрій не подана; *б* – порядок фаз С-А-В; *в* – порядок фаз В-А-С

Бібліотека **Filters.h** для оброблення аналогових сигналів

В програмі передбачається використовувати бібліотеку **Filters.h** для оцінювання діючого значення напруги, що вимірюється давачем ZMPT101В.

Для використання підключення бібліотеки для програми використовується директива:

```
#include <Filters.h> //підключення бібліотеки роботи з давачем напруги
```

Необхідно оголосити наступні змінні:

```
float testFrequency = 50; // частота напруги, Гц
float windowLength = 40.0/testFrequency; // рахувати діюче значення напруги
// на основі 40 періодів
int Sensor = 0; // номер аналогового входу, до
// якого підключено давач напруги
float x; // результат перетворення
float U; // фактичне значення напруги
float b= -3.88; // коефіцієнти функції корекції
float k = 1.05; // U=k*x+b
```

При цьому калібрування давача здійснюється з використанням лінійної функції корекції. Якщо позначити результат перетворення x , а фактичне значення напруги U , то функція корекції має вигляд:

$$U = k \cdot x + b.$$

Для оновлення показів кожну секунду необхідно оголосити наступні змінні:

```
unsigned long printPeriod = 1000; //час оновлення показів, мс  
unsigned long previousMillis = 0; //попередній час зняття показів
```

В основному циклі loop програми для отримання діючих значень напруги кожну секунду необхідно виконати наступні команди:

```
void loop()  
{  
  RunningStatistics inputStats;  
  inputStats.setWindowSecs( windowLength );  
while(true)  
  {  
    Sensor = analogRead(A0); // зчитування сигналу  
    inputStats.input(Sensor);  
    if((unsigned long)(millis() - previousMillis) >= printPeriod) {  
      previousMillis = millis(); // оновлювати значення кожну секунду  
        //Обрахунок фактичного значення напруги  
      x=inputStats.sigma();  
      U = b + k*x;  
  
      // аналіз фактичного значення напруги, що зберігає U  
  
    } //while  
  } //loop
```

Бібліотека LiquidCrystal.h для роботи з рідкокристалічним текстовим дисплеєм

Дана бібліотека за замовчуванням доступна в ArduinoIDE. Для її використання в програмі можна скористатися директивою:

```
#include <LiquidCrystal.h>
```

Детально з функціями бібліотеки можна ознайомитися за посиланням:

<https://www.arduino.cc/reference/en/libraries/liquidcrystal/>

Модель пристрою у Proteus

Для відлагодження програмного забезпечення можна скористатися фізичним пристроєм, що знаходиться в ауд. 509 (рис. 2.8), або використати модель пристрою у симуляторі Proteus, рис. 2.9. Модель включає трифазне джерело V1, до якого підключено ламповий вказівник послідовності чергування фаз HL1, C, HL2. Для вимірювання напруги на лампах використовуються вольтметри. Кнопка S1 забезпечує подачу напруги на схему лампового вказівника. Кнопка S2 забезпечує зміну фаз, що підключені до затискачів X1, X2 вказівника. До складу моделі входить підсистема, що моделює давач напруги ZMPT101B.

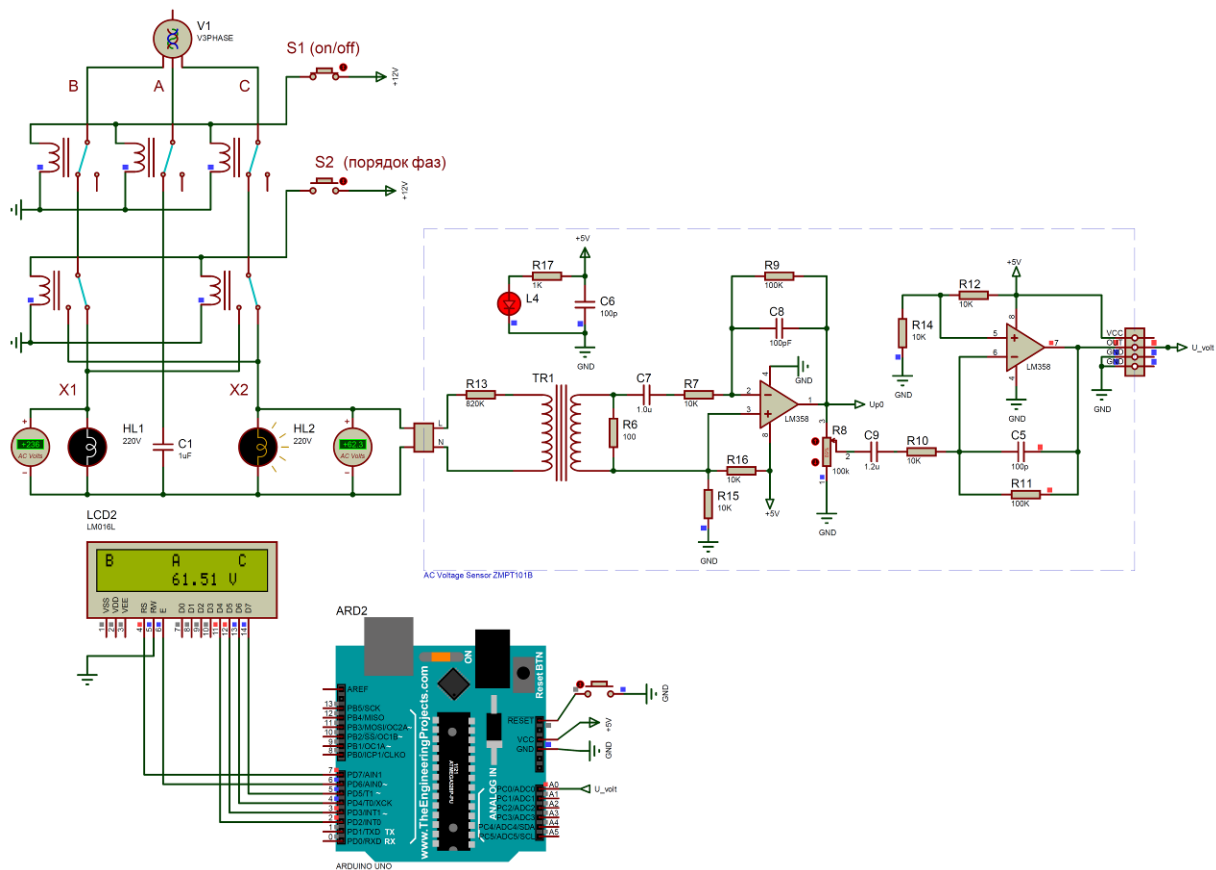


Рисунок 2.9 – Модель пристрою визначення порядку чергування фаз у Proteus

ЗАВДАННЯ

Написати програму для плати ArduinoUNO у складі пристрою визначення порядку чергування фаз. Пристрій має визначати та виводити на дисплей назви фаз В і С при фіксованому підключенні фази А відповідно. Також на дисплей має виводитися поточне діюче значення напруги на одній з ламп лампового вказівника. Інформація на дисплеї має оновлюватися кожну секунду. В разі відсутності напруги на вказівнику, на дисплей має виводитися відповідне повідомлення та прізвище виконавця (наприклад: NO VOLTAGE; Kovalenko).

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Завантажити бібліотеку Filters.h <https://cutt.ly/vBrtxxw>

Вказану бібліотеку необхідно розпакувати і папки з бібліотеками розмістити в каталозі: c:\Program Files (x86)\Arduino\libraries\.

2. Скласти блок-схему алгоритму роботи пристрою визначення порядку чергування фаз.

3. Написати програму для плати ArduinoUNO в середовищі ArduinoIDE відповідно до розробленого алгоритму.

4. Для компіляції програми в середовищі Arduino виконати команду меню Скетч–Експорт скомпільованого бінарника. В разі наявності помилок –

виправити їх. В папці проекту Arduino з'явиться файл ...ino.standard.hex з кодами скомпільованої програми.

5. В Proteus-моделі мікроконтролера вказати шлях до отриманого файлу з кодами скомпільованої програми. Запустити модель та виконати тестування працездатності пристрою.

Вміст звіту з лабораторної роботи

1. Тема, мета роботи.
2. Блок-схема алгоритму роботи пристрою визначення порядку чергування фаз.
3. Текст програми.
4. Вигляд працездатної моделі у Proteus під час моделювання (або фото пристрою): при відсутності напруги; при порядку фаз В, А, С; при порядку фаз С, А, В.
5. Висновки.

Контрольні запитання

1. В чому різниця між миттєвими і діючим значенням змінної напруги живлення?
2. Що таке амплітуда змінної напруги і частота?
3. Як можна визначити порядок чергування фаз трифазної системи?
4. Чому дорівнює ємнісний опір конденсатора?
5. Чому у ламповому вказівнику порядку чергування фаз лампа у фазі В горить яскравіше, ніж у фазі С?
6. Яким чином вимірювалося діюче значення напруги в лабораторній роботі?
7. Яку функцію виконує трансформатор у складі давача напруги типу ZMPT101B?
8. Яку функцію виконують операційні підсилювачі у складі давача напруги типу ZMPT101B?
9. Яким вимогам має відповідати вхідний сигнал АЦП у складі мікроконтролера ATmega328p?
10. Яке призначення аналогового мультиплексора у складі модуля АЦП, що входить до складу мікроконтролера?
11. Поясніть алгоритм функціонування пристрою за складеною Вами блок-схемою.
12. Поясніть призначення команд складеної Вами програми для пристрою.

ЛАБОРАТОРНА РОБОТА №3

Програмування мікропроцесорного лічильника для технічного обліку електроенергії, що споживається освітленням цеху

Мета: навчитися програмно реалізовувати вимірювання електричної енергії.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Загальні відомості щодо освітлення промислових приміщень

Для гарантування безпеки праці та її продуктивності необхідно забезпечити необхідний рівень освітлення робочого місця працівника, а також маршрутів переміщення людей по території підприємства, рис. 3.1. При цьому на кожному підприємстві є власні особливості та специфічні вимоги до освітлення, що визначаються особливостями виробничого процесу. Вимоги до рівня освітленості робочого місця визначаються ДБН В.2.5-28:2018 «Природне і штучне освітлення».

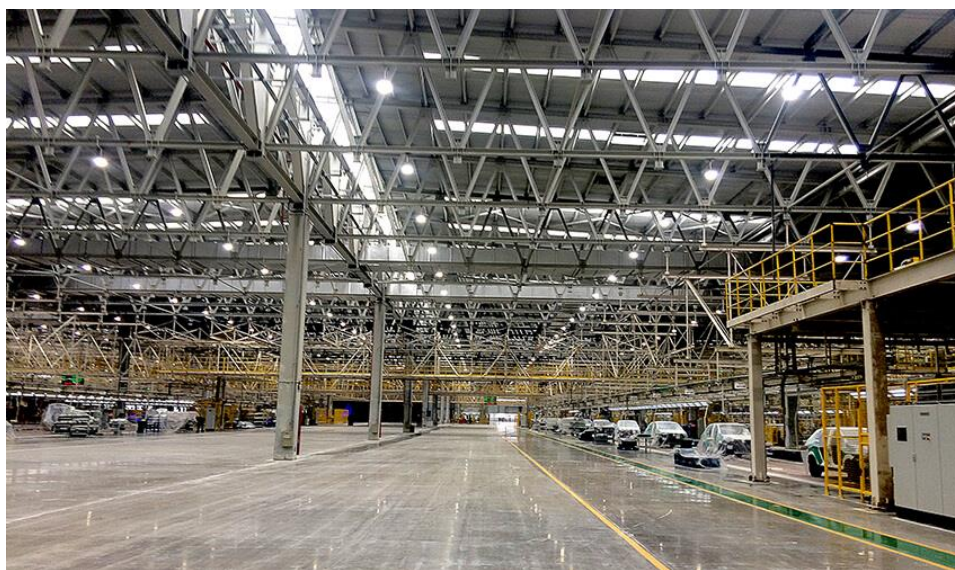


Рисунок 3.1 – Розташування освітлювальних приладів в цеху промислового підприємства

Відповідно до ПУЕ, живлення мереж освітлення має бути виконано від мереж напругою 220/380 В з системою заземлення TN-S або TN-C-S. В якості джерела живлення освітлювальної мережі цеха підприємства можна прийняти шини 0,4 кВ цехової трансформаторної підстанції.

