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

04-03-387М

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт з навчальної дисципліни «Мікропроцесорні системи управління та захисту в енергетиці» для здобувачів вищої освіти другого (магістерського) рівня за освітньо-професійною програмою «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» денної та заочної форм навчання

Рекомендовано науково-методичною радою з якості ННІ ЕАВГ
Протокол № 8 від 23.04.2024 р.

Рівне – 2024

Методичні вказівки до виконання лабораторних робіт з навчальної дисципліни «Мікропроцесорні системи управління та захисту в енергетиці» для здобувачів вищої освіти другого (магістерського) рівня за освітньо-професійною програмою «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» денної та заочної форм навчання. [Електронне видання] / Василець С. В., Василець К. С., Ільчук В. В. – Рівне : НУВГП, 2024. – 56 с.

Укладачі:

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

Василець К. С. старша викладачка кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій, докторка філософії;

Ільчук В. В. старший викладач 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 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»:
Рудик А. В., професор кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій, доктор технічних наук, професор.

ЗМІСТ

ВСТУП

Освітня компонента «Мікропроцесорні системи управління та захисту в енергетиці» має сформувати у здобувачів вищої освіти вміння та навички проектування, програмування та експлуатації мікропроцесорних систем, що використовуються для управління та захисту об’єктів електроенергетичної галузі.

Здобувачі мають одержати знання щодо особливостей функціонування неперервних та дискретних систем автоматичного управління та релейного захисту, знати способи фільтрації дискретизованих сигналів на програмному рівні мікропроцесорних пристроїв. Також здобувач набуде знань щодо підходів до математичного моделювання цифрових пристроїв керування, давачів струмів, напруг, особливостей побудови систем управління на цифровій елементній базі, улаштування цифрових органів у складі пристроїв релейного захисту та автоматики.

Лабораторні роботи базуються на теоретичному матеріалі, тому підготовка до виконання лабораторної роботи передбачає не тільки ознайомлення з даними методичними вказівками, але і закріплення лекційного курсу. Для виконання лабораторних робіт використовуються плати ARDUINO, дисплеї, моделі мікропроцесорних пристроїв у PROTEUS, програмне забезпечення IDE Arduino, Proteus, SinaProg.

Результати, що були одержані в результаті виконання лабораторної роботи, оформлюються у вигляді звіту. Вимоги до вмісту звіту наведені у вказівках до кожної лабораторної роботи. Звіт необхідно підготувати та захистити на наступному лабораторному занятті.

ЛАБОРАТОРНА РОБОТА №1

Програмування графічного дисплея

Мета: оволодіти навичками виведення графічних зображень на дисплей.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Для розширення фунціональних можливостей термінали релейного захисту обладнують графічним дисплеєю. Це розширює можливості з відображення інформації, спрощує налаштування значень параметрів. Зокрема, на рис. 1.1 наведено пристрій РЗА типу SIPROTEC 5, що випусткається фірмою Siemens. Графічний дисплей забезпечує відображення схеми силових кіл шафи та величини струму, напруги тощо.

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

Найчастіше у складі терміналів використовуються TFT LCD дисплеї. Це рідкокристалічні дисплеї, що обладнані активною матрицею, для керування якою використовуються тонкоплівкові транзистори. Найчастіше застосовуються полікремнієві тонкоплівкові транзистори (P-Si TFT). Вони дозволяють вмикати окремі пікселі та змінювати їх колір. Дисплеї мають незначну затримку (до 30 мс), що дає змогу відрображати відеосигнал в реальному часі.

В лабораторній роботі досліджується дисплей TFT LCD, що має діагональ 1,8 дюйма, характеризується роздільною здатністю 128x160 точок, глибина кольору становить 262000 кольорів на пиксель. Такий дисплей виконано на

базі контролера ST7735, рис. 1.2. Для керування дисплеєм використовується інтерфейс SPI.

Рисунок 1.2 – TFT LCD дисплей на базі контролера ST7735

Керування дисплеєм здійснюється за допомогою наступних виводів:

$\overline{\text{CS}}$ – вибір мікросхеми;

SCK – тактовий сигнал;

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

SDA (MOSI) – лінія передачі даних до дисплея.

$\overline{\text{RES}}$ – 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

Для відображення текстової та графічної інформації на LCD та OLED дисплеях, що підключені до Arduino, використовуються можливості бібліотеки Adafruit_GFX. Така бібліотека для видачі даних на конкретний дисплей використовує додатову бібліотеку, яка адаптована для дисплея визначеного типу. Наприклад, для TFT LCD дисплея, який обладнано чіпом ST7735, в якості додаткової використовується бібліотека Adafruit_ST7735. Обмін даними з дисплеєм здійнюється за протоколом SPI, тому використовуються функції бібліотеки SPI.h.

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 tft, яка дозволяє адресувати дисплей, визначається функцією:

```
Adafruit_ST7735 tft = Adafruit_ST7735(10,  9);
```

де перший аргумент визначає номер виводу плати /CS, до якого підключений дисплей (в даному випадку 10), другий аргумент – номер виводу плати DC (в даному випадку 9).

Для ініціалізації дисплея, що обладнаний контролером ST7735S, використовується команда:

```
tft.initR(INITR_BLACKTAB);
```

Система координат

Цифрове зображення на дисплеї складається з окремих пікселів – точок, колір кожної з яких може задаватися. Пікселі мають квадратну форму, розмір пікселя фіксований. Кількість пікселів на екрані визначає розділну здатність. Пікселі організовані у декартовій системі координат, початок координат суміщений з верхнім лівим кутком екрану, рис. 1.3.

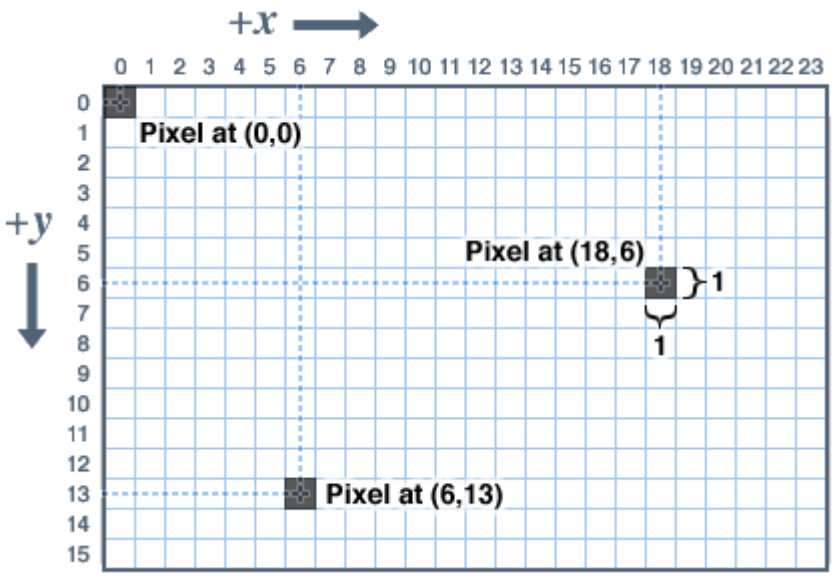


Рисунок 1.3 – Координатна сітка дисплея

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

Колір кожного пікселя на дисплеї задається 16 розрядним двійковим числом, рис. 14. Для формування загального кольору використовуються три базових: червоний (старші 5 біт), зелений (середні 6 біт), синій (молодші 5 біт).

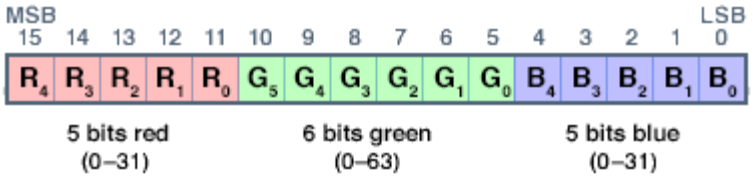


Рисунок 1.4 – Двобайтовий формат представлення кольору

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 (крапка)

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);

де перший аргумент – координата пікселя за віссю абсцис;

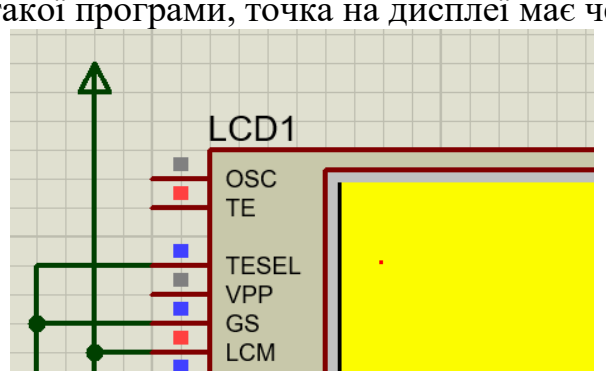
другий аргумент – координата пікселя за віссю ординат;

третій аргумент – символічне ім'я кольору пікселя.

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

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

Adafruit_ST7735 tft = Adafruit_ST7735(10,  9);

void setup() {
   tft.initR(INITR_BLACKTAB);
   tft.fillScreen(ST7735_YELLOW);
   tft.drawPixel(10,20,ST7735_RED);
}

void loop() {

}
```

Після завантаження такої програми, точка на дисплеї має червоний колір:



Лінія

Для креслення ліній 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);

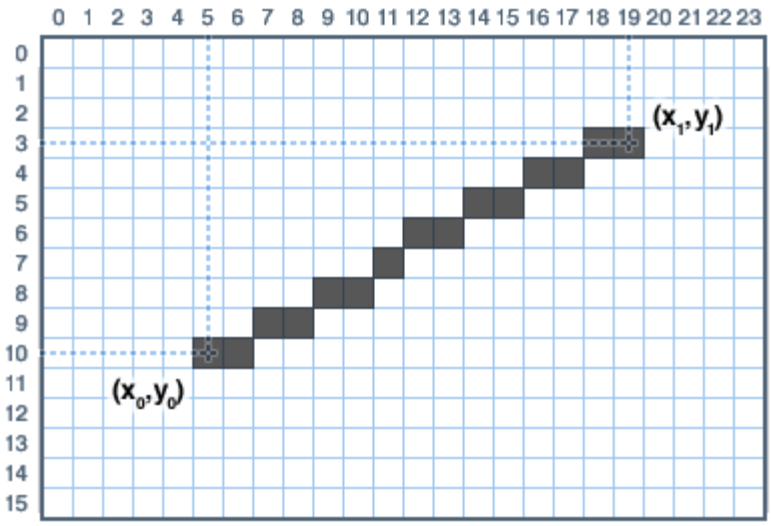


Рисунок 1.5 – Координати лінії

Для того, аби накреслити лінію зеленого кольору, яка йде з початку координат до точки з координатами (20, 50), необхідно використати наступну команду:

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

Для вертикальних (горизонтальних ліній застосовується команда drawFastVLine (drawFastHLine). 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(uin86_t x0, uin86_t y0, uint8_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);
```

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);
```

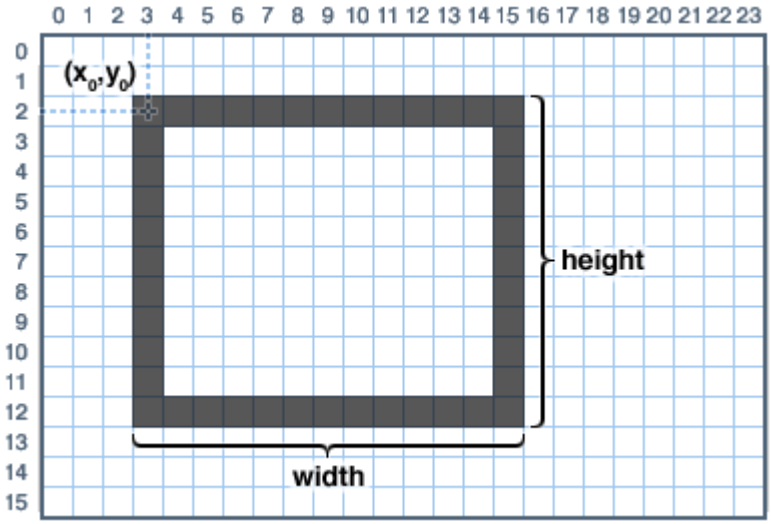


Рисунок 1.6 – Параметри прямокутника

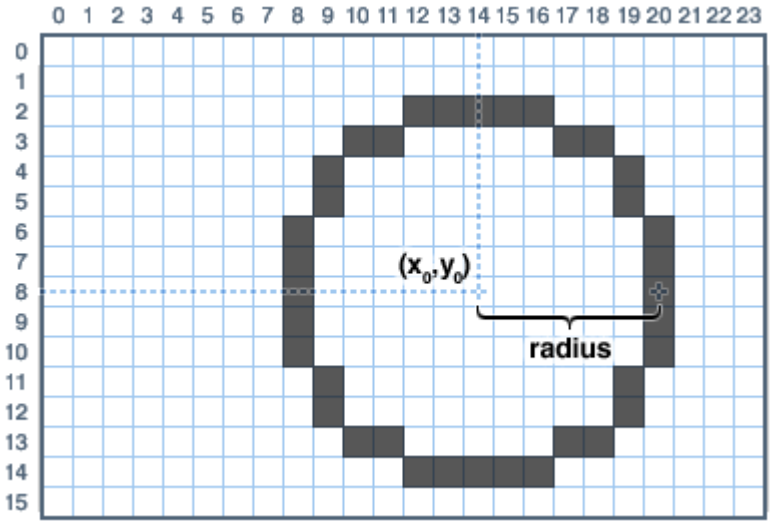


Рисунок 1.7 – Параметри кола

Прямокутник із заокругленими кутами

Така команда потребує задавання координат лівого верхнього кута (позначаються x_0, y_0), ширини (w), висоти (h) прямокутника та радіусу (r) заокруглення кута. Значення таких параметрів вказується в пікселях. Також в якості параметра задається 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);
```