Робоче та аварійне освітлення живляться від незалежних джерел живлення. Схема освітлення, відповідно до ПУЕ, може включати три ступені розподілу електроенергії, відповідно у схемі наявні три низьковольтні пристрої розподілу: РП 0,4 кВ підстанції, щит освітлення, групові щитки освітлення. Мережа від РП 0,4 кВ до щита освітлення називається живлячою, від щита

освітлення до групових щитків – розподільчою, від групових щитків до світильників – груповою.

Для живлення освітлювальних приладів внутрішнього та зовнішнього освітлення, як правило, має застосовуватися напруга не вище 220 В. Напруга 380 В може застосовуватися як виключення (п. 6.1.13 ПУЕ).

Групові мережі освітлення виконуються або однофазними трипровідними або трифазними п'ятипровідними. Освітлювальні прилади намагаються рівномірно розподіляти між фазами (рис. 3.2).

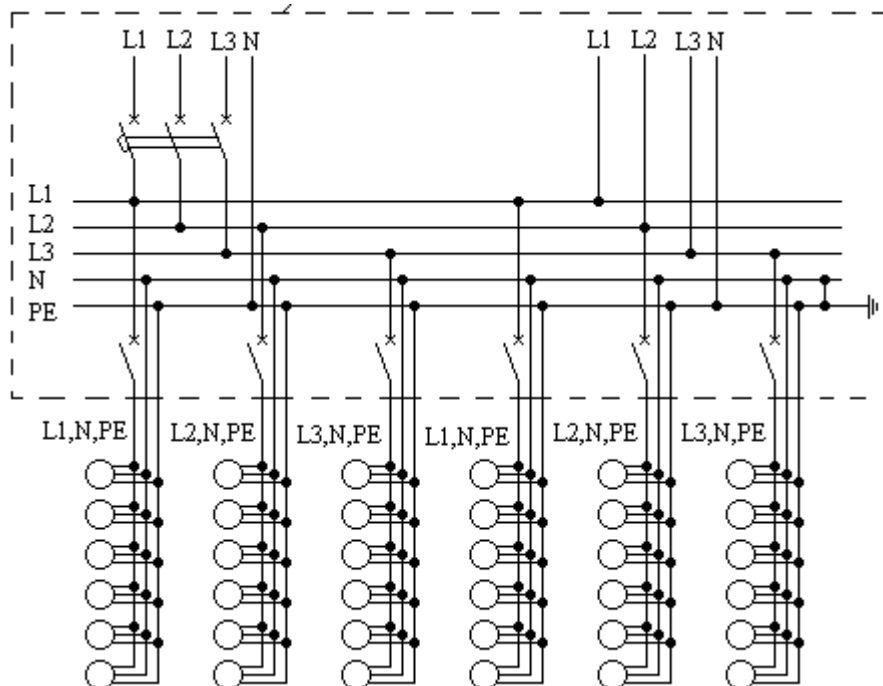


Рисунок 3.2 – Принципова схема групової мережі освітлення для цеха

Серійні пристрої технічного обліку

Технічний (контрольний) облік електроенергії - облік для контролю витрат електроенергії на електростанції, підстанції, підприємстві, а також для обчислення і аналізу втрат електроенергії в електричних мережах всіх класів напруги. Лічильники, що встановлюються для технічного обліку, мають назву лічильників технічного обліку.

Функції лічильників технічного обліку в більшості випадків виконують реєстратори електричних параметрів або аналізатори параметрів мережі. Наприклад, ТОВ «НОВАТЕК-ЕЛЕКТРО» (м. Одеса) випускає реєстратор електричних параметрів мікропроцесорний РПМ-416, аналізатор параметрів електромережі Digis A40 виробництва французької компанії Socomes (рис. 3.3).



а



б

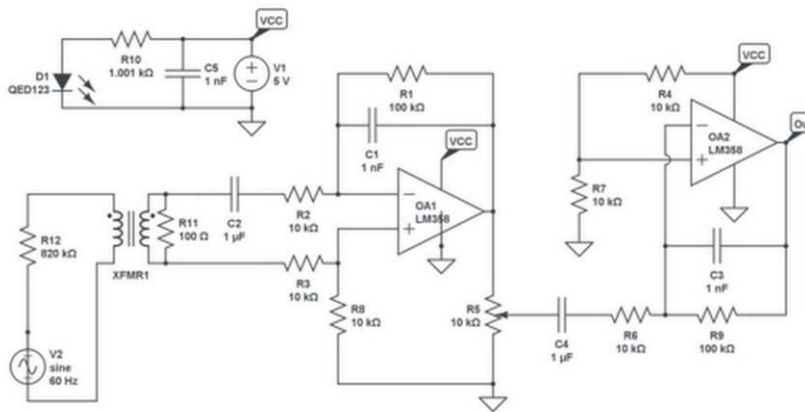
Рисунок 3.3 – Мікропроцесорні пристрої, що, серед інших, виконують функції технічного обліку електроенергії: *а* – РПМ-416; *б* – аналізатор Diris A40

Бібліотека ZMPT101B.h для роботи з датчиком напруги ZMPT101B

Для вимірювання фазних напруг електромережі передбачається використовувати трансформаторний датчик типу ZMPT101B, рис. 3.4.



а



б

Рисунок 3.4 – Трансформаторний датчик напруги ZMPT101B: *а* – загальний вигляд; *б* – принципова схема

Для роботи з датчиком можна використовувати бібліотеку ZMPT101B.h, для оголошення якої використовується директива:

```
#include < ZMPT101B.h >
```

Для створення об'єктної змінної датчика, що підключений до плати Arduino, використовується команда:

```
ZMPT101B voltageSensorA(A0);
```

де ZMPT101B – назва бібліотеки;

voltageSensorA – об'єктна змінна, що відповідає датчеві;

A0 – аналоговий вхід, до якого підключено датч.

Для визначення діючого значення напруги (на частоті 50 Гц) використовується метод:

```
Ua = voltageSensorA.getVoltageAC();
```

Бібліотека ACS712.h для роботи з датчем струму ACS712

Для вимірювання струму використовується датч ACS712, що працює з використанням ефекту Холла. Такі датчі випускаються на номінальні первинні струми 5А, 20А, 30А, рис. 3.5.

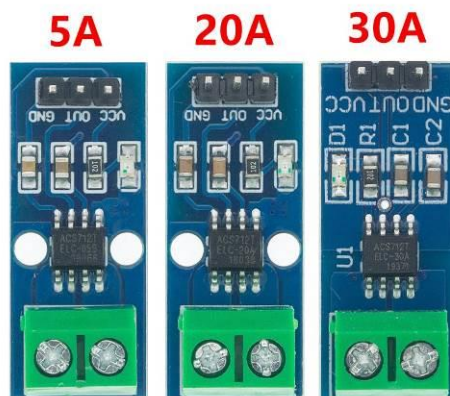


Рисунок 5 – Датчі струму ACS712

Для роботи з датчами використовується бібліотека ACS712.h:

```
#include <ACS712.h>
```

Для створення об'єктної змінної датча, що розрахований на номінальний струм 20А та підключений до плати Arduino, використовується команда:

```
ACS712 currentSensorA(ACS712_20A, A3);
```

Для визначення діючого значення струму (на частоті 50 Гц) використовується метод:

```
Ia = currentSensorA.getCurrentAC();
```

Модель лічильника у Proteus

Для виконання лабораторної роботи використовується модель мікропроцесорного лічильника для технічного обліку електроенергії в симуляторі Proteus, рис. 3.6. Модель включає трифазне джерело живлення V1-V3 лінійною напругою 380В, до якого підключено активні опори RA, RB, RC. Також пристрій включає три датча фазних напруг, три датча струму, мікроконтролерну плату ArduinoUNO та графічний дисплей ST7735R.

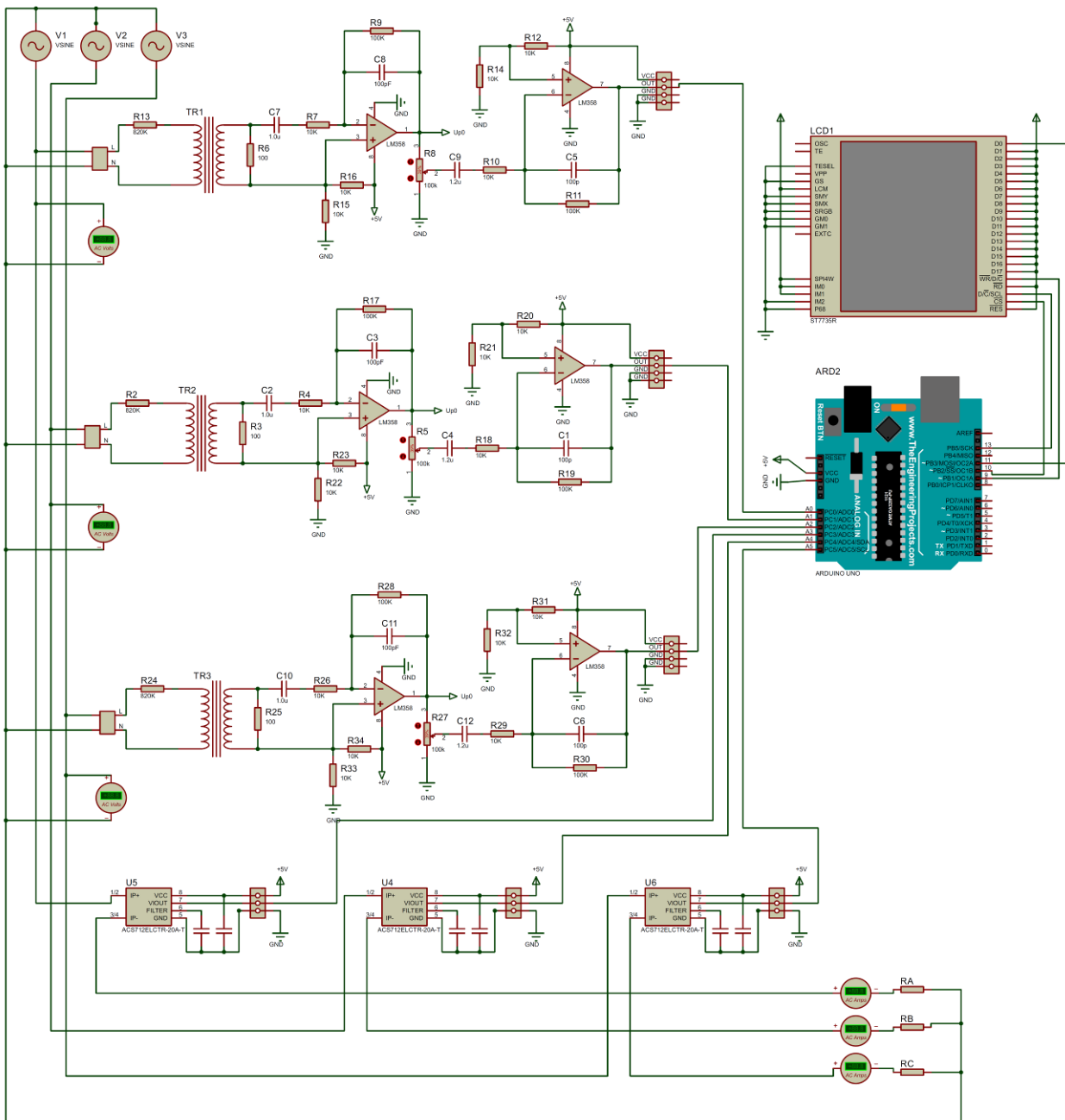


Рисунок 3.6 - Модель мікропроцесорного лічильника для технічного обліку електроенергії в симуляторі Proteus

ЗАВДАННЯ

Розробити програмне забезпечення для мікропроцесорного лічильника технічного обліку електроенергії, модель якого представлена на рис. 3.6. На дисплей мають виводитися дані відповідно до рис. 3.7.

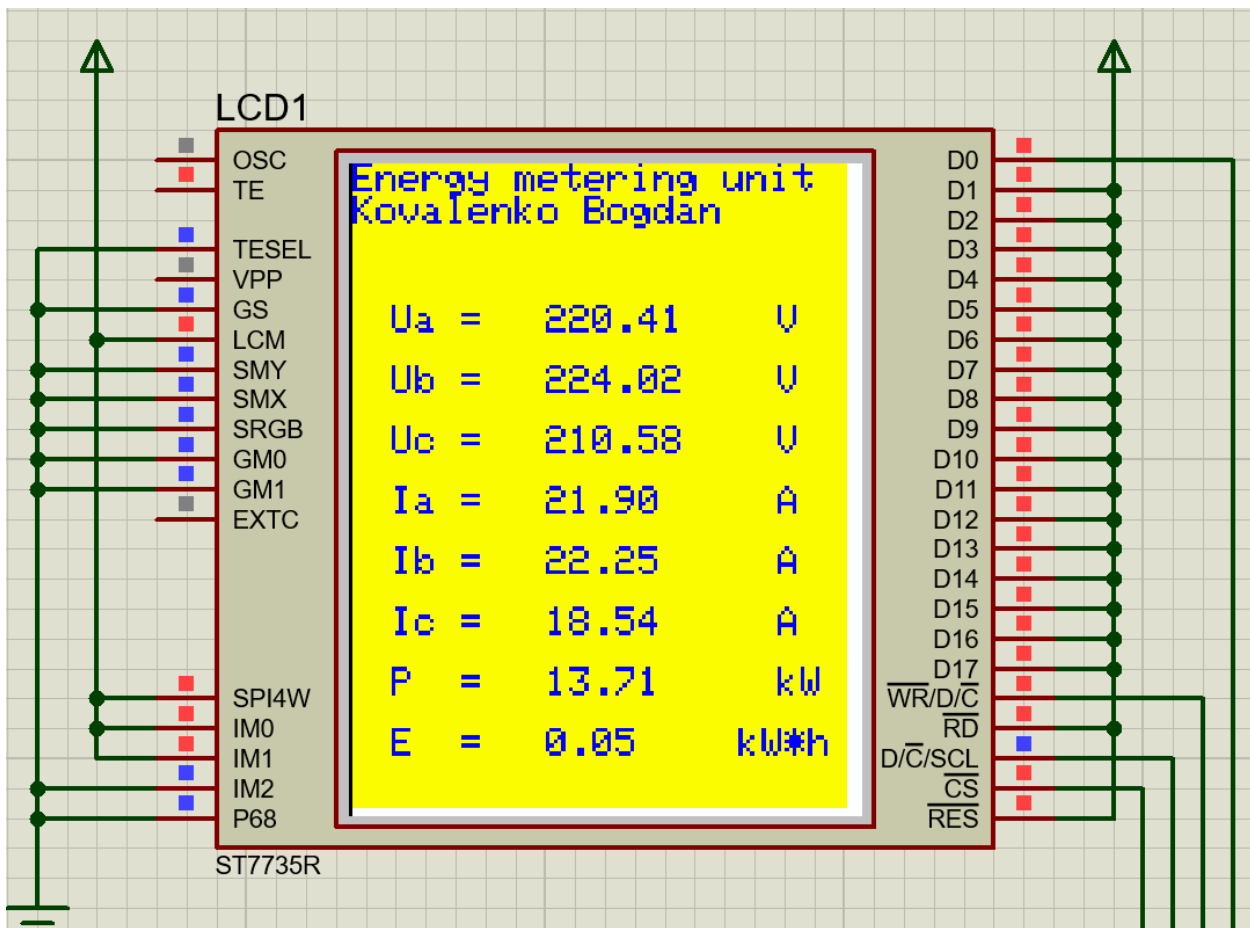


Рисунок 3.7 – Відображення даних про стан навантаження на дисплеї лічильника

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Завантажити бібліотеки:

ZMPT101B.h <https://cutt.ly/PBoDyan>

ACS712.h <https://cutt.ly/JBoDaCI>

Вказані бібліотеки необхідно розпакувати і папки з бібліотеками розмістити в каталозі: c:\Program Files (x86)\Arduino\libraries\ .

2. Скласти блок-схему алгоритму роботи лічильника.

3. Написати програму для плати ArduinoUNO в середовищі ArduinoIDE відповідно до розробленого алгоритму.

4. Для компіляції програми в середовищі Arduino виконати команду меню Скетч–Експорт скомпільованого бінарника. В разі наявності помилок – виправити їх. В папці проекту Arduino з'явиться файл ...ino.standard.hex з кодами скомпільованої програми.

5. В Proteus-моделі мікроконтролера вказати шлях до отриманого файлу з кодами скомпільованої програми. Запустити модель та виконати тестування працездатності пристрою.

Вміст звіту з лабораторної роботи

1. Тема, мета роботи.
2. Блок-схема алгоритму роботи лічильника для технічного обліку електроенергії.
3. Текст програми.
4. Вигляд працездатної моделі у Proteus під час моделювання.
5. Висновки.

Контрольні запитання

1. Який основний недолік розробленого пристрою? Яким чином цей недолік можна виправити?
2. Як розраховується потужність трифазного активного навантаження за фазними струмами та напругами?
3. Поясніть, яким чином у Вашій програмі розраховувалася активна енергія?
4. Поясніть принцип дії давача струму типу ACS712.
5. У чому полягає ефект Холла?
6. Поясніть принцип дії давача напруги ZMPT101B.
7. Маючи миттєві значення напруги за певний період часу, яким чином можна обчислити діюче значення?
8. В яких одиницях вимірюється активна, реактивна та повна потужність?
9. В яких одиницях вимірюється активна енергія?
10. Яким чином перевести величину, що вимірюється у «Вт·с», у «кВт·год»?

ЛАБОРАТОРНА РОБОТА №4

Програмування мікропроцесорного струмового захисту силового приєднання з розпізнаванням аварійного режиму

Мета: оволодіти навичками написання програми для аналізу діючих значень струмів силового приєднання.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Розглянемо мікропроцесорний струмовий захист, структурна схема якого зображена на рис. 4.1. Захист забезпечує захист силового приєднання, опори фаз якого становлять $Z1$, $Z2$, $Z3$. Живлення на приєднання подається комутаційним апаратом $Q1$. Вимірювання струмів фаз забезпечується давачами струму $TA1$ – $TA3$. Перетворення вторинних струмів вказаних давачів у сигнали за напругою з параметрами, що задовольняють вимогам АЦП, здійснюють перетворювачі сигналів $ПС1$ – $ПС3$. Аналого-цифрове перетворення одержаних сигналів за напругою передбачається здійснювати за допомогою АЦП, що вбудовано у мікроконтролер $МК$. Поточна інформація про функціонування силового приєднання відображається на дисплеї TFT . Сигнал на відключення комутаційного апарата $Q1$ попередньо підсилюється перетворювачем сигналу $ПС4$.

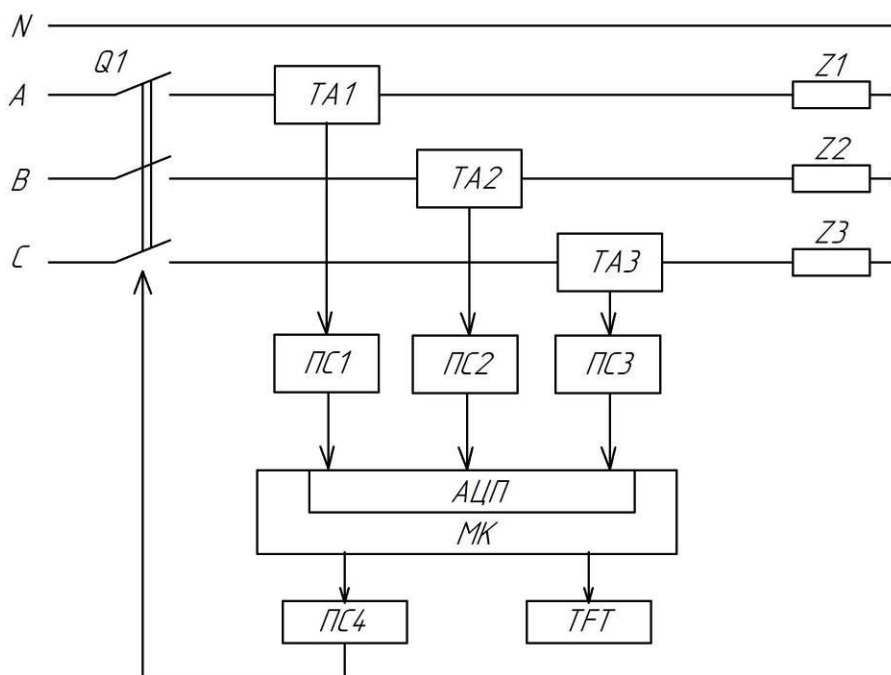


Рисунок 4.1 – Структурна схема мікропроцесорного струмового захисту

Для спрощення виявлення поточного стану силового приєднання пропонується класифікувати стан кожної фази за допомогою цифрової ознаки x_j , де $j = \{A, B, C\}$ – позначення фази. Передбачається, що ознака приймає значення з множини $\{1, 2, 3\}$. Для встановлення поточного значення ознаки

пропонується порівнювати вимірне діюче значення струму фази i_j з нижньою I_L та верхньою I_H уставками. Нижня уставка I_L відповідає обриву фази, верхня I_H – виникненню замикання. Тобто цифрова ознака визначається відповідно до залежності:

$$x_j = \begin{cases} 1, & \text{якщо } i_j < I_L \\ 2, & \text{якщо } I_L \leq i_j \leq I_H \\ 3, & \text{якщо } i_j > I_H \end{cases} \quad (4.1)$$

Поточний стан силового приєднання характеризується інтегральною цифровою ознакою X :

$$X = 100 \cdot x_C + 10 \cdot x_B + x_A \quad (4.2)$$

Можливі типи аварійних станів силового приєднання та відповідні значення цифрових ознак наведені у табл. 4.1.

Таблиця 4.1

Відповідність між типами аварійних станів силового приєднання та значенням цифрових ознак

Тип замикання	x_C	x_B	x_A	X
Однофазне на землю фази А	2	2	3	223
Однофазне на землю фази В	2	3	2	232
Однофазне на землю фази С	3	2	2	322
Обрив фази А	2	2	1	221
Обрив фази В	2	1	2	212
Обрив фази С	1	2	2	122
Двофазне к.з. між фазами АВ	2	3	3	333
Двофазне к.з. між фазами ВС	3	3	2	332
Двофазне к.з. між фазами СА	3	2	3	323
Трифазне к.з. АВС	3	3	3	333

Для обчислення величини цифрової ознаки фази за залежністю (4.1) можна скористатися блок-схемою на рис. 4.2. Такий алгоритм може бути оформлений окремою підпрограмою.