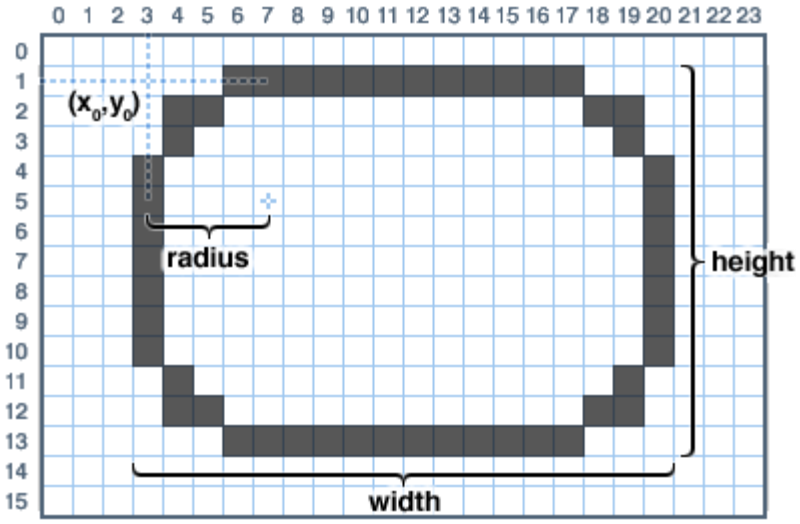


Рисунок 1.8 – Параметри прямокутника із заокругленими кутами

Трикутник

Для того, аби накреслити трикутник, необхідно задати наступні дані (рис. 1.9):

(x_0, y_0) - координати першого кута;
(x_1, y_1) - координати другого кута;
(x_2, y_2) - координати третього кута;
колір.

Для виводу трикутника слід використовувати наступну функцію:

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

Для виведення залитого кольором трикутника використовується функція:

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

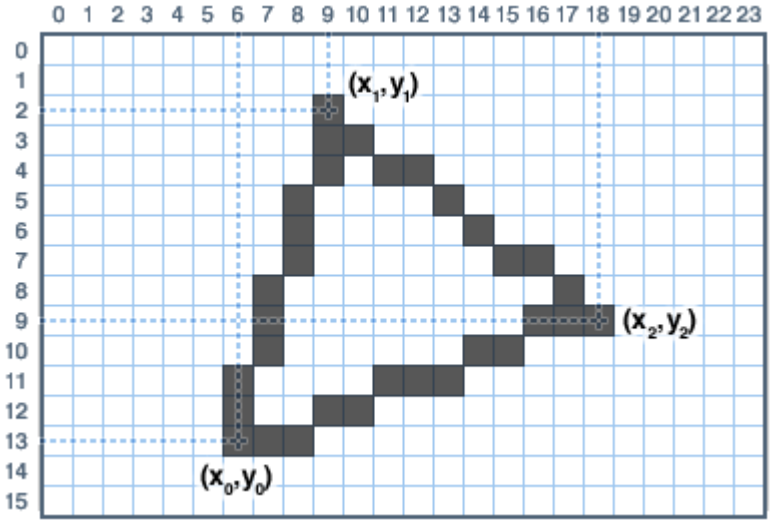


Рисунок 1.9 – Параметри трикутника

Відображення символа

Один символ виводиться на екран з використанням функції drawChar. Символ має розмір 5x8 пікселів, рис. 1.10. Розмір символа можна масштабувати. Для масштабного коефіцієнта size=2 розмір символу розмір символу зміниться на 10x16 пікселів. Синтаксис функції drawChar:

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

До параметрів функції drawChar відносяться:

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.

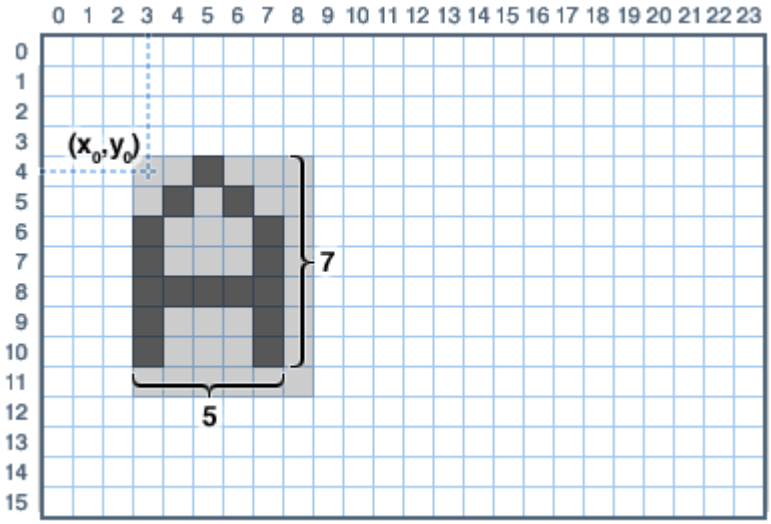
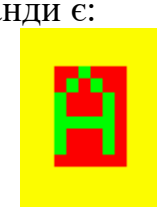


Рисунок 1.10 – Параметри символа

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

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

Результатом виконання команди є:



Виведення рядка символів (текста)

Виведення рядків передбачає використання декількох команд:

– встановити координати верхнього лівого кутка рядка – функція setCursor(x, y);

– задати колір текста (color) та фона (backgroundcolor, цей параметр можна не вказувати), на якому текст відображається – функція setTextColor(color, backgroundcolor);

– задати розмір символів (size=1, 2 або 3) – функція setTextSize(size);

– визначити необхідність перенесення тексту (w=1 – перенесення дозволено, w=0 – заборонено) – функція setTextWrap(boolean w).

Синтаксис таких функцій:

```
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);
```

Виведення рядка символів здійснюється з використанням функції print() або println(). Друга функція переводить курсор 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 унеможливлює їх обчислення на Arduino.

```
setRotation(uint8_t rotation);
```

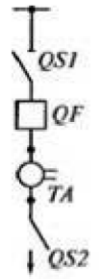
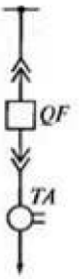
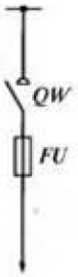
де rotation – кут повороту екрана в градусах: 0 = 0°, 1 = 90°, 2 = 180°, 3 = 270°.

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. На піксельній системі координат накреслити схему головних з’єднань шафи, рис. 1.11.

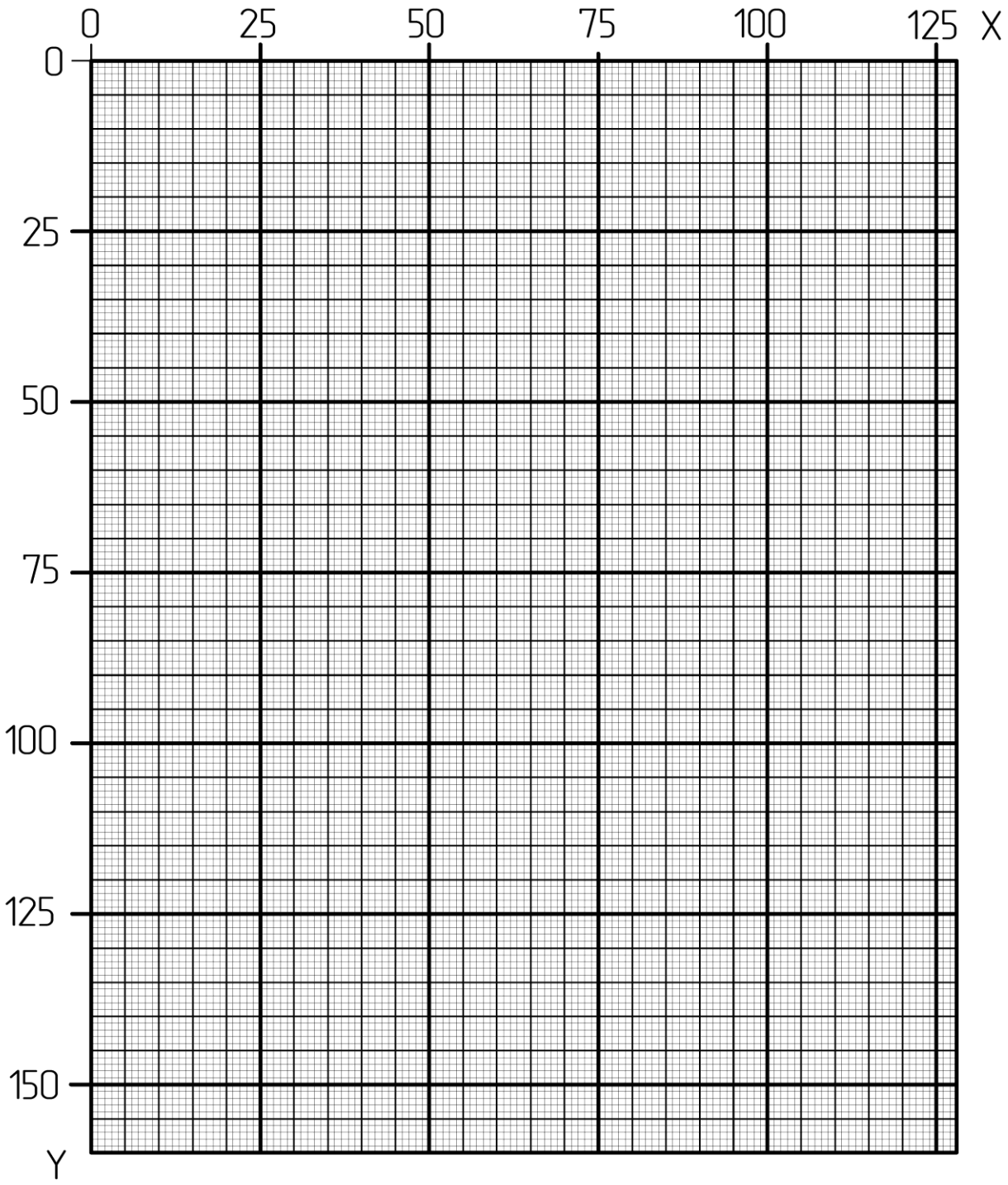


Рисунок 1.11 – Система координат дисплея роздільною здатністю 128x160

2. Для кожного графічного примітива (лінія, коло, прямокутник тощо) визначити координати опорних точок, рис. 1.12.

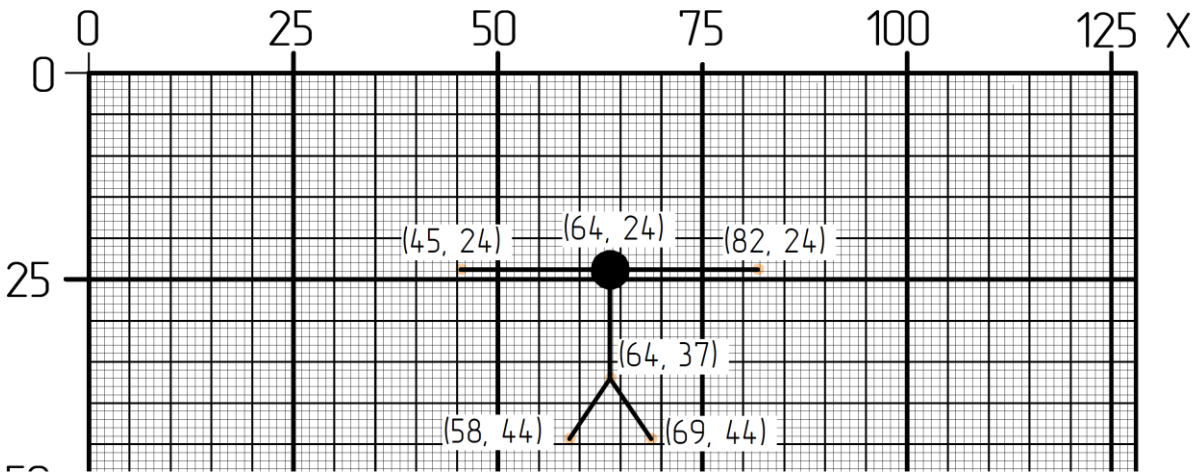


Рисунок 1.12 – Приклад нанесення координат графічних примітивів

3. Виконали модель в симуляторі Proteus, рис. 1.13.