На рис. 4.3 наведено блок-схему алгоритму функціонування мікропроцесорного пристрою релейного захисту. Після скидання мікроконтролера виконується ініціалізація дисплея (блок 2), а також задаються нижня I_L та верхня I_H уставки (блок 3). За допомогою АЦП визначаються поточні діючі значення струмів фаз, блок 4. За допомогою підпрограми (рис. 4.2) обчислюються значення цифрових ознак x_A , x_B , x_C , блоки 5-7. У блоці 8, згідно з (4.2), обраховується значення інтегральної ознаки. Відповідно до табл. 4.1, блоки 9, 11, 13, 15, 17, 19, 21, 23, 25, 27 забезпечують виявлення типу аварійного стану. Якщо аварійний стан виявлено, блоки 10, 12, 14, 16, 18, 20, 22, 24, 26, 28 забезпечують виведення на дисплей відповідно повідомлення.

Команда у блоці 30 забезпечує відключення силового вимикача приєднання (блок 30). В разі відсутності аварійних станів, видається повідомлення про нормальний режим роботи (блок 29).

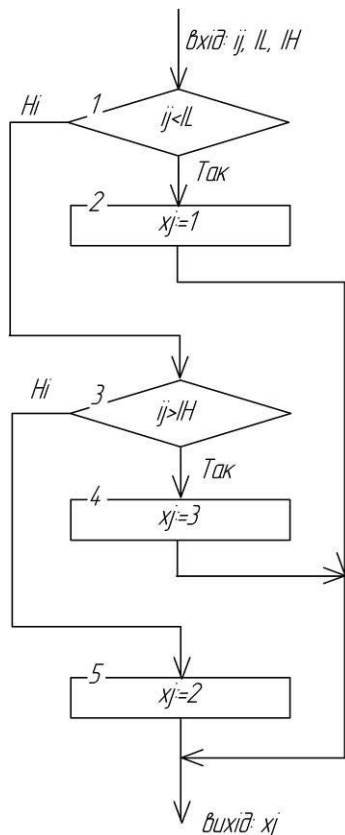


Рисунок 4.2 – Блок-схема підпрограми визначення цифрової ознаки стану j -ї фази

Для виконання лабораторної роботи використовується модель мікропроцесорного струмового захисту силового приєднання в симуляторі Proteus, рис. 4.4. Модель включає трифазне джерело живлення V3PHASE лінійною напругою 380В, до якого через комутаційний апарат, що моделюється реле K1–K3, підключено активно-індуктивні опори R11, L11, R21, L21, R31, L31. Вимірювальні трансформатори струму, для спрощення моделі, не враховано. Пристрій включає три перетворювача струму ASC712, мікроконтролерну плату ArduinoUNO та графічний дисплей ST7735R. Для імітації замикання фаз на землю використовуються кнопки S_A, S_B, S_C. Обрив фаз імітується натисканням кнопок S_LA, S_LB, S_LC, а двофазні короткі замикання – S_AB, S_BC, S_CA. Відображення повідомлень на дисплеї в різних режимах ілюструє рис. 4.5.

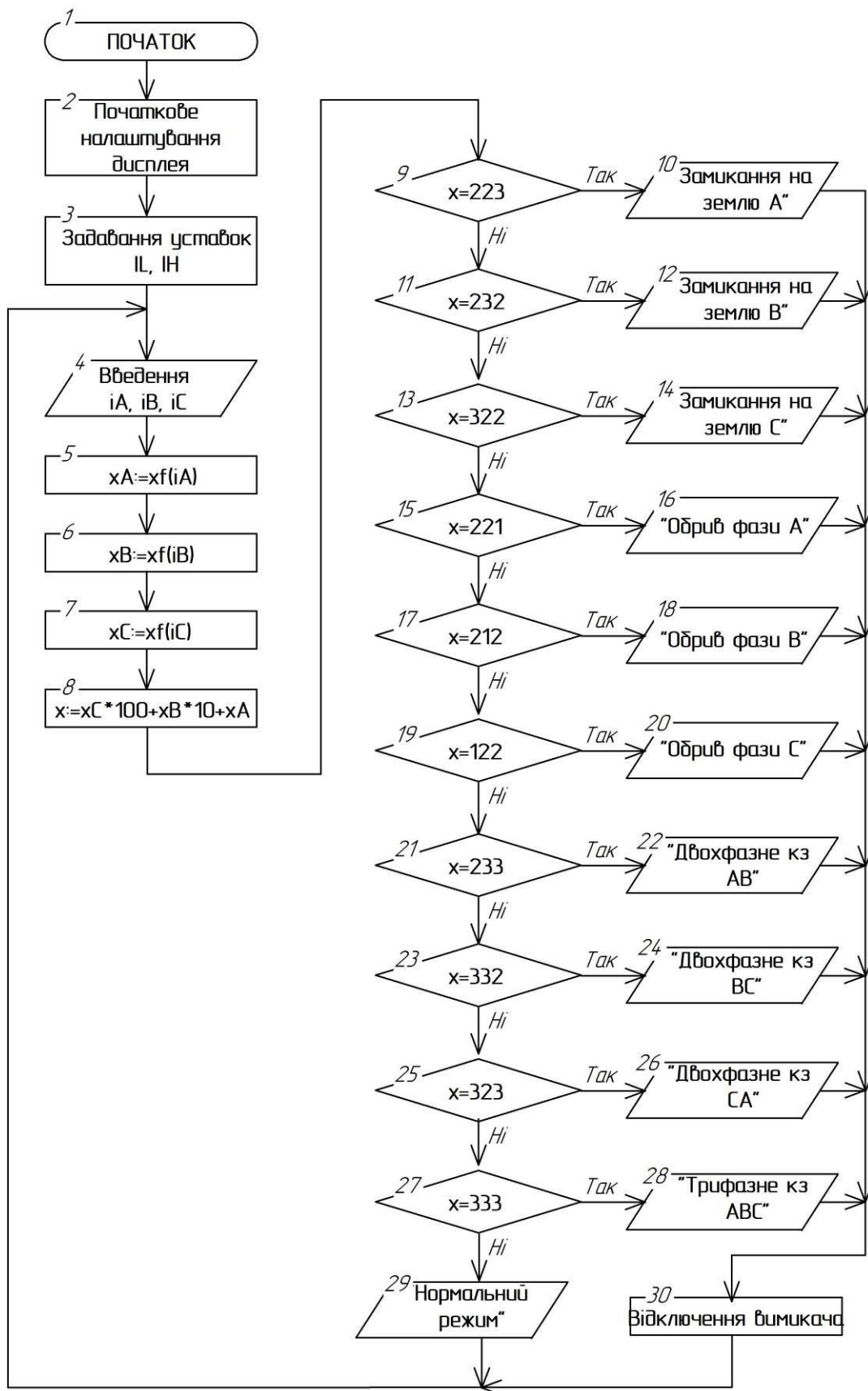


Рисунок 4.3 – Блок-схема алгоритму функціонування мікропроцесорного пристрою релейного захисту

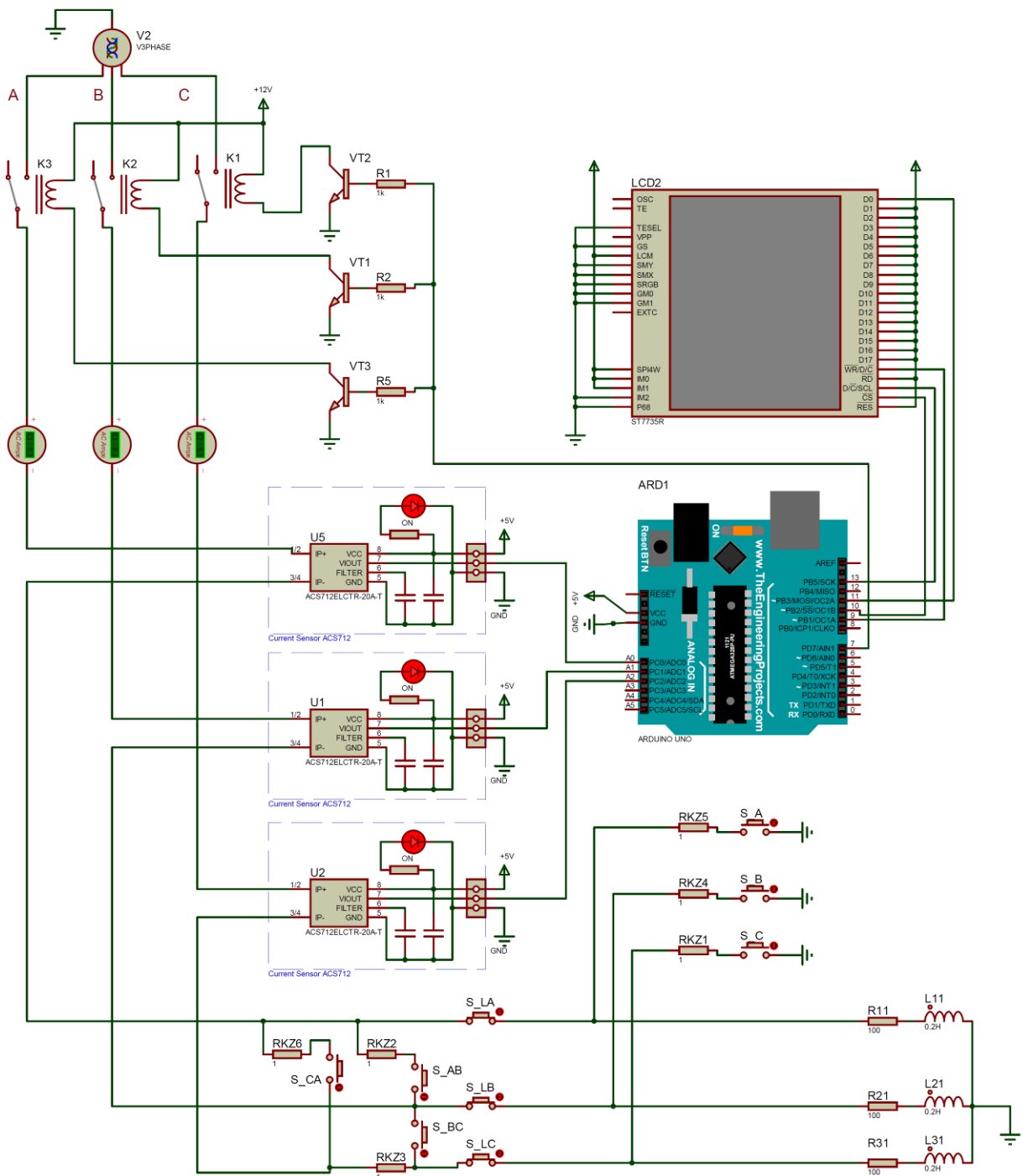
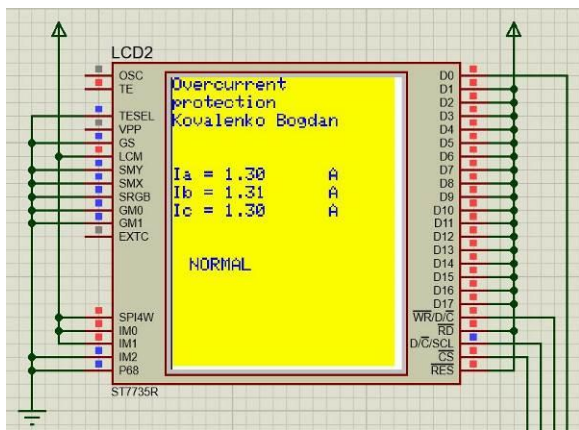
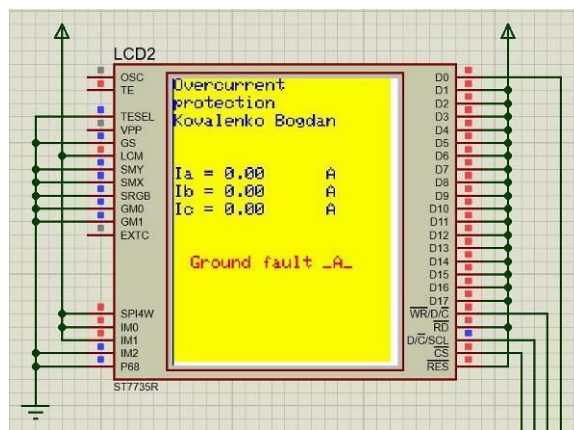


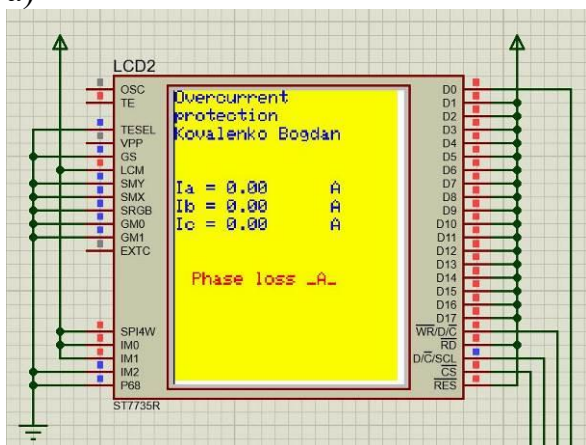
Рисунок 4.4 – Модель мікропроцесорного струмового захисту силового приєднання в симуляторі Proteus



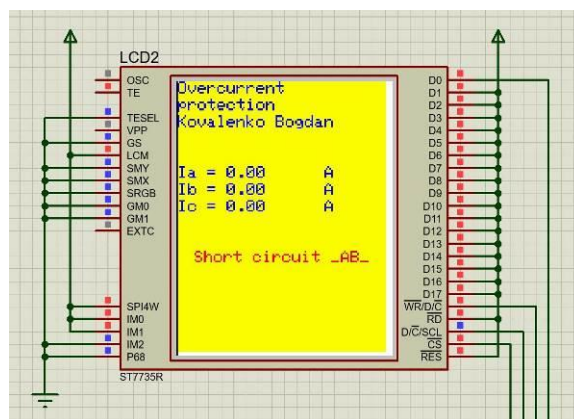
а)



б)



в)



г)

Рисунок 4.5 – Відображення повідомлень на дисплеї в різних режимах:
 а – нормальний режим; б – замикання на землю фази А; в – обрив фази А;
 г – коротке замикання між фазами А і В

ЗАВДАННЯ

Написати програму для мікропроцесорного струмового захисту силового приєднання (модель представлена на рис. 4.4), що дозволяє розпізнавати наступні аварійні режими: однофазне замикання на землю; обрив фази; двофазне коротке замикання; трифазне коротке замикання.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Написати програму для плати ArduinoUNO в середовищі ArduinoIDE відповідно до розробленого алгоритму (рис. 4.3).

2. Для компіляції програми в середовищі Arduino виконати команду меню Скетч–Експорт скомпільованого бінарника. В разі наявності помилок – виправити їх. В папці проекту Arduino з'явиться файл ...ino.standard.hex з кодами скомпільованої програми.

3. В Proteus-моделі мікроконтролера вказати шлях до отриманого файлу з кодами скопійованої програми. Запустити модель та виконати тестування працездатності пристрою.

Вміст звіту з лабораторної роботи

1. Тема, мета роботи.
2. Блок-схема алгоритму роботи мікропроцесорного струмового захисту.
3. Текст програми.
4. Вигляд працездатної моделі у Proteus під час моделювання, вигляд повідомлень на дисплеї в різних аварійних режимах.
5. Висновки.

Контрольні запитання

1. Поясніть функціонування непошкоджених фаз в трифазній чотирипровідній системі при обриві однієї фази.
2. Які ознаки двофазного к.з. в трифазній чотирипровідній системі.
3. Дайте визначення поняттю «фаза».
4. Чому для передачі електроенергії обрано систему з трьома фазами? Які її переваги перед системою з 2, 4, 5, ... фазами?
5. Дайте визначення струмовому захисту.
6. Які види струмового захисту Вам відомі?
7. Які особливості функціонування максимального струмового захисту?
8. Які особливості функціонування струмової відсічки?
9. Які недоліки притаманні мікропроцесорним струмовим захистам, що оперують з діючими значеннями струмів?
10. Як можна підвищити точність функціонування розробленого в роботі захисту?
11. Які недоліки розробленого захисту Ви можете назвати?

ЛАБОРАТОРНА РОБОТА №5

Програмування мікропроцесорного реле мінімального опору для дистанційного релейного захисту

Мета: оволодіти навичками написання програми для перевірки умови спрацювання реле мінімального з коловою характеристикою

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Принцип дії дистанційного релейного захисту

Дистанційний захист застосовується для захисту високовольтних ліній електропередачі від коротких замикань. Це захист з відносною селективністю. Принцип його дії базується на контролі імпедансу лінії. В нормальному режимі роботи ЛЕП її струм та напруга на лінії мають певні усталені значення, позначимо їх $\dot{I}_{норм}$, $\dot{U}_{норм}$. Тоді імпеданс на затискачах захисту, що встановлений на початку ЛЕП, становитиме:

$$\underline{Z} = \frac{\dot{U}_{норм}}{\dot{I}_{норм}}. \quad (5.1)$$

В разі виникнення короткого замикання напруга на шинах зменшується, а струм суттєво збільшується, оскільки опір, що контролюється захистом, зменшується:

$$\underline{Z}_{кз} = \frac{\dot{U}_{кз}}{\dot{I}_{кз}} < \underline{Z}_н. \quad (5.2)$$

При цьому імпеданс $\underline{Z}_{кз}$ лінії від місця встановлення захисту до точки к.з. пропорційний питомому імпедансу \underline{Z}_0 ЛЕП:

$$\underline{Z}_{кз} = \underline{Z}_0 \cdot l_{кз}, \quad (5.3)$$

де $l_{кз}$ - відстань від місця встановлення захисту до точки к.з.

Тобто шляхом контролю імпедансу можливо виявити факт наявності короткого замикання та визначити віддаленість пошкодження.

Дистанційний захист, як правило, виконують триступеневим. Кожний ступінь має різний час спрацювання і призначений для захисту від к.з. різних віддаленостей. Перший ступінь, як правило, охоплює близько 85% довжини лінії, яка захищається. Перший ступінь виконують без витримки часу, тобто при замикання в зоні його дії час t^1 відключення мережі визначається тільки власним часом спрацювання захисту та часом відключення високовольтного вимикача, рис. 5.1.

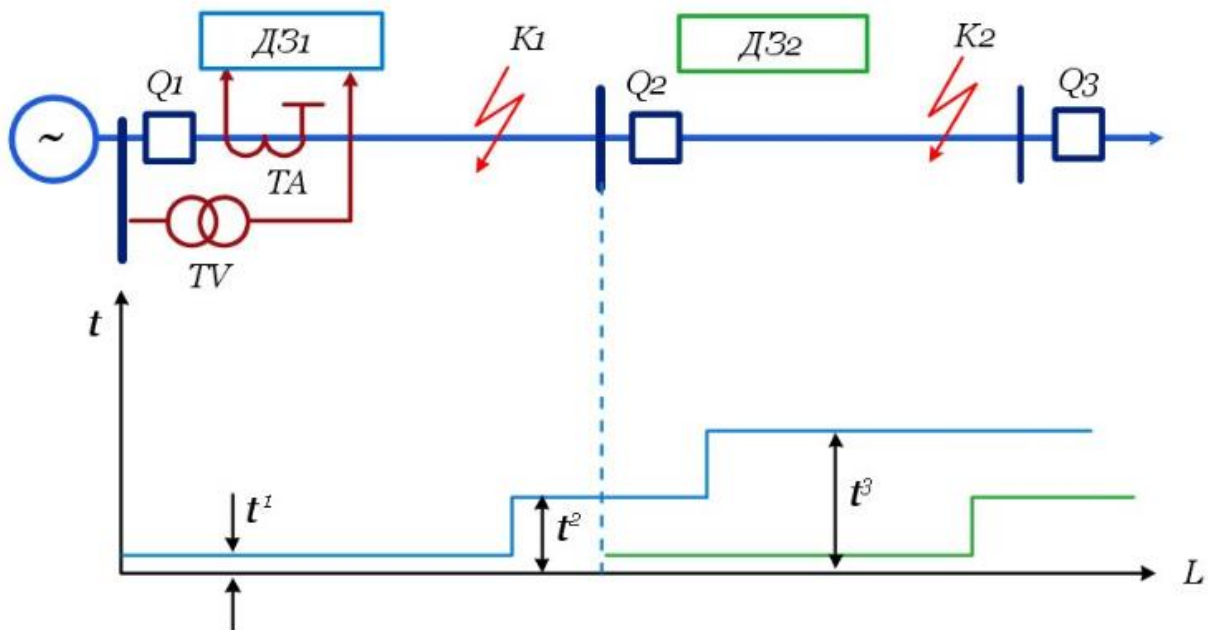


Рисунок 5.1 – Принцип дії дистанційного захисту

Другий ступінь дистанційного захисту забезпечує охоплення всієї першої лінії та початку наступної. Час спрацювання другого ступеня становить $t^2=0,4-0,5$ с. Третій степінь має ще більший час спрацювання, оскільки виконує функції ближнього та дальнього резервування.

У складі дистанційного захисту для вимірювання імпедансу використовуються реле мінімального опору. У перших захистах використовувалися реле на індукційному принципі, потім застосовувалися напівпровідникові реле. Зараз використовуються мікропроцесорні реле, в яких алгоритм визначення імпедансу електромережі та порівняння його з характеристикою спрацювання виконується програмним чином.

Оскільки імпеданс, який вимірюється, має активну та реактивну складові $Z = R + jX$, то уставка спрацювання реле мінімального опору представляється не одним числом, а характеристикою на комплексній площині. Характеристики бувають різних форм, для задавання більшості з них використовується модуль імпедансу спрацювання $|Z_{сп}|$ та кут максимальної чутливості реле $\varphi_{мч}$. При цьому необхідно забезпечити надійне відведення характеристик вимірювальних органів дистанційних захистів від навантажувальних режимів з урахуванням впливу дуги.

Розглянемо основні види характеристик спрацювання реле опору. Колова характеристика з центром у початку координат, рис. 2, являє собою зону комплексної площини, що обмежена колом. Якщо фактичний імпеданс попадає всередину характеристики, реле мінімального опору спрацює. Реле, що мають такі характеристики, називають реле повного опору, оскільки спрацювання цих реле не залежить від кута φ_p між струмом та напругою електромережі.

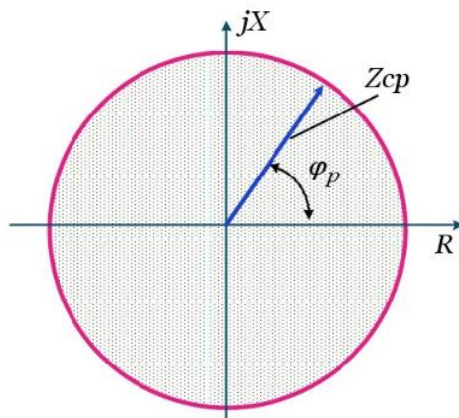


Рисунок 5.2 – Колова характеристика з центром у початку координат спрацювання реле мінімального опору

Також широко використовується колова характеристика, що проходить через початок координат, рис. 5.3. Реле, що має таку характеристику, є спрямованим, оскільки спрацьовує тільки при напрямку потужності від шин до лінії. При протіканні потужності в зворотному напрямку реле не спрацьовує, оскільки характеристика не охоплює третій квадрант. Точка 0 відповідає нульовому опору на початку лінії, що захищається. Кут φ , при якому модуль імпедансу спрацювання досягає максимуму, є кутом максимальної чутливості.