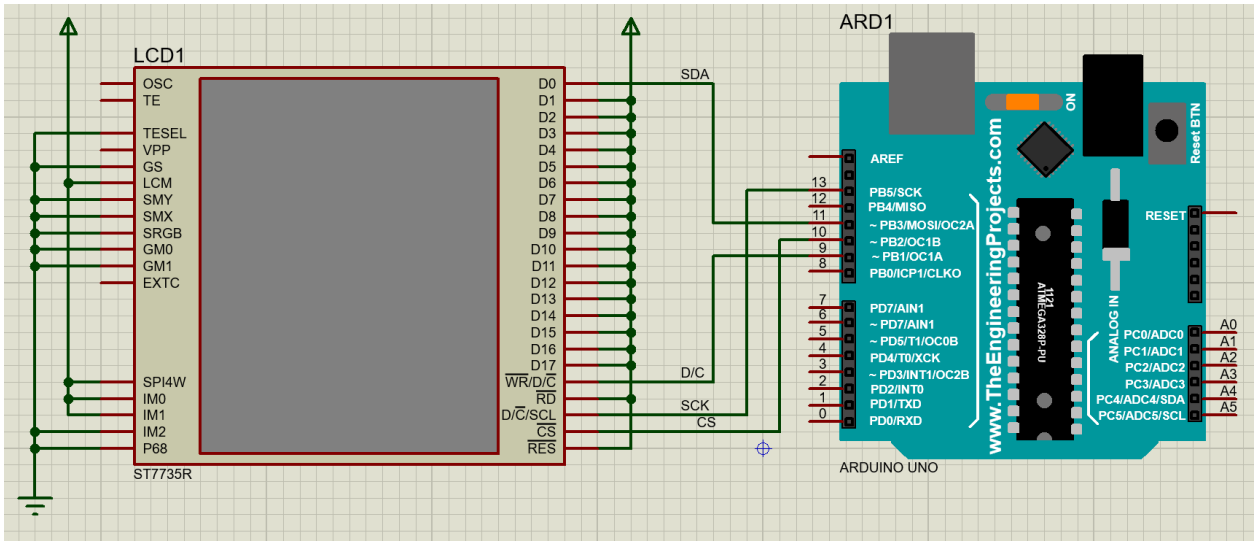


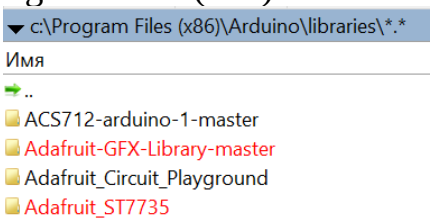
Рисунок 1.13 – Модель приєдання TFT LCD диспля з контролером ST7735 до плати ArduinoUNO

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.

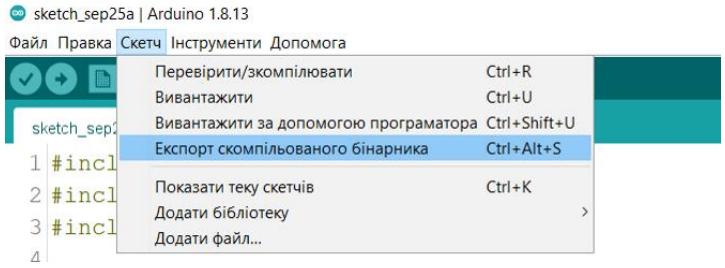
6. Скласти програму для плати Arduino, яка відображає на дисплеї задане креслення. В якості додаткового джерела інформації можна користуватися тестовою програмою:

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

Результат роботи програми:

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

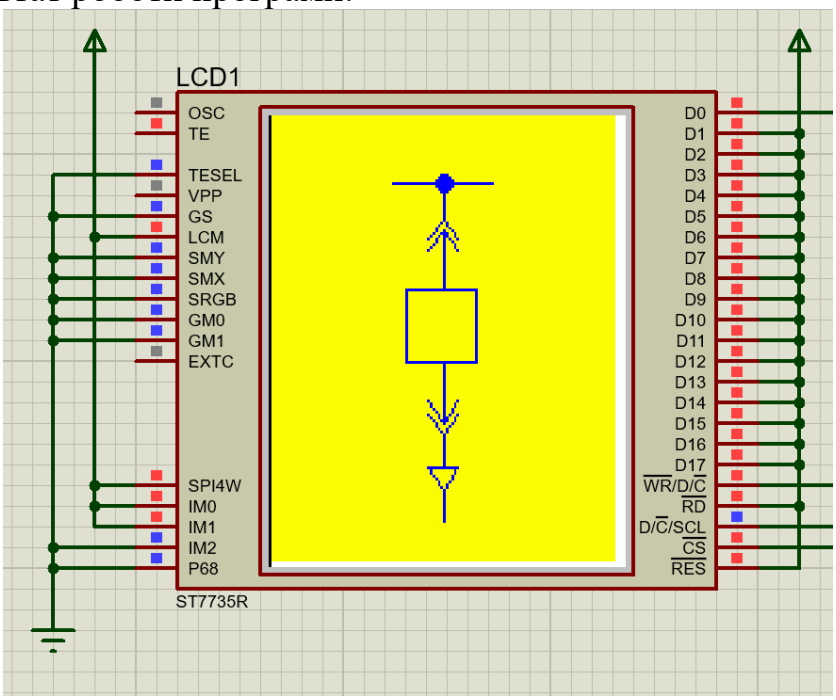
7. Компіляція кода виконується наступним чином:



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. Результат роботи програми:



Вміст звіту з лабораторної роботи

1. Тема, мета роботи.
2. Система координат дисплея з нанесеними координатами графічних примітивів.
3. Програма для для мікроконтролера.
4. Результат виконання програми.

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. Які функції змінює колір пікселя, відображають лінію? Які параметри таких функцій?
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

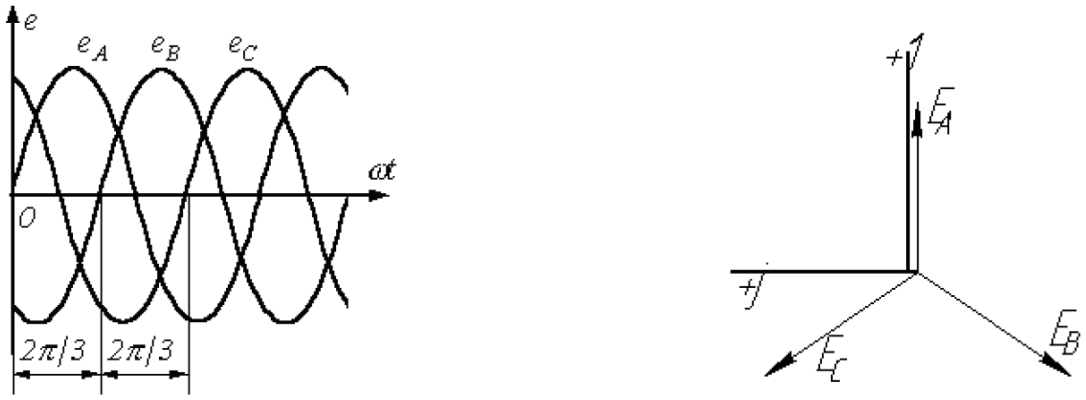
Розроблення програмного забезпечення для мікропроцесорного пристрою визначення порядку чергування фаз

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.



а) б)

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

Розглянемо випадок, коли три однакові котушки розміщені так, що кут між їх осями становить 120º (рис. 2.2). Припустимо, що по котушках протікають струми, які є синусоїдними:

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

$$i_2 = I_m \sin(\omega t - 120°);$$

$$i_3 = I_m \sin(\omega t + 120°).$$

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.

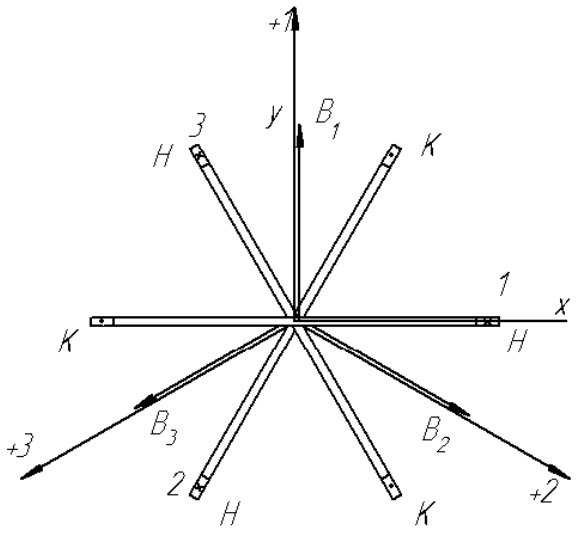


Рисунок 2.2 – Утворення обертового магнітного поля трьома котушками, що зсунуті на 120 ел. градусів

Для коректного функціонування робочого механізму необхідно забезпечити правильний напрямок обертання двигуна. Як відомо, напрямок обертання асинхронного двигуна визначається порідком чергування фаз. Тому, перед підключенням двигуна, необхідно бути впевненим, що порядок чергування фаз відомий і забезпечує необхідний напрямок обертання, що не допустить виходу з ладу технологічного обладнання.

Для визначення порядку чергування фаз може бути застосований вказівник, до складу якого входять дві лампи розжарювання HL1, HL2 (однакової потужності) та конденсатор С (рис. 2.3). Позначимо потужність лампи P, тоді умовою вибору величини ємності є тотожність ємнісного опору конденсатора X_C та активного опору R лампи:

$$R = X_C. \tag{1}$$

Оскільки

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

то

$$C = \frac{P}{\omega U^2}. \tag{3}$$

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

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

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

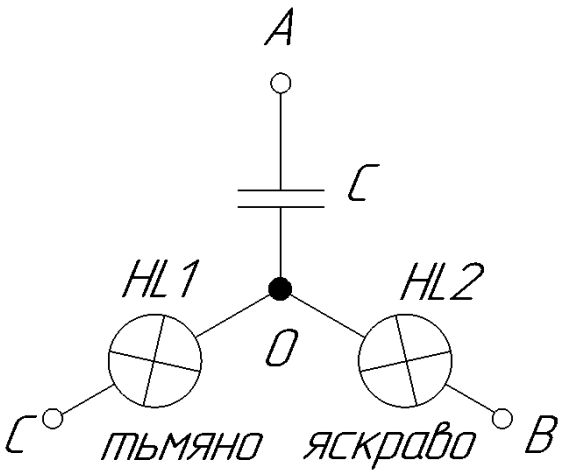


Рисунок 2.3 – Схема вказівника послідовності чергування фаз

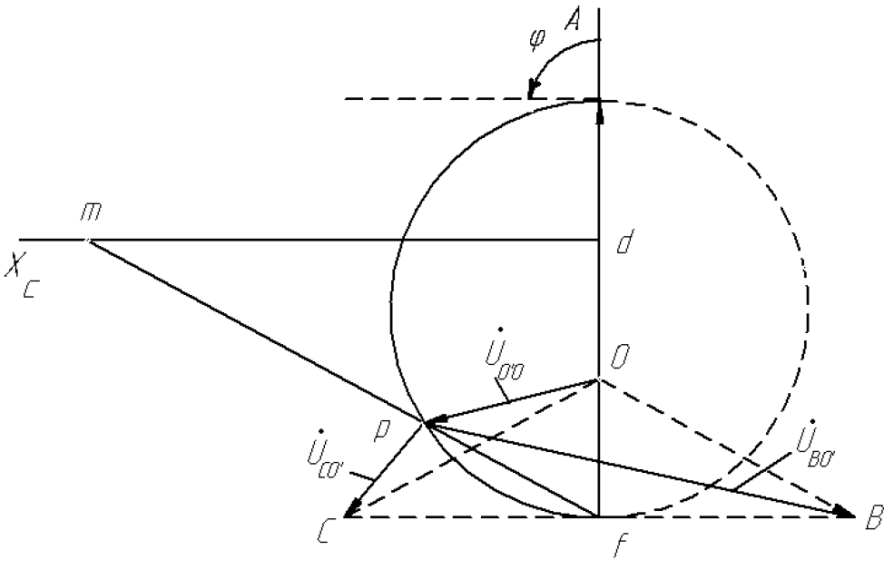


Рисунок 2.4 – Векторна діаграма, що пояснює функціонування вказівника фаз

Позначимо фазу, яка приєднана до конденсатора, А. Тоді, як показує векторна діаграма, напруга на лампах не буде тотожною. Напруга на лампі, що живиться від фази В, яка відстає від фази А на 120º, відповідає вектору $\dot{U}_{BO'}$. Напруга на лампі, що живиться від фази С, яка випереджає фазу А на 120º, відповідає вектору $\dot{U}_{CO'}$. Як видно з векторної діаграми $\dot{U}_{BO'} > \dot{U}_{CO'}$. Тому лампа, що ввімкнена у фазу В, буде горіти яскравіше, ніж ввімкнена у фазу С.

Таким чином, якщо фазу, що підключена до конденсатора, прийняти за А, то фаза, до якої підключена яскравіша лампа, є фазою В, а фаза, де лампа горить тьмяно, є фазою С.

Принцип дії пристрою визначення порядку чергування фаз

Пристрій визначає порядок чергування фаз шляхом вимірювання напруги на лампі розжарювання. Виміряна напруга порівнюється з уставкою. В разі перевищення уставки (відповідна лампа яскраво горить), на дисплеї навпроти такої лампи необхідно вивести текст «фаза В». А біля лампи, яка світиться тьмяно, має виводитися текст «фаза С». Навпроти вивода, який приєднаний до конденсатора, постійно має відображатися текст «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 маштабування 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.

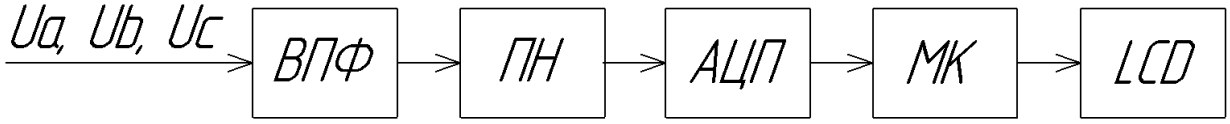


Рисунок 2.5 – Структурна схема мікропроцесорного пристрою визначення порядка чергування фаз трифазної електромережі

Принципова електрична схема пристрою

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

До складу пристрою входять лампи HL1, HL2 і конденсатор С. Такі елементи з'єднані за схемою «зірка». Затискач, до якого приєднано конденсатор, має позначку «А». Виводи ламп HL1, HL2 позначені X1, X2, відповідно. Схема DA1, що вимірює напругу, приєднана між X2 та нульовою точкою зірки. Використовується перетворювач типу ZMPT101B (рис. 2.7).

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 мільтиплексора, 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.

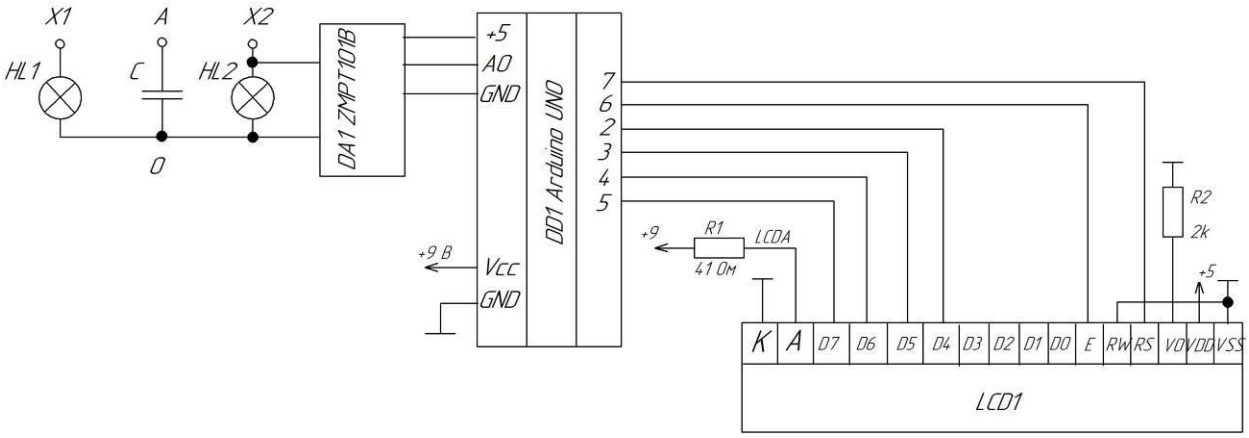
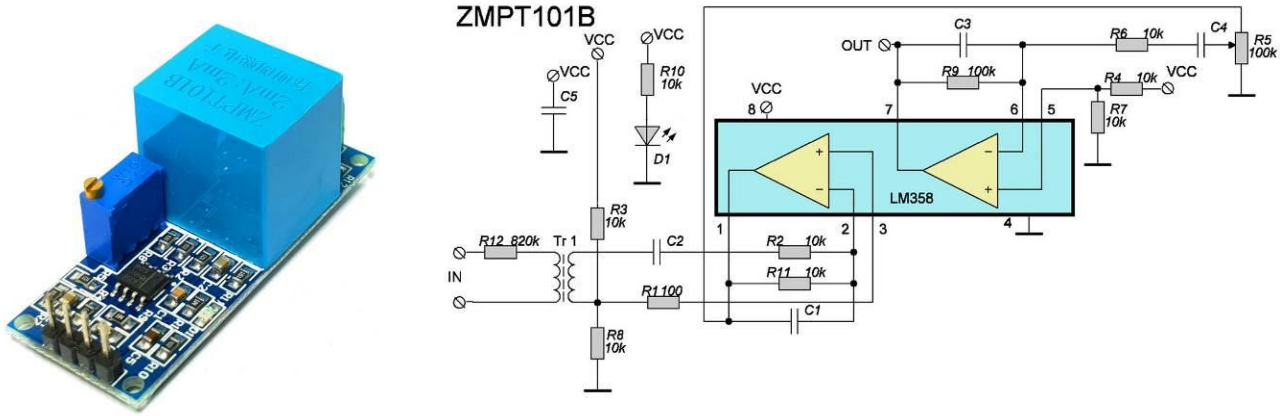


Рисунок 2.6 - Принципова електрична схема мікропроцесорного пристрою визначення порядка чергування фаз трифазної електромережі

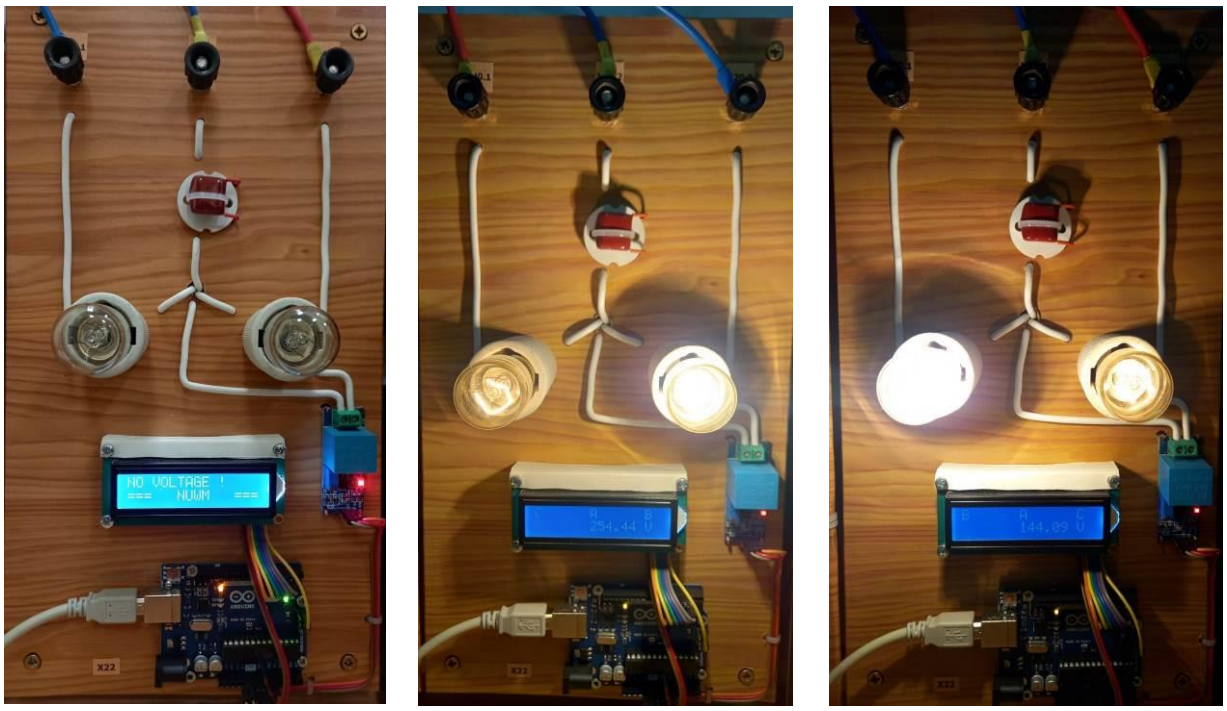


а) б)

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

Для відображення порядку чергування фаз використовується рідкокристалічний індикатор LCD1 типу WH1602A, що підключається до мікроконтролера за допомогою паралельного 4х 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.