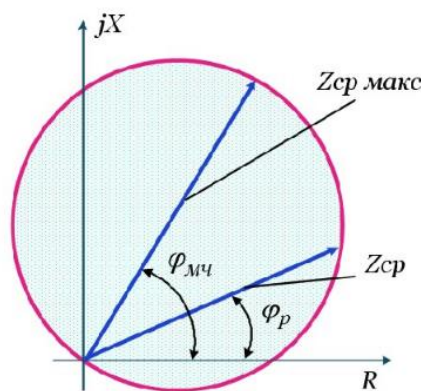


Рисунок 5.3 – Колова характеристика, що проходить через початок координат, спрацювання реле мінімального опору

Для відведення від робочих режимів та підвищення чутливості дистанційного захисту для третього ступеня використовуються еліптичні характеристики, рис. 5.4.

Застосування мікропроцесорних засобів для реалізацій функцій дистанційного захисту дає змогу реалізувати багатокутні характеристики спрацювання. Зокрема, чотирикутна характеристика (рис. 5.5, а) використовується для виконання другого та третього ступеня захисту. Верхня сторона характеристики використовується для фіксації кінців зони, що захищається. Права сторона забезпечує відведення від робочих режимів, а ліва

– відведення від потужностей навантаження, що передаються до місця встановлення захисту. Нижній край характеристики забезпечує роботу захисту при близьких коротких замиканнях через перехідний опір.

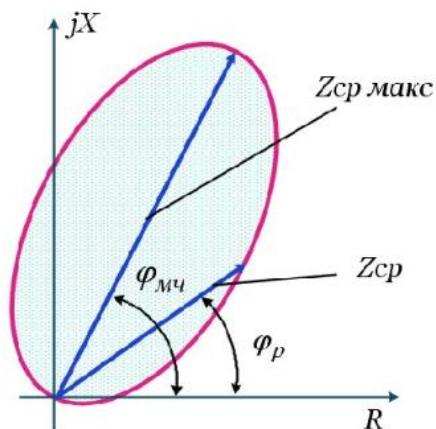


Рисунок 5.4 – Еліптична характеристика спрацювання реле мінімального опору

Для реле мінімального опору третього ступеня може застосовуватися трикутна характеристика, рис. 5.5, б. Така форма характеристики дає змогу відвести захист від навантажувальних режимів та забезпечити необхідну чутливість.

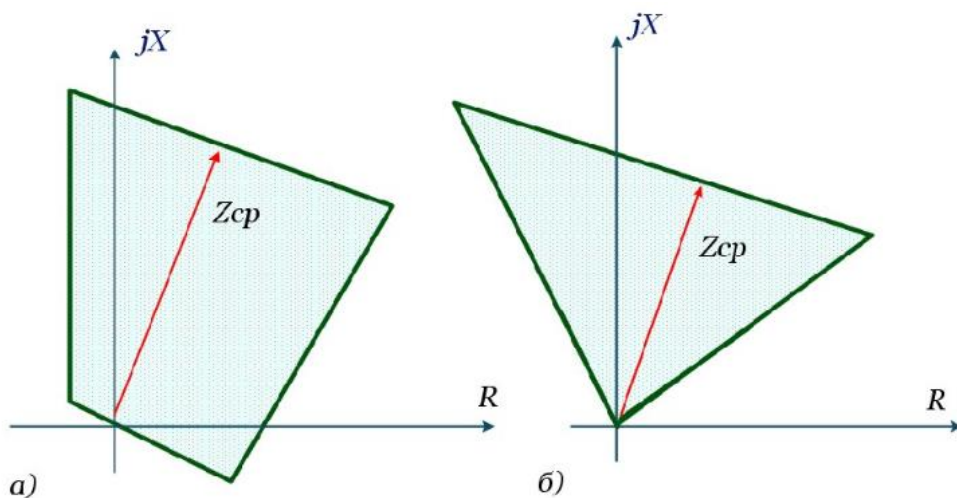


Рисунок 5.5 – Багатокутна характеристика спрацювання реле мінімального опору

На ринку представлено багато типів сучасних мікропроцесорних дистанційних релейних захистів. Зокрема, розглянемо захист SIPROTEC 7SA610, рис. 5.6.



Рисунок 5.6 – Дистанційний релейний захист SIPROTEC 7SA610

Такий релейний захист призначено для виконання наступних захисних функцій:

- 6 контурний захист без перемикання вхідних величин (21/21N);
- захист від замикань на землю при незначних струмах замикань;
- виявлення к.з. на землю в мережах з ізольованою та компенсованою нейтраллю;
- дистанційний захист з високочастотним керуванням;
- визначення відстані до місця ушкодження;
- виявлення хитань;
- струмовий захист;
- захист від ввімкнення на пошкодження;
- захист від зниження та підвищення напруги;
- захист від зміни частоти;
- функція АПВ;
- захист від теплового перевантаження.

Пристрій забезпечує реалізацію наступних функцій контролю:

- контроль вимірювальних кіл;
- самоконтроль реле;
- збереження до енергонезалежної пам'яті осцилограм зміни параметрів;
- збереження даних щодо ввімкнення-відключення високовольтного вимикача.

Пристрій релейного захисту забезпечений інтерфейсами для зв'язку з персональним комп'ютером, IEC 60870-5-103, PROFIBUS-FMS/-DP, DNP 3.0, IEC 618501 Ethernet. Також є в наявності послідовний порт для передачі команд високочастотного управління.

Також для захисту високовольтних електромереж використовується дистанційний релейний захист 7SA6 Siprotec 4 Siemens, рис. 5.7.



Рисунок 5.7 – Дистанційний релейний захист 7SA6 Siprotec 4 Siemens

Цей релейний захист забезпечує широкий спектр захисних функцій на базі системи SIPROTEC 4. Пристрій характеризується високою швидкістю, може захищати короткі ЛЕП. Забезпечує автоматичне виявлення хитань потужності з частотою до 7 Гц. Реалізована можливість пофазного захисту з високочастотним управлінням. Також захист реалізує функції автоматичного повторного ввімкнення. Функції захисту аналогічні переліченим вище.

Обґрунтування принципу функціонування реле мінімального опору

Основними компонентами дистанційного релейного захисту є реле мінімального опору, які спрацьовують в разі зниження імпедансу електромережі, що виявляється за входженням годоргафа повного опору в зону, обмежену характеристикою спрацювання певного ступеня захисту.

Виходячи з цього, в лабораторній роботі розглядається мікропроцесорне реле мінімального опору. Таке реле оцінює стан системи за показами вимірювальних трансформаторів струму та вимірювальних трансформаторів напруги. На рис. 5.8 наведена структурна схема вимірювальних каналів такого реле.

Вимірювання напруги електромережі, миттєві значення якої позначимо $u_A(t)$, здійснюється за допомогою вимірювального трансформатора напруги TV. В якості вимірювальних трансформаторів напруги для мережі 110 кВ можуть бути використані трансформатори серії «НКФ-123-П» (НКФ-110-П). Це однофазні, індуктивні, оливонаповнені трансформатори зовнішнього встановлення. На рис. 5.9 показано підключення кожної фази ТН до системи шин (СШ) через трифазний роз'єднувач із заземлюючими ножами, що комутуються. Для захисту шин від грозових перенапруг використовуються

високовольтні вентильні розрядники. Безпосередньо під опорами ТН монтується розподільча шафа, у якому збираються вторинні кола кожної фази.

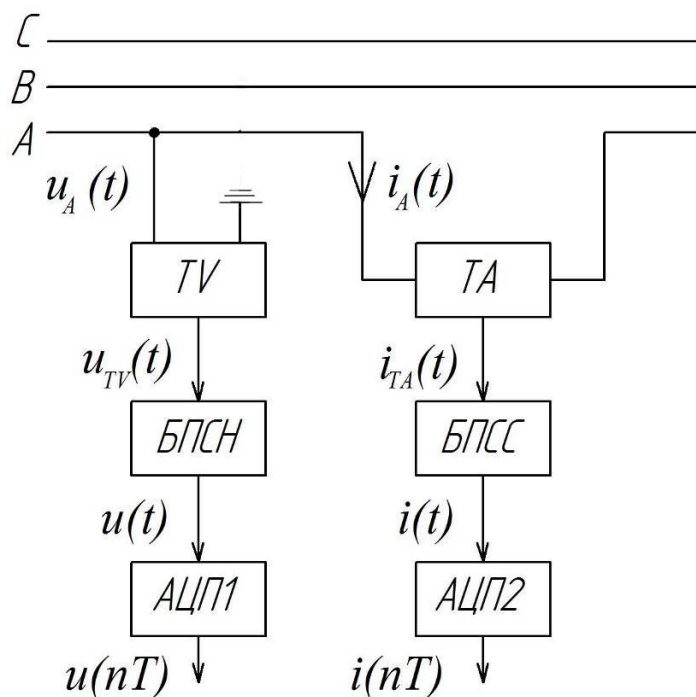


Рисунок 5.8 – Структурна схема вимірювальних каналів напруги і струму у складі навчального реле мінімального опору



Рисунок 5.9 – Розташування вимірювальних трансформаторів напруги на підстанції 110 кВ

Вихідний сигнал $u_{TV}(t)$ вимірювального трансформатора напруги подається до блоку лінійного перетворення сигналів напруги БПСН. Блок призначено для адаптації сигналу до аналого-цифрового перетворення в блоці АЦП1. На виході такого перетворювача видаються дискретизовані значення $u(nT)$, що відповідають миттєвим значенням напруги електромережі в моменти часу $t=0, T, 2T, 3T, \dots$, причому T – період дискретизації сигналу за часом.

Вимірювання струму електромережі здійснюється за допомогою вимірювального трансформатора струму ТА. Для мережі 110 кВ можуть використовуватися трансформатори типу ТФЗМ-110, рис. 5.10. Вторинний струм $i_{TA}(t)$ вимірювального трансформатора подається до блоку перетворювача струмового сигналу БПСС. Цей блок виконує перетворення сигналу за струмом в сигнал за напругою, нормує його. Нормований сигнал подається до АЦП2. Останній видає дискретизовані значення $i(nT)$.



Рисунок 5.10 – Розташування вимірювальних трансформаторів струму типу ТФЗМ-110

Для аналітичного опису дискретизованих значень використовуються вирази:

$$u(nT) = U \cdot \sin(\omega_0 nT + \varphi_u) \quad (5.4)$$

$$i(nT) = I \cdot \sin(\omega_0 nT + \varphi_i) \quad (5.5)$$

де $\omega_0 = 2\pi f_0$ – колова частота напруги (струму) мережі, рад/с;

$f_0 = 50$ Гц – частота напруги мережі.