а *б* *в*

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

Обчислення середньоквадратичного значення напруги за показами перетворювача ZMPT101B здійснюється за допомогою бібліотеки Filters.h. Підключення такої бібліотеки здійснюється директивою:

```
#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

Відлагодження програми здійснюється на пристрої в ауд. 509 (рис. 2.8), або з використанням моделі Proteus, рис. 2.9. До складу моделі входить трифазне джерело V1, що живить ламповий вказівник HL1, C, HL2. Вимірювання напруги здійснюється вольтметрами. Кнопка S1 подає напругу на вказівник. S2 змінює порядок чергування фаз. Окрема підсистема симулює давач напруги ZMPT101B.

ЗАВДАННЯ

Сформувати програму для мікроконтролера, що функціонує у складі пристрою визначення послідовності фаз. Положення фази А є фіксованим, пристрій має визначати місце підключення фази В, що відстає від фази А, та фази С, що випереджає фазу А. Крім того, на дисплеї має відображатися середньоквадратичне значення напруги, що вимірюється давачем на одній з

ламп. Дані на дисплеї мають оновлюватися раз на 1 с. При відсутності напруги на виході давача, на дисплеї виводиться повідомлення з прізвищем виконавця.

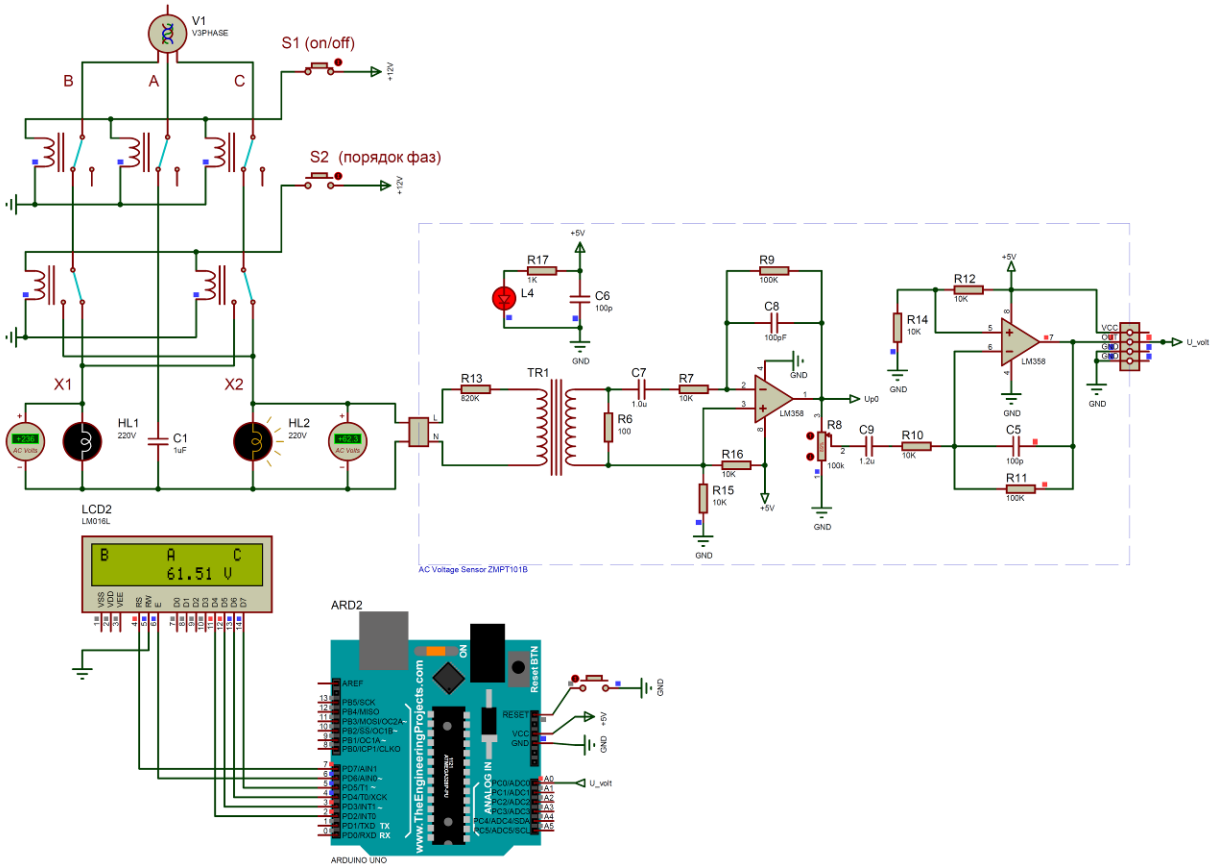


Рисунок 2.9 – Модель пристрою визначення порядку чергування фаз у Proteus

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. Сформувати блок-схему, що відображає алгоритм функціонування пристрою визначення послідовності фаз.

3. Скласти текст програми для ArduinoUNO в інтегрованому середовищі ArduinoIDE, що реалізує алгоритм.

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 B, A, C; при порядку фаз C, A, B.
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. Пояснити, чому лампа, яка живиться від фази B, горить яскравіше за лампу, що живиться від фази C.
6. Яким чином здійснюється вимірювання середньоквадратичного значення змінної напруги?
7. Для чого використовується трансформатор у перетворювачі ZMPT101B?
8. Для чого використовуються операційні підсилювачі у перетворювачі 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. Поясність 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

Рівень освітленості робочого місця має важливе значення для створення безпечних умов праці та руху людей по виробничим приміщенням, рис. 3.1. Вимоги до рівня освітленості робочих місця працівників різних підприємств наведені у ДБН В.2.5-28:2018 «Природне і штучне освітлення».

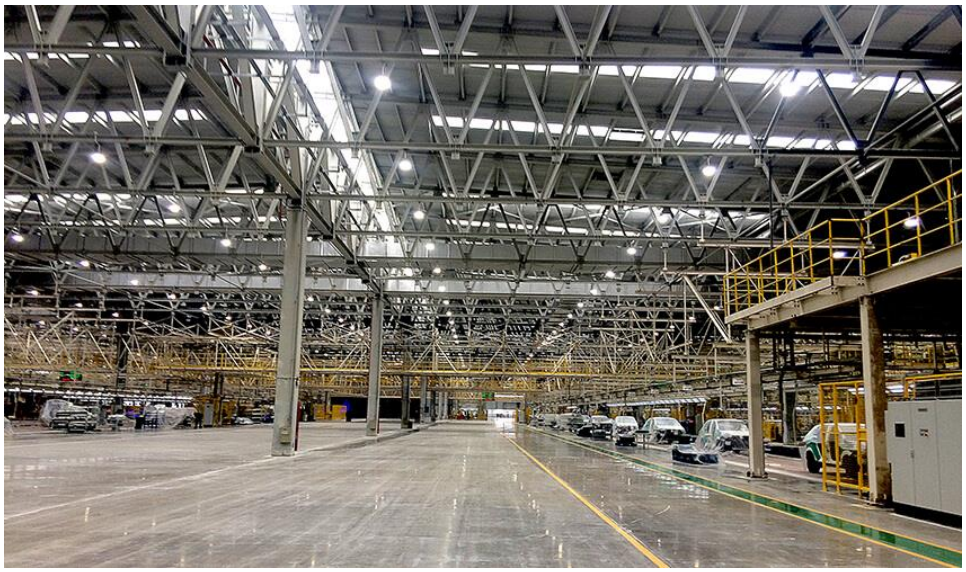


Рисунок 3.1 – Розташування освітлювальних приладів в цеху промислового підприємства

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 степені 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.

Живлення приладів освітлення як всередині приміщень, так і назовні, в більшості випадків здійснюється напругою не більше 220 В. Як виключення, відповідно до п. 6.1.13 ПУЕ, може застосовуватися напруга 380 В.

Групові мережі, що живлять освітлювальні прилади, можуть бути однофазними трипровідними або трифазними п'ятипровідними. Освітлювальні прилади бажано розподілити між фазами рівномірно (рис. 3.2).

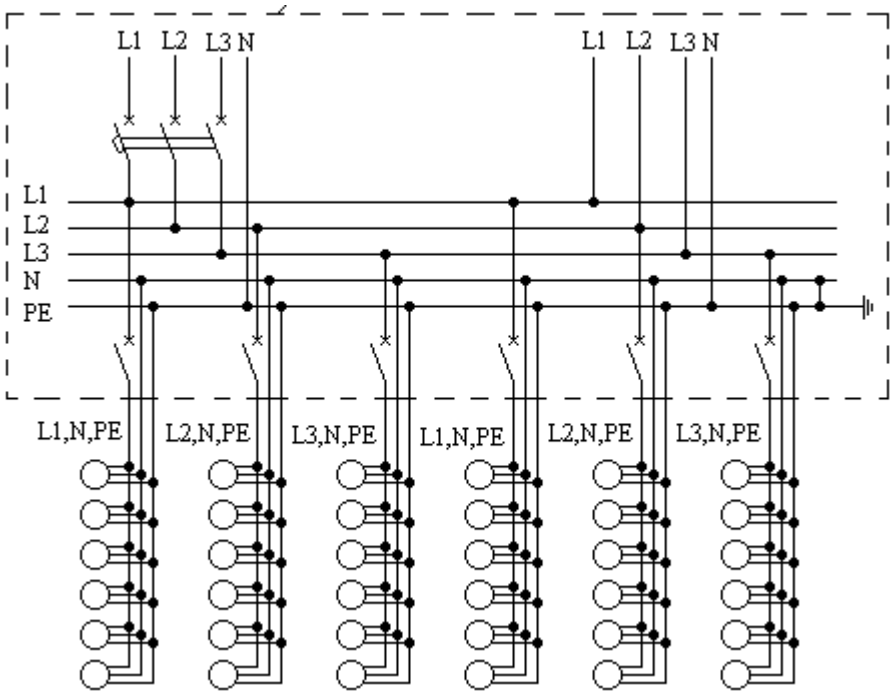


Рисунок 3.2 – Принципова схема групової мережі освітлення для цеха

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.

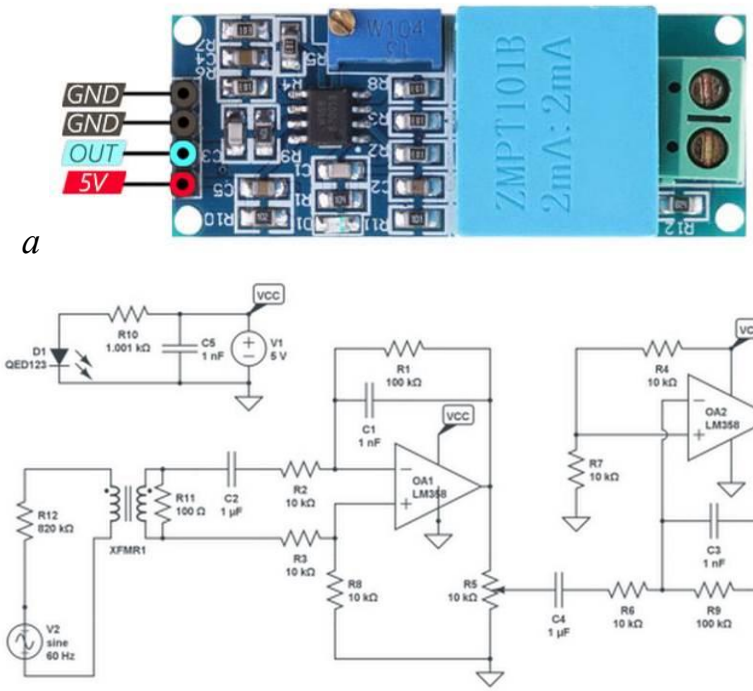
Для технічного обліку електроенергії, що споживається освітленням, застосовують реєстратори або аналізатори параметрів електромережі. Зокрема, ТОВ «НОВАТЕК-ЕЛЕКТРО» (м. Одеса) виготовляє реєстратор РПМ-416, компанія Socomec випускає аналізатор Diris A40 (рис. 3.3).

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 давачем напруги ZMPT101B

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



а

б

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

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

#include < ZMPT101B.h>

Для створення об’єктної змінної давача, що підключений до плати Arduino, використовується команда:

ZMPT101B voltageSensorA(A0);

де ZMPT101B – назва бібліотеки;

voltageSensorA – об’єктна змінна, що відповідає давачеві;

A0 – аналоговий вхід, до якого підключено давач.

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