Дискретизовані значення описуються обертовими векторами наступним чином:

$$\underline{U}_r(nT) = U \cdot e^{j(\omega_0 nT + \varphi_u)} = U_{xr}(nT) + jU_{yr}(nT) \quad (5.6)$$

$$\underline{I}_r(nT) = I \cdot e^{j(\omega_0 nT + \varphi_i)} = I_{xr}(nT) + jI_{yr}(nT) \quad (5.7)$$

Проекції обертових векторів за вимірними миттєвими значеннями можна обчислити наступним чином:

$$U_{xr}(nT) = \frac{u(nT) - u(nT - T)}{\omega_0 T}; \quad (5.8)$$

$$U_{yr}(nT) = u(nT); \quad (5.9)$$

$$I_{xr}(nT) = \frac{i(nT) - i(nT - T)}{\omega_0 T}; \quad (5.10)$$

$$I_{yr}(nT) = i(nT). \quad (5.11)$$

Обертові вектори можна зупинити, якщо помножити на $e^{-j\omega_0 nT}$. В результаті отримуємо нерухомі комплексні вектори, які не є функціями часу $t = 0, T, 2T, 3T, \dots$, а саме:

$$\underline{U}_f = U_{xf} + jU_{yf} = U \cdot e^{j(\omega_0 nT + \varphi_u)} \cdot e^{-j\omega_0 nT} = U \cdot e^{j\varphi_u} \quad (5.12)$$

Проекції обертових векторів можна знайти зі співвідношення:

$$\begin{aligned} \underline{U}_f &= \underline{U}_r(nT) \cdot e^{-j\omega_0 nT} = \\ &= [U_{xr}(nT) + jU_{yr}(nT)] \cdot [\cos(\omega_0 nT) - j \cdot \sin(\omega_0 nT)] \end{aligned} \quad (5.13)$$

Тоді проекції нерухомого (загальмованого) вектора напруги дорівнюють:

$$U_{xf} = U_{xr}(nT) \cdot \cos(\omega_0 nT) + U_{yr}(nT) \cdot \sin(\omega_0 nT) \quad (5.14)$$

$$U_{yf} = U_{yr}(nT) \cdot \cos(\omega_0 nT) - U_{xr}(nT) \cdot \sin(\omega_0 nT) \quad (5.15)$$

Для вектора струму аналогічно одержуємо:

$$\begin{aligned} \underline{I}_f &= I_{xf} + jI_{yf} = I \cdot e^{j\varphi_i} = \underline{I}_r(nT) \cdot e^{-j\omega_0 nT} = \\ &= [I_{xr}(nT) + jI_{yr}(nT)] \cdot [\cos(\omega_0 nT) - j \cdot \sin(\omega_0 nT)] \end{aligned} \quad (5.16)$$

$$I_{xf} = I_{xr}(nT) \cdot \cos(\omega_0 nT) + I_{yr}(nT) \cdot \sin(\omega_0 nT) \quad (5.17)$$

$$I_{yf} = I_{yr}(nT) \cdot \cos(\omega_0 nT) - I_{xr}(nT) \cdot \sin(\omega_0 nT) \quad (5.18)$$

Фактичний імпеданс, що вимірюється реле мінімального опору, може бути визначений наступним чином:

$$\underline{Z}_l = \frac{\underline{U}_f}{\underline{I}_f} = \frac{U_{xf} + jU_{yf}}{I_{xf} + jI_{yf}} = \frac{U_{xf} \cdot I_{xf} + U_{yf} \cdot I_{yf}}{I_{xf}^2 + I_{yf}^2} + j \cdot \frac{I_{xf} \cdot U_{yf} - U_{xf} \cdot I_{yf}}{I_{xf}^2 + I_{yf}^2}. \quad (5.19)$$

Тобто активний опір лінії, що вимірюється реле мінімального опору, становить:

$$R_l = \text{Re}[\underline{Z}_l] = \frac{U_{xf} \cdot I_{xf} + U_{yf} \cdot I_{yf}}{I_{xf}^2 + I_{yf}^2} \quad (5.20)$$

Індуктивний опір лінії дорівнює:

$$X_l = \text{Im}[\underline{Z}_l] = \frac{I_{xf} \cdot U_{yf} - U_{xf} \cdot I_{yf}}{I_{xf}^2 + I_{yf}^2} \quad (5.21)$$

Приймається, що характеристика спрацювання реле мінімального опору має форму кола, рис. 5.11.

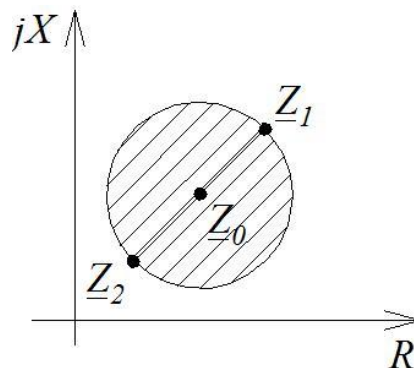


Рисунок 5.11 – Загальний вигляд колової характеристики спрацювання реле опору

Така характеристика спрацювання проходить через опорні точки $\underline{Z}_1 = R_1 + jX_1$ та $\underline{Z}_2 = R_2 + jX_2$. В такому випадку умова спрацювання реле мінімального опору має вигляд:

$$U_{xf}^2 + U_{yf}^2 - a \cdot (R_1 + R_2) - b \cdot (X_1 + X_2) + [I_{xf}^2 + I_{yf}^2] \cdot (R_1 R_2 + X_1 X_2) \leq 0 \quad (5.22)$$

де позначено:

$$a = I_{xf} \cdot U_{xf} + I_{yf} \cdot U_{yf} \quad (5.23)$$

$$b = I_{xf} \cdot U_{yf} - I_{yf} \cdot U_{xf} \quad (5.24)$$

Елементна база навчального реле мінімального опору

Для перетворення змінної вторинної напруги вимірювального трансформатора передбачається використання перетворювача типу ZMPT101B, рис. 5.12. Схема перетворювача включає операційні підсилювачі LM358, які реалізують масштабування сигналу на зсув його у додатну напівплощину для того, щоб сигнал задовольняв вимогам АЦП до вхідного сигналу – від 0 до +5В.

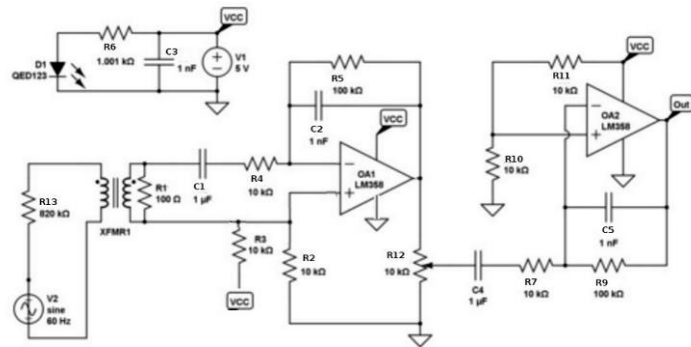
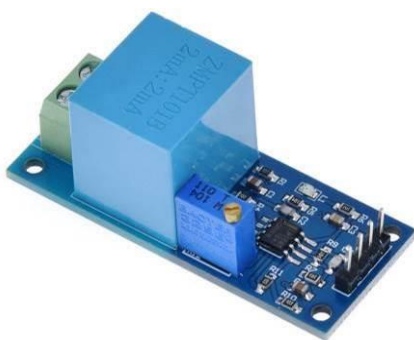


Рисунок 5.12 – Перетворювач напруги ZMPT101B

Для перетворення вторинного струму вимірювального трансформатора струму у сигнал за напругою передбачається використовувати перетворювач типу ACS712, рис. 5.13, функціонування якого засноване на ефекті Холла.

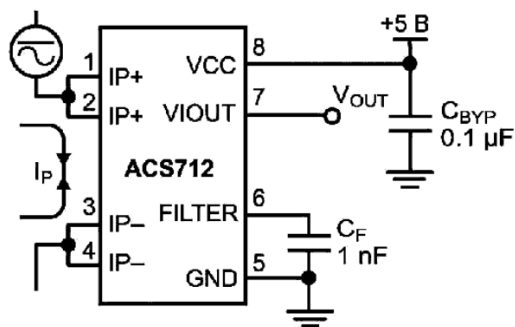
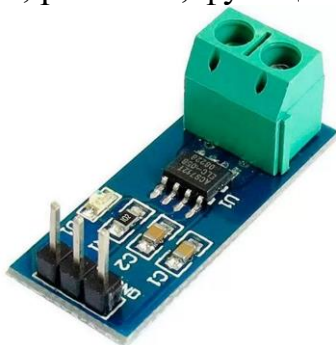


Рисунок 5.13 – Перетворювач струму в напругу ACS712

Необхідно звернути уваги: оскільки принцип дії передбачає використання миттєвих значень напруг та струмів, то в застосуванні додаткових бібліотек для оброблення сигналів з давачів немає потреби.

В якості мікроконтролера може бути використаний 8-розрядний мікроконтролер типу ATmega328p, що розташований на платі типу Arduino UNO.

Навчальна модель мікропроцесорного реле мінімального опору

Для виконання лабораторної роботи використовується навчальна модель мікропроцесорного реле мінімального опору в симуляторі Proteus, рис. 5.14. Модель є спрощеною і включає джерело синусоїдної напруги VSINE (амплітуда 310 В, частота 50 Гц), до якого підключено активний опір $R=18\ \Omega$ та індуктивність $L=0,01\ \text{Гн}$, що імітують опір лінії в нормальному режимі. Опір кола короткого замикання імітується резистором R_{kz} .

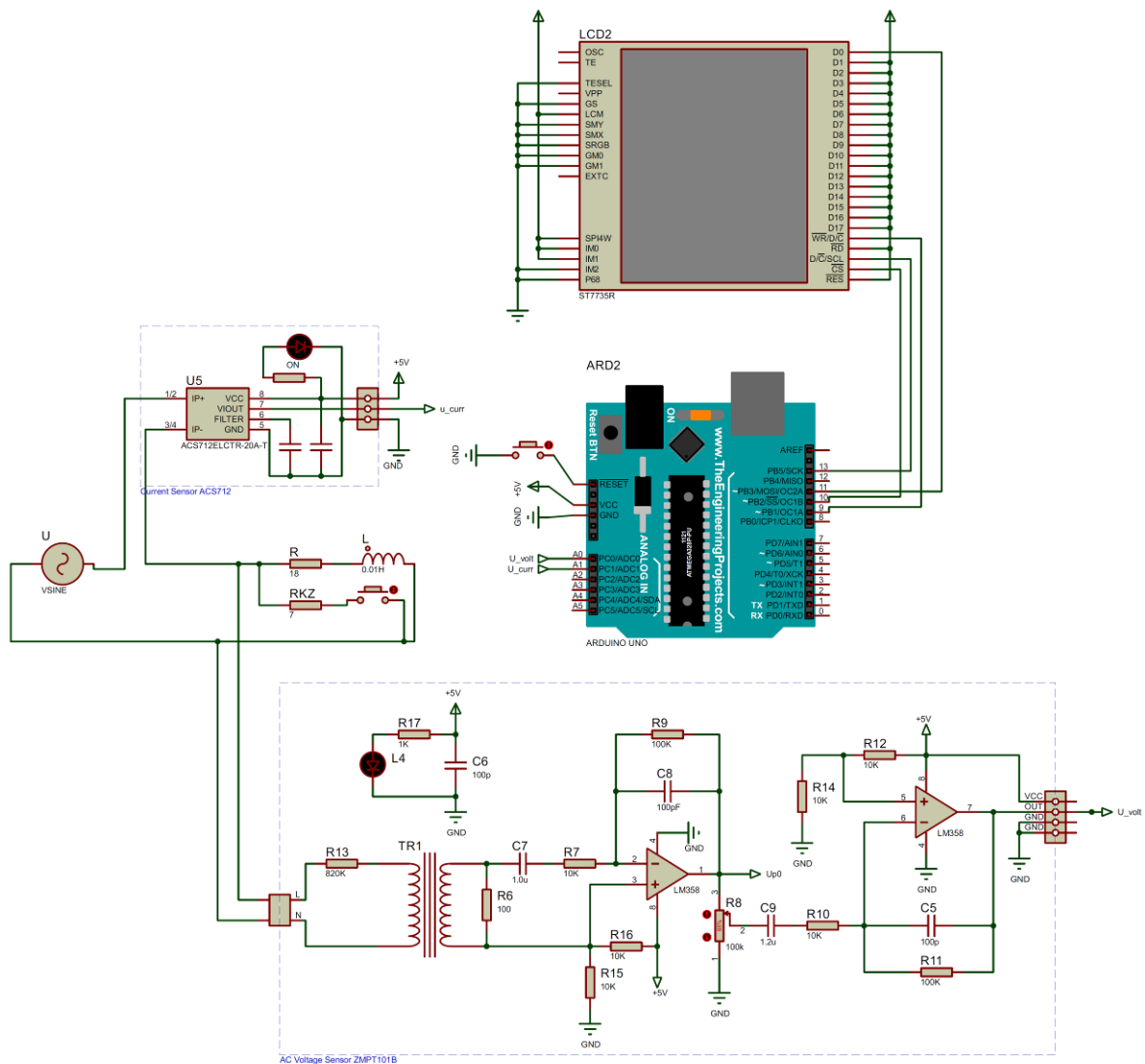


Рисунок 5.14 - Навчальна модель мікропроцесорного реле мінімального опору

Результати функціонування навчального реле мінімального опору представлені на рис. 5.15. На дисплей виводяться значення Z_1 , Z_2 опорних точок характеристики спрацювання реле, значення U_f нерухомого вектора напруги та I_f нерухомого вектора струму, вимірний імпеданс Z_l лінії та значення сигналу off на відключення вимикача в разі виникнення к.з. (off=0 – сигнал на відключення відсутній; off=1 – сигнал на відключення наявний).