Ua = voltageSensorA.getVoltageAC();

Бібліотека ACS712.h для роботи з давачем струму ACS712

Вимірювання миттєвих значень струму здійснюється з використанням давача ACS712, який функціонує на основі ефекту Холла. Номінальні струми таких давачів становлять 5А, 20А, 30А, рис. 3.5.

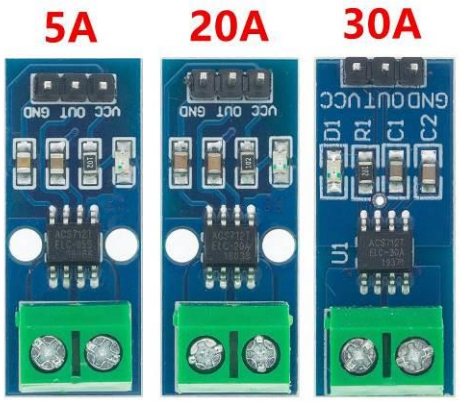


Рисунок 5 – Давачі струму ACS712

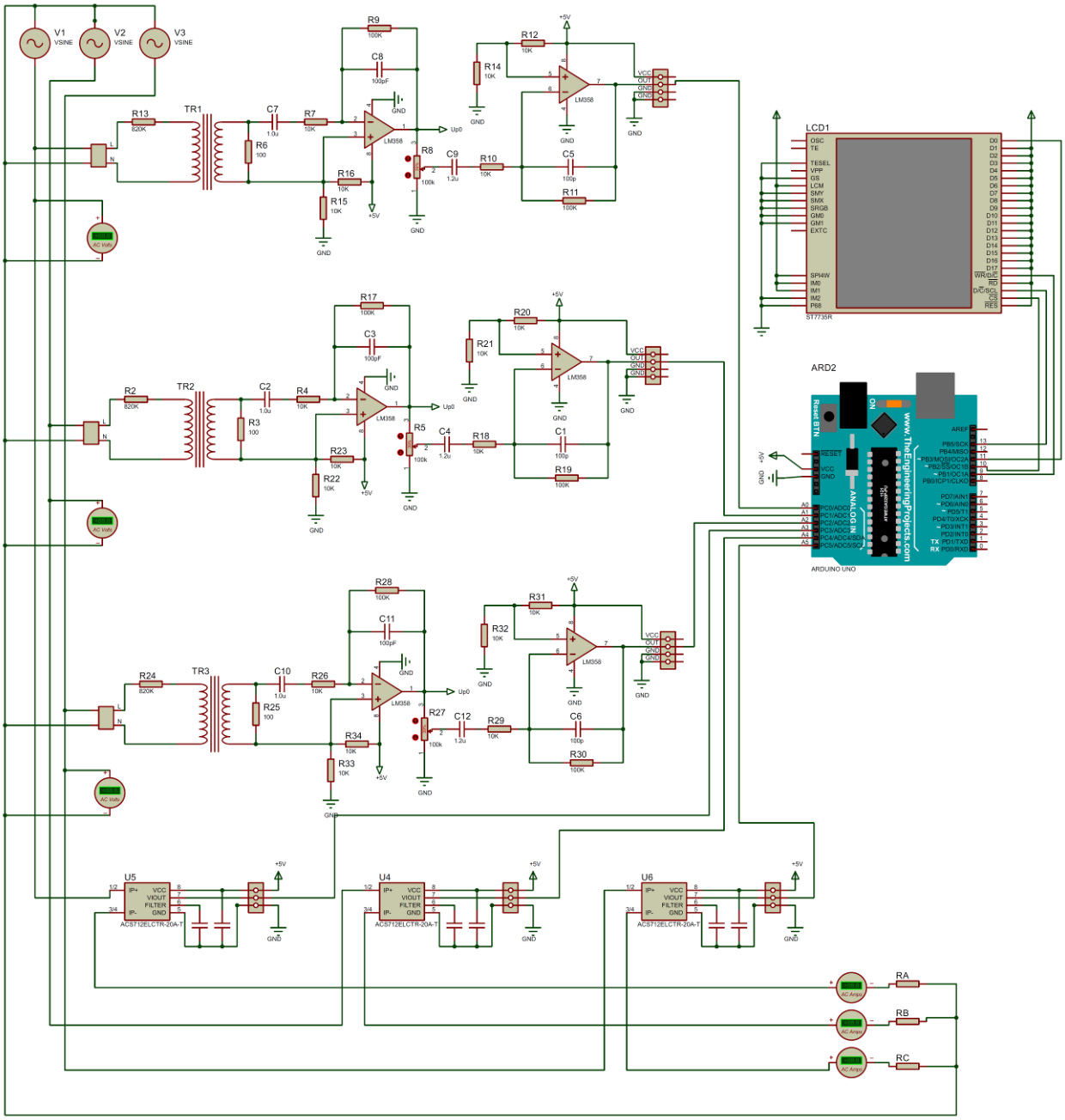


Рисунок 3.6 - Модель пристрою технічного обліку електроенергії в Proteus

Підключення бібліотеки ACS712.h, що використовується для роботи з давачами, здійснюється директивою:

#include <ACS712.h>

Створення програмної змінної, яка відповідає давачеві з номінальним струмом 20 А, здійснюється командою:

ACS712 currentSensorA(ACS712_20A, A3);

Для мережі промислової частоти обчислення середньоквадратичного значення струму здійснюється за допомогою методу:

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 давача фазних напруг, три давача струму, мікроконтролерну плату ArduinoUNO та графічний дисплей ST7735R.

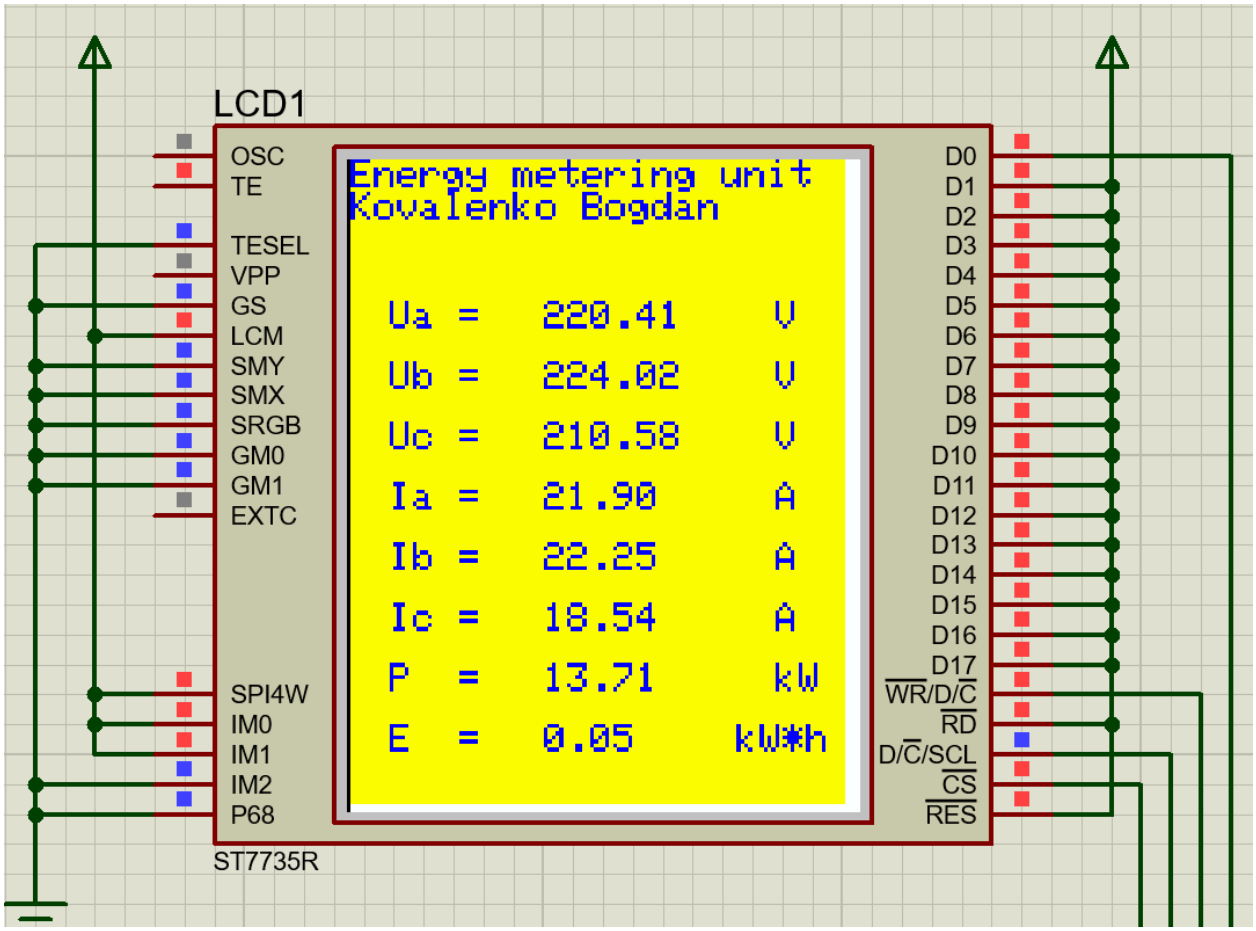


Рисунок 3.7 – Результат роботи програми

ЗАВДАННЯ

Написати програму для пристрою технічного обліку електроенергії на основі мікроконтролера, рис. 3.6. Результат роботи програми має відповідати рис. 3.7.

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. Скласти текст програми для плати ArduinoUNO.

4. Скомпілювати програми (Скетч–Експорт скомпільованого бінарника).

5. В Proteus-моделі мікроконтролера вказати шлях до отриманого файлу …ino.standard.hex. 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. Поясність, яким чином у Вашій програмі розраховувалася активна енергія?

4. Поясніть принцип дії давача струму типу ACS712.

5. Розкрити сутність ефекту Холла?

6. Розкрийте принцип роботи давача напруги ZMPT101B.

7. Яким чином обчислюється середньоквадратичне значення напруги за визначений часовий інтервал за наявності миттєвих значень напруги?

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

Структуру мікропроцесорного струмового захисту ілюструє схема на рис. 4.1. Імпеданси фаз навантаження позначені Z1, Z2, Z3. Напругу живлення на навантаження подає вимикач Q1. В якості первинних вимірювальних перетворювачів використовуються трансформатори струму ТА1–ТА3. Вторинні вимірювальні перетворювачі позначені ПС1–ПС3. Перетворення аналогових сигналів, що відповідають струмам фаз, у цифрову форму виконує інтегроване до мікроконтролера АЦП. Для відображення інформації використовується TFT дисплей. Підсилювач ПС4 використовується для посилення сигналу, що подається на відключення Q1.

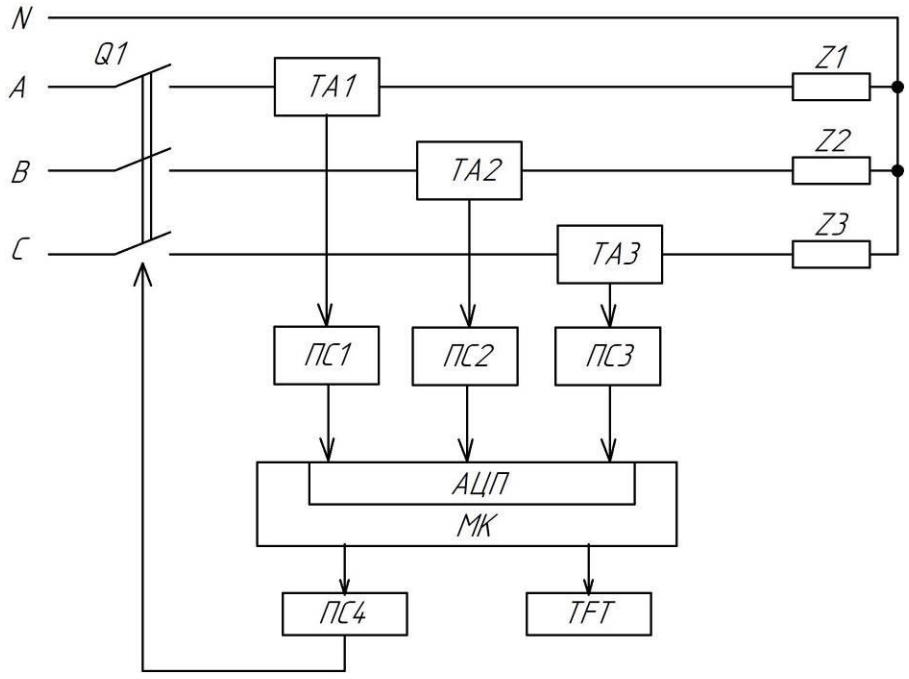


Рисунок 4.1 – Структурна схема мікропроцесорного струмового захисту

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 виміряне 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:

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

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.

Блок схема, що відображає алгоритм роботи пристрою РЗ, наведена на рис. 4.3. Робота пристрою розпочинається з ініціалізації дисплея, яка визначається командами блока 2. Після цього задаються значення нижньої I_L та верхньої I_H уставок (блок 3). Блок 4 об'єднує вимірювання миттєвих значень струмів фаз та обчислення на їх основі середньоквадратичних величин. Блоки 5, 6, 7 забезпечують обчислення цифрових ознак x_A, x_B, x_C. Блок 8 забезпечує обрахування величини інтегральної ознаки. Блоки 9, 11, 13, 15, 17, 19, 21, 23, 25, 27 здійснюють ідентифікацію типу аварійного стану. В разі виявлення аварійного стану, блоки 10, 12, 14, 16, 18, 20, 22, 24, 26, 28 видають на дисплей відповідні повідомлення. Блок 30 забезпечує видачу сигналу на відключення силового вимикача. У випадку нормального режиму функціонування споживача, відображається повідомлення про нормальний режим роботи (блок 29).

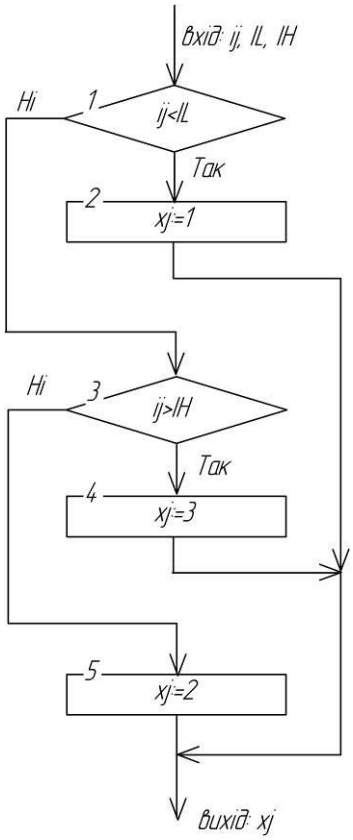


Рисунок 4.2 – Блок-схема підпрограми визначення цифрової ознаки стану *j*-ї фази

Лабораторна робота виконується з використанням моделі пристрою РЗ, яка наведена на рис. 4.4. До складу моделі входить трифазне джерело V3PHASE напругою 380В. Силовий вимикач моделюється реле К1–К3. Навантаження подається елементами R11, L11, R21, L21, R31, L31. 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.

ЗАВДАННЯ

Скласти програму для пристрою РЗ, яка здійснює виявлення аварійного режиму з перелічених: однофазне замикання на землю; обрив фази; двофазне КЗ; трифазне КЗ.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

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

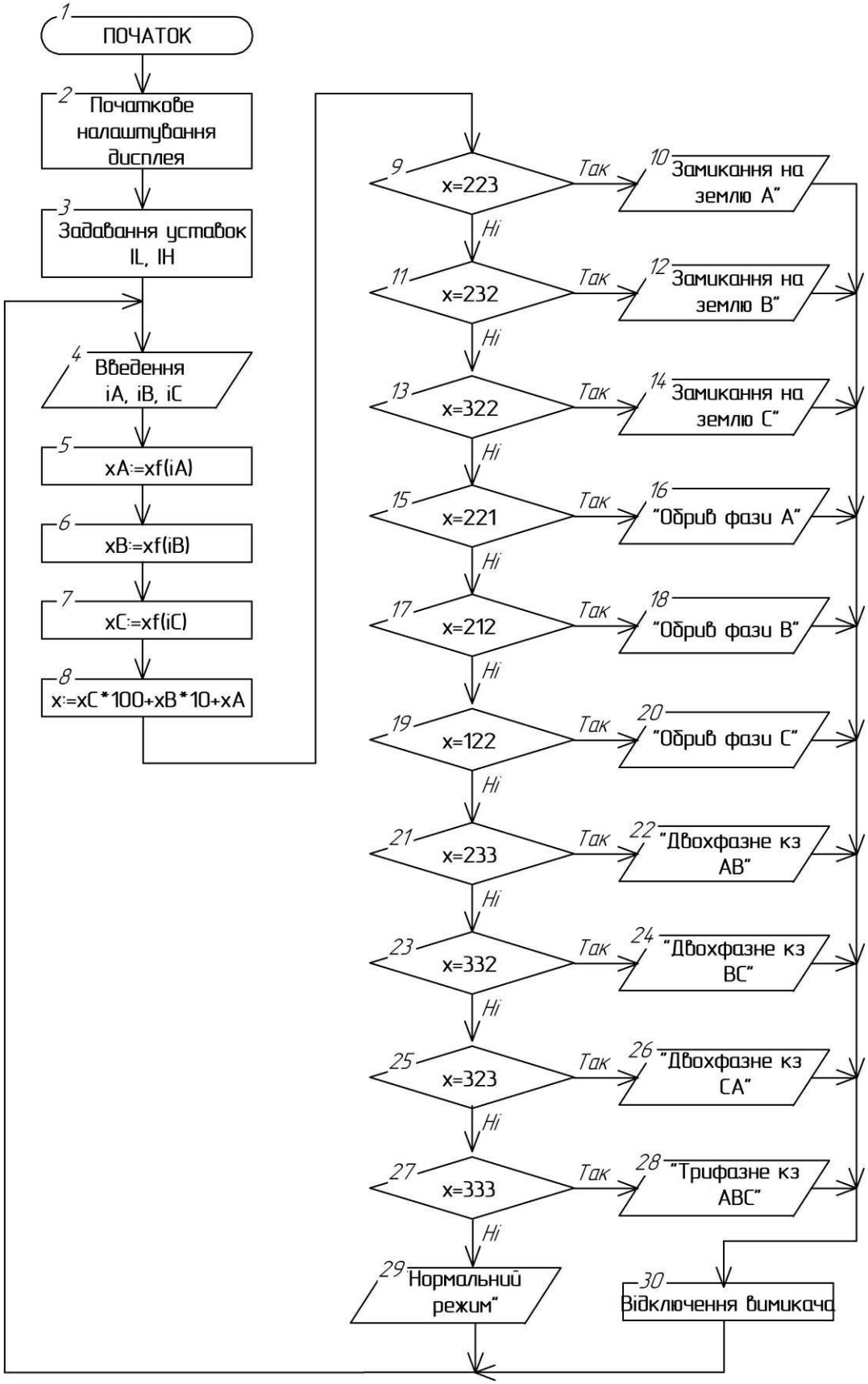


Рисунок 4.3 – Блок-схема, що відповідає алгоритму роботи пристрою

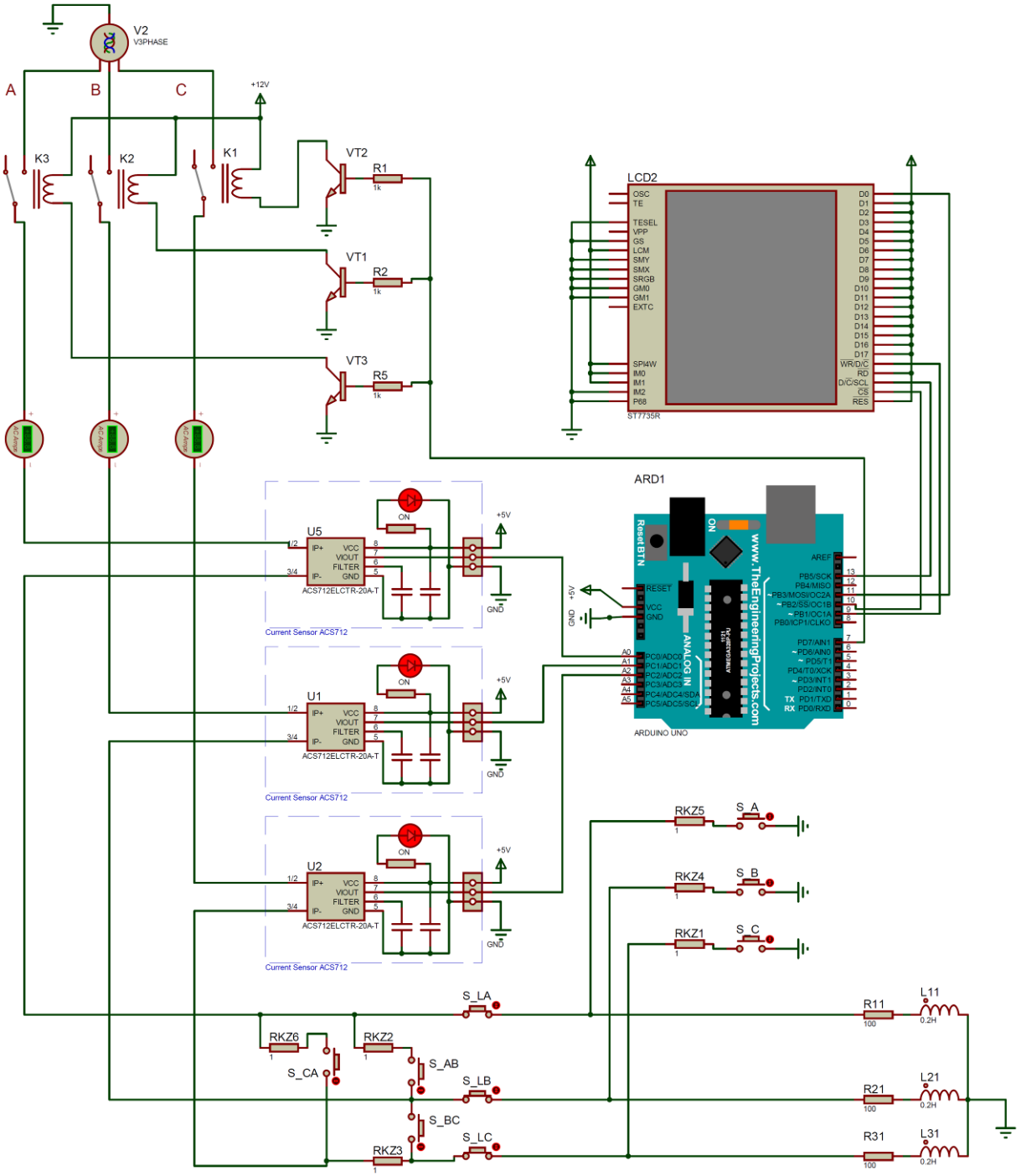
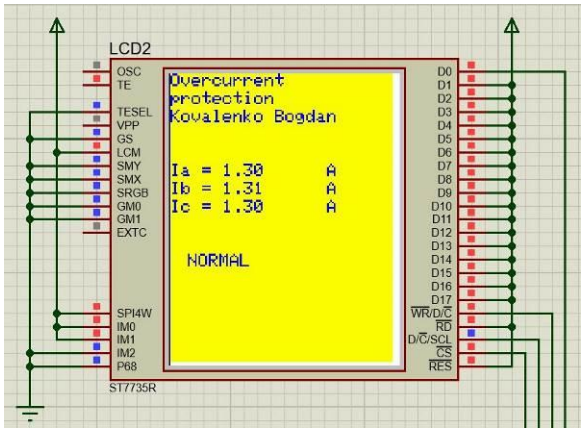
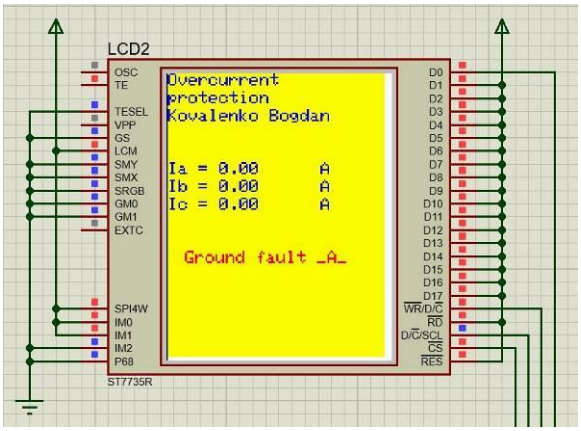


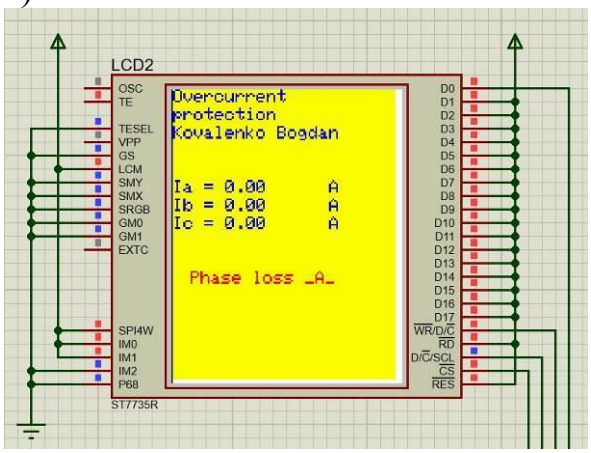
Рисунок 4.4 – Proteus-модель пристрою струмового захисту



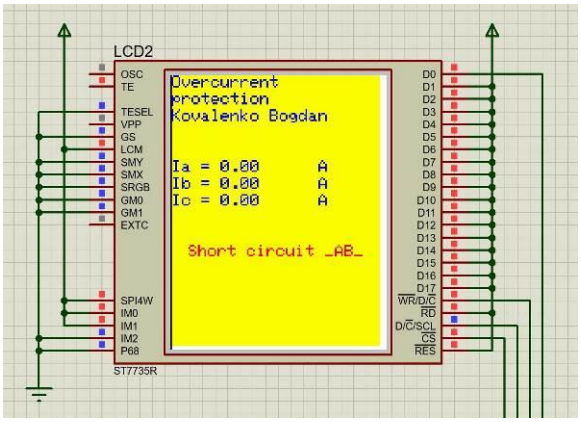
а)



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

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 скомпільованої 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. Які переваги трифазної системи перед системами з іншою кількістю фаз.

5. Наведіть характерині особливості струмовому захисту.

6. Назвіть види струмових захистів?

7. Які особливості функціонування МСЗ?

8. Які особливості функціонування СВ?

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

Для захисту ЛЕП високої напруги від коротких замикань використовується дистанційний. Такий захист характеризується відносною селективністю. В основу функціонування захисту покладено вимірювання імпедансу лінії. Імпеданс в місці встановлення захисту може бути обрахований за векторами струму $\dot{I}_{норм}$ та напруги $\dot{U}_{норм}$:

$$\underline{Z} = \frac{\dot{U}_{норм}}{\dot{I}_{норм}}. \tag{5.1}$$

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

$$\underline{Z}_{кз} = \frac{\dot{U}_{кз}}{\dot{I}_{кз}} < \underline{Z}_{н}. \tag{5.2}$$

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

$$\underline{Z}_{кз} = \underline{Z}_0 \cdot l_{кз}, \tag{5.3}$$

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.

В більшості випадків дистанційний захист виконується триступеневим. Для кожного ступеня встановлено визначений час спрацювання. Ступень I охоплює біля 85% від довжини лінії, що контролюється, він не має затримки часу спрацювання. Час його дії t^1 включає власний час спрацювання та час відключення вимикача, рис. 5.1.

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

Вимірювання повного опору лінії виконує реле мінімального опору. Повний опір є комплексним числом $Z = R + jX$, дійсна складова якого відповідає активному опору, уявна – реактивному. Тому уставка спрацювання не може бути задана одним числом.

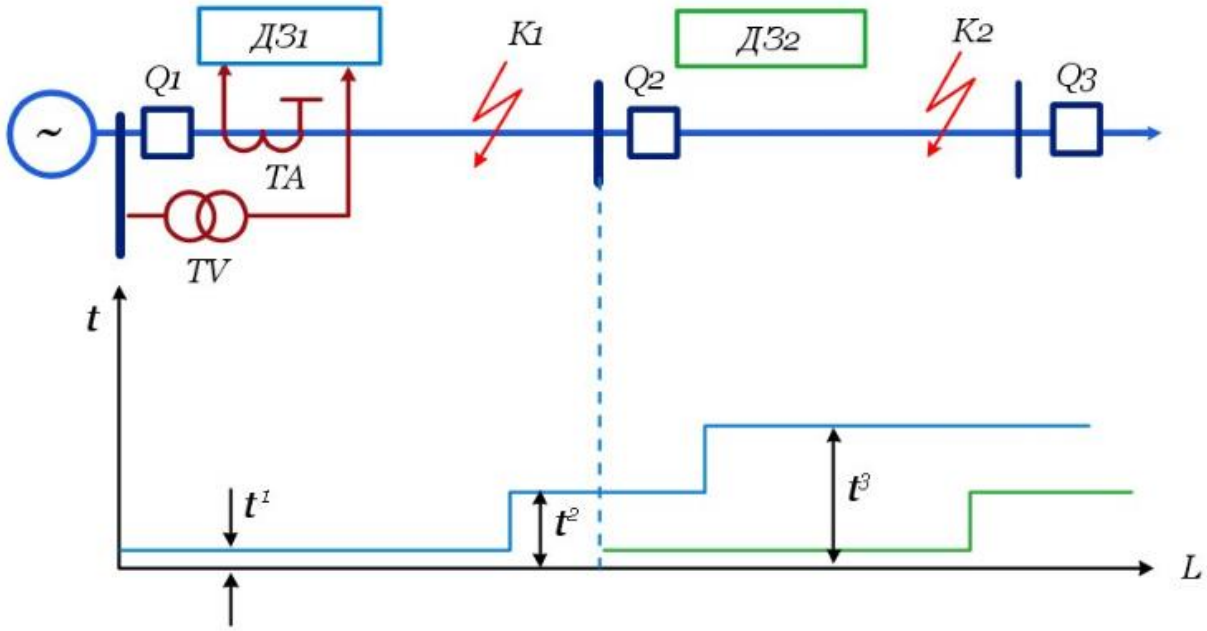


Рисунок 5.1 – Принцип дії дистанційного захисту

Спрацювання реле характеризується кривою (характеристикою) на комплексній площині. Типи характеристик спрацювання бувають різними, проте більшість з них задається модулем імпедансу спрацювання $\left|\underline{Z}_{cp}\right|$ та кутом максимальної чутливості реле $\varphi_{мч}$. При визначенні значень таких характеристик спрацювання необхідно забезпечити відведення їх від зміни навантаження, пусків двигунів, виникнення 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 спрацьовує. Реле, що мають такі характеристики, називають реле повного опору, оскільки спрацювання цих реле не залежить від кута φ_p між струмом та напругою електромережі.

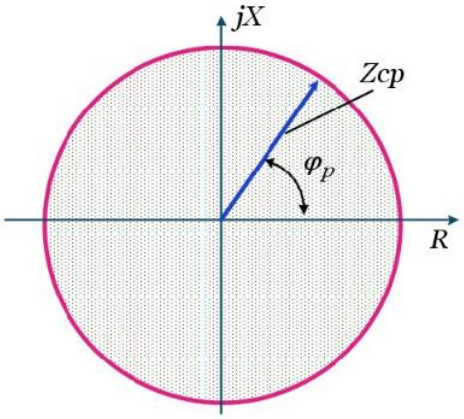


Рисунок 5.2 – Колова характеристика з центром у початку координат спрацювання реле мінімального опору

Також в якості характеристики спрацювання реле мінімального опору може бути використано коло, що проходить через початок координат, рис. 5.3. Таке несиметричне відносно початку координат розташування характерстики визначає спрямований характер захисту – реле реагуватиме тільки на виникнення замикання в одному напрямку, що відповідає додатному значенню дійсної та уявної частин імпедансу.

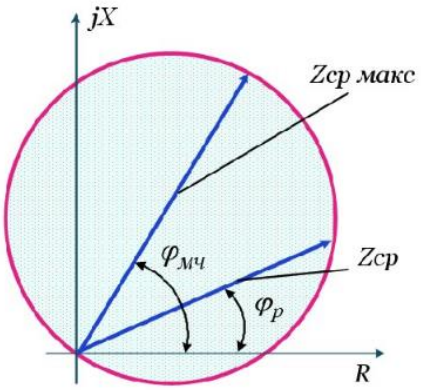


Рисунок 5.3 – Колова характеристика, що проходить через початок координат, спрацювання реле мінімального опору

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.

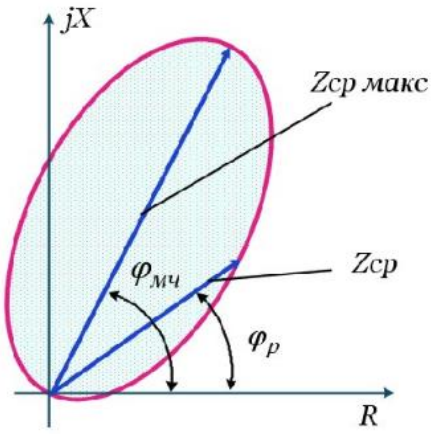


Рисунок 5.4 – Еліптична характеристика спрацювання реле мінімального опору

Для налаштування реле третього ступеня захисту може бути використана характеристика трикутної форми, рис. 5.5, б. Це дозволяє відвести захист від навантажувальних режимів, а також надати захисту необхідної чутливості.

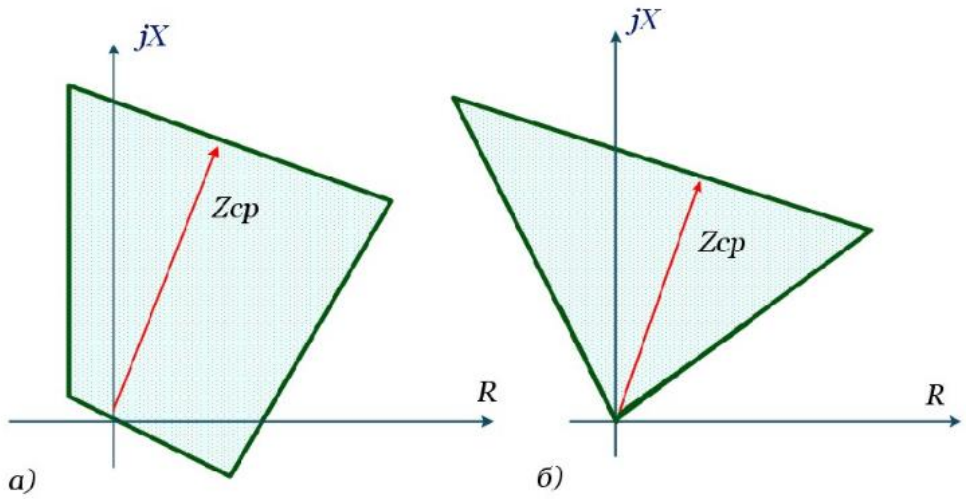


Рисунок 5.5 – Багатокутна характеристика спрацювання реле мінімального опору

Відома велика кількість мікропроцесорних терміналів, що реалізують алгоритм роботи дистанційного релейного захисту. Наприклад, захист SIPROTEC 7SA610, рис. 5.6.

Рисунок 5.6 – Мікропроцесорний термінал SIPROTEC 7SA610

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

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.

В якості первинного вимірювального перетворювача напруги $u_A(t)$ використовується вимірювальний трансформатор TV, наприклад, серії НКФ-123-II (НКФ-110-II). Рис. 5.9 ілюструє приєднання трьох однофазних ТН вказаного типу за допомогою трифазного роз'єднувача із ножами заземлення. Захист від грозових перенапруг здійснюється вентильними розрядниками.

Сигнал $u_{TV}(t)$ з виходу вимірювального ТН надходить до блоку БПСН, який здійснює лінійне перетворення сигналів за напругою. Блок здійснює підготовку сигналу до перетворення у АЦП1. З виходу АЦП1 надходять дискретизовані значення $u(nT)$, які є результатом дискретизації миттєвих значеннь напруги електромережі в моменти часу $t = 0, T, 2T, 3T, ...$, де період дискретизації позначено T.

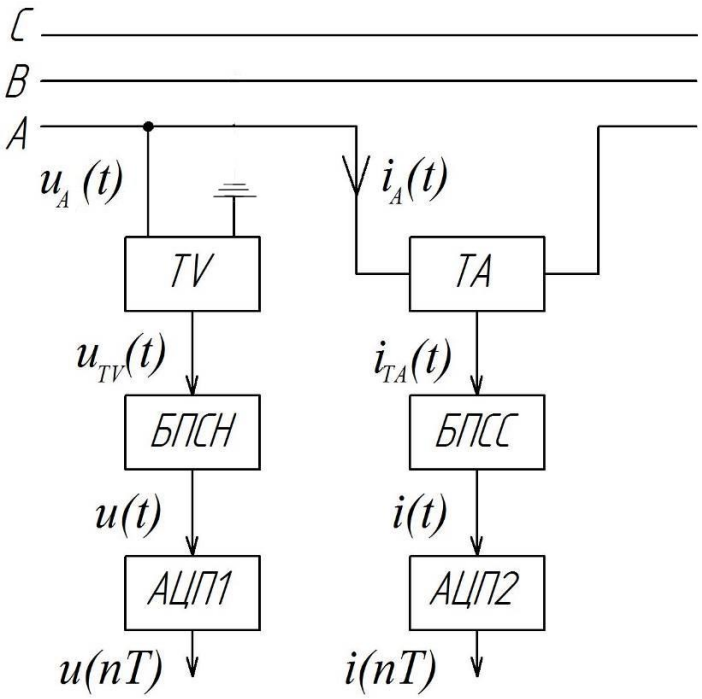


Рисунок 5.8 – Структурна схема вимірювальних каналів напруги і струму у складі навчального реле мінімального опору

Рисунок 5.9 – Розташування вимірювальних трансформаторів напруги на підстанції 110 кВ

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 виміряними миттєвими значеннями можна обчислити наступним чином:

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

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

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

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

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

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

Проекції обертових векторів можна знайти зі співвідношення:

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

Тоді проекції нерухомого (загальмованого) вектора напруги дорівнюють:

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

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

Для вектора струму аналогічно одержуємо:

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

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

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

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:

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

Індуктивний опір лінії дорівнює:

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

Для захисту, що проектується, задамо колову характеристику спрацювання реле, рис. 5.11.

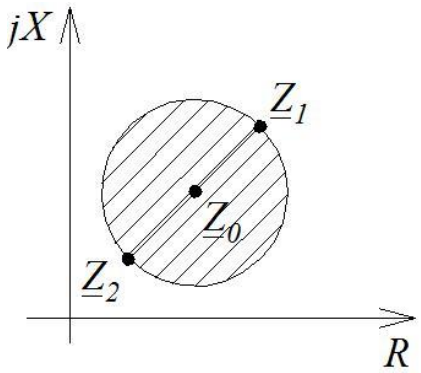


Рисунок 5.11 – Загальний вигляд колової характеристики спрацювання реле опору

Опорними для такої характеристики є дві точки $\underline{Z}_1 = R_1 + jX_1$ та $\underline{Z}_2 = R_2 + jX_2$. Тоді В реле спрацює при виконанні умови:

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

де позначено:

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

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

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.

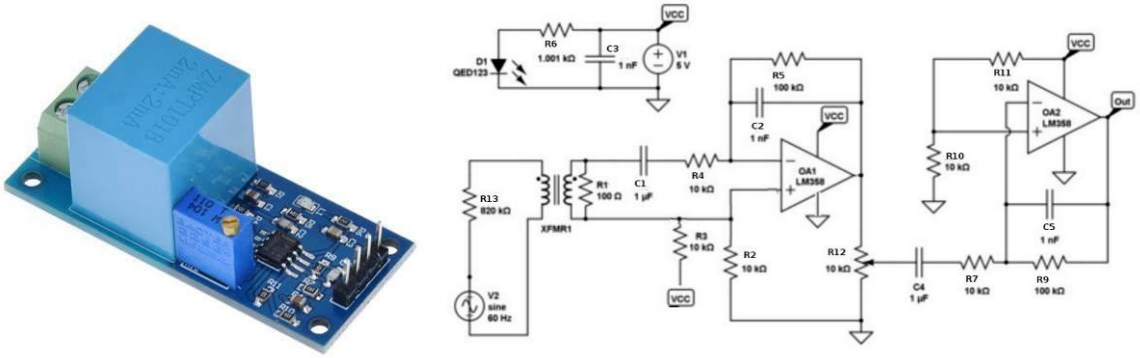


Рисунок 5.12 – Перетворювач напруги ZMPT101B

В якості вторинного вимірювального перетворювача струмових кіл передбачається використовувати перетворювач ACS712, що зображений на рис. 5.13.

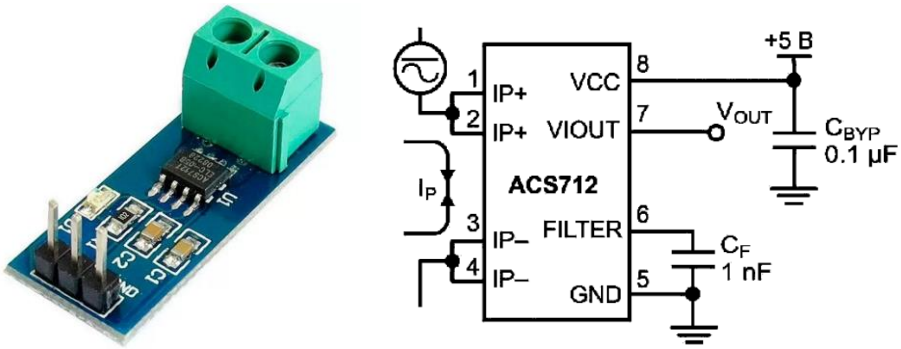


Рисунок 5.13 – Перетворювач струму в напругу ACS712

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

Лабораторна робота виконується з використанням навчальної моделі мікропроцесорного реле мінімального опору в Proteus, що наведена на рис. 5.14. До складу такої спрощеної моделі входить джерело синусоїдної напруги

VSINE, причому амплітуда становить 310 В, а частота дорівнює 50 Гц). До джерела приєднано опір ЛЕП в нормальному