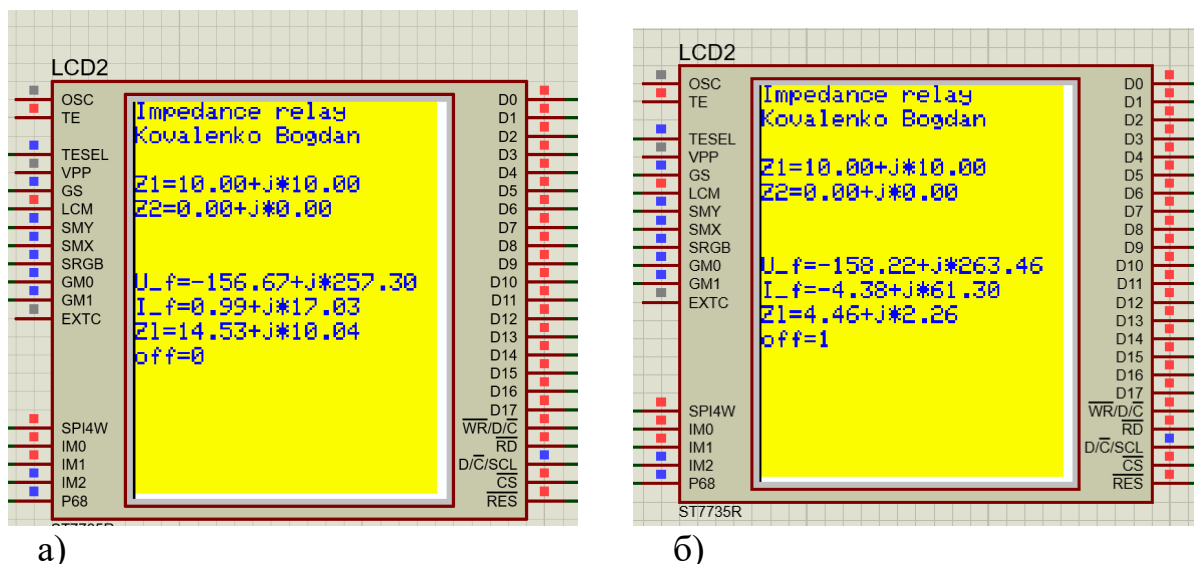


Рисунок 5.15 – Результати функціонування навчального реле мінімального опору: а – в нормальному режимі; б – в разі виникнення короткого замикання

ЗАВДАННЯ

Скласти алгоритм функціонування у вигляді блок-схеми та написати програму для навчального реле мінімального опору (рис. 15.4, 5.15) з коловою характеристикою спрацювання (рис. 5.11). Вихідні дані обираються відповідно до табл. 5.1.

Таблиця 5.1

Вихідні дані для виконання лабораторної роботи

Варіант	Опорні точки характеристики спрацювання				Параметри моделі реле		
	$Z_1 = R_1 + jX_1$		$Z_2 = R_2 + jX_2$		R, Ом	L, Гн	Rkz, Ом
	R_1 , Ом	X_1 , Ом	R_2 , Ом	X_2 , Ом			
1	4	4	0	0	18	0,02	6
2	5	5	-4	-4	16	0,01	5
3	6	4	-3	-2	19	0,03	7
4	4,5	4,6	0	0	22	0,05	4
5	6	6	-1	-1	18	0,07	2
6	5	6	-2	-1	19	0,03	5

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Скласти блок-схему алгоритму функціонування та програму на мові C for Arduino реле мінімального опору, що передбачає:

1.1 Ініціалізацію дисплея

1.2 Відображення на дисплеї незмінного тексту

1.3 Встановлення значень опорних точок характеристики спрацювання. Встановити початкове значення сигналу відключення у лог. 0.

1.4 Запам'ятовування поточного моменту часу (в мікросекундах), цей момент часу вважається початковим для елементарного інтервалу, за який оцінюється положення векторів. Для цього можна скористатися функцією:

$t1 = \text{micros}();$

1.5 Вимірювання миттєвих значень напруги та струму (з одночасним масштабуванням, коефіцієнти обрані для навчальної моделі релейного захисту):

$u1 = 1.04 * (\text{analogRead}(0) - 512);$

$i1 = 0.049 * (\text{analogRead}(1) - 512);$

Кожне з вимірювань триває близько 100 мкс.

1.6 Аналогічно до п. 1.4 та п. 1.5 запам'ятати поточний момент часу ($t2$) та здійснити повторні вимірювання ($u2, i2$).

1.7 Обчислити проекції обертових векторів відповідно до залежностей (5.8)-(5.11). При цьому слід мати на увазі, що тривалість періода T дискретизації вимірювань за часом відповідає різниці збережених моментів часу ($t2 - t1$). Значення вказаної різниці необхідно перерахувати з мкс до секунд. Значення напруги для поточного вимірювання $u(nT)$ відповідає вимірянній величині $u2$, а для попереднього вимірювання – значенню $u1$. Для струму – аналогічно.

1.8 Обчислити проекції загальмованих векторів відповідно до (5.14), (5.15), (5.17), (5.18), при цьому момент часу nT відповідає вимірянній величині $t2$.

1.9 Обчислити активну та реактивну складові імпедансу лінії відповідно до (5.20), (5.21).

1.9 Обчислити проміжні величини відповідно до (5.23), (5.24)

1.10 Обчислити значення умови спрацювання реле мінімального опору відповідно до (5.22). Виконати перевірку: якщо умова (5.22) виконується, то значення сигналу спрацювання встановити у лог. 1.

1.11 Вивести на дисплей обчислені величини відповідно до рис. 5.15. Для підвищення ефективності функціонування програми відображення на дисплеї можливо виконувати не для кожної обчисленої величини, а з проріджуванням. Наприклад, можна виводити значення для кожної 100-ї обчисленої точки.

1.12 Алгоритм продовжує виконувати, починаючи з п. 1.4.

2. Виконати завантаження написаної програми до моделі реле мінімального опору та переконатися у працездатності програми.

3. Накреслити характеристику спрацювання захисту (відповідно до заданого варіанту) з позначеними імпедансами кола в нормальному та аварійному режимах, рис. 5.16.

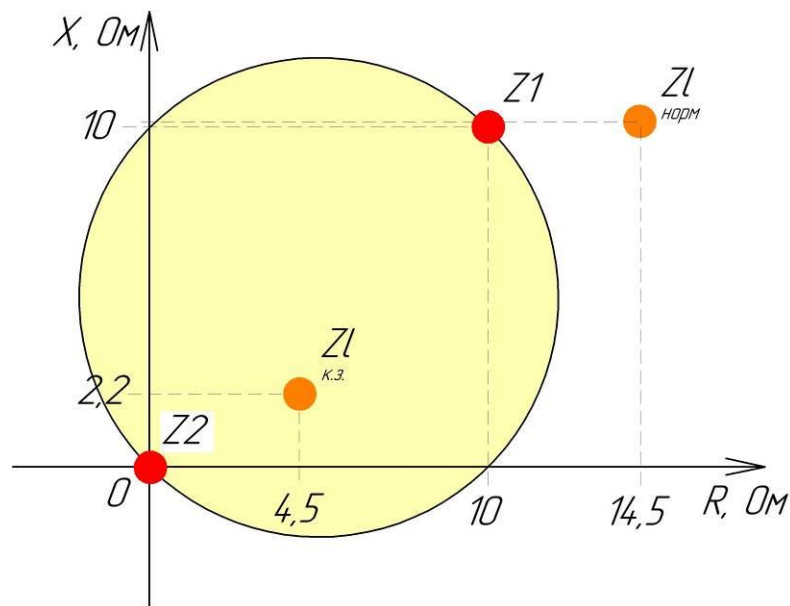


Рисунок 5.16 – Характеристика спрацювання реле мінімального опору та фактичні імпеданси кола в нормальному режимі та при к.з. для параметрів відповідно до рис. 15

Вміст звіту з лабораторної роботи

1. Тема, мета роботи.
2. Блок-схема алгоритму роботи мікропроцесорного реле мінімального опору.
3. Текст програми.
4. Вигляд працездатної моделі у Proteus під час моделювання, вигляд повідомлень на дисплеї в нормальному та аварійному режимах.
5. Характеристика спрацювання захисту (відповідно до заданого варіанту) з позначеними імпедансами кола в нормальному та аварійному режимах (аналогічно до рис. 5.16)
6. Висновки.

Контрольні запитання

1. Призначення та принцип дії дистанційного релейного захисту.
2. Призначення та принцип дії реле мінімального опору.
3. В яких координатах виконується характеристика спрацювання реле мінімального опору? Пояснити суть цієї характеристики.
4. Опишіть види характеристик спрацювання реле мінімального опору.
5. Яку селективність має дистанційних захист? Чим це пояснюється?
6. За рахунок чого дистанційний захист визначає відстань до місця к.з.?

7. Чому дорівнює номінальна вихідна напруги вимірювального трансформатора напруги? Чому дорівнює номінальний вторинний струм вимірювального трансформатора струму?
8. З якою частотою обертається вектор напруги?
9. Як зрозуміти: дискретизовані миттєві значення напруги?
10. Яким чином можна зупинити обертові вектори?
11. Для чого зупиняють обертові вектори?
12. Яким чином пов'язаний модуль загальмованого вектора та діюче значення виміряної напруги?
13. Чим визначається положення загальмованого вектора напруги (або струму) на комплексній площині?
14. Доведіть правильність виразу (5.19).

ЛІТЕРАТУРА

1. Попович М. Г., Ковальчук О.В. Теорія автоматичного керування : підручник. 2-ге вид., перероб. і доп. К. : Либідь, 2007. 656 с.
2. Новацький А. О. Мікропроцесорні та мікроконтролерні системи : підручник. У 2 ч. Ч. 1. Мікропроцесорні системи. Київ : КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2020. 361с.
3. Колонтаєвський Ю. П., Сосков А. Г. Електроніка та мікросхемотехніка : підручник 2-е вид. К. : Каравела, 2009. 416 с.
4. Жуйков В. Я., Терещенко Т. О., Ямненко Ю. С., Заграничний А. В. Мікропроцесорна техніка : підручник. НТУУ «КПІ»; ред. О. В. Борисов. Київ : НТУУ «КПІ», 2016. 440 с.
5. Електроніка і мікропроцесорна техніка / Сенько В. І. та ін. К. : «Агроосвіта», 2015. 676 с.
6. Сучасні мікроконтролери в електронній та інформаційно-вимірювальній техніці: навч. посіб. / Вовна О. В. та ін. Покровськ : ДВНЗ «ДонНТУ», 2020. 311 с.
7. Промислові мережі та інтеграційні технології в автоматизованих системах : навч. посібник / Пупена О. М. та ін. К. : Вид-во «Ліра-К», 2011. 552 с.
8. Яндульський О. С., Дмитренко О. О. Релейний захист. Цифрові пристрої релейного захисту, автоматики та управління електроенергетичних систем : навч. посіб. К. : НТУУ «КПІ», 2016. 102 с.
9. Thorpe E. Arduino: Advanced Methods and Strategies of Using Arduino. Independently Published, 2020. 224 p.
10. Geddes M. Arduino Project Handbook. 25 Practical Projects to Get You Started. San Francisco: 2016. 275 p.
11. Кідиба В. П. Релейний захист електроенергетичних систем : підручник. Львів : Видавництво Національного університету "Львівська політехніка", 2013. 533 с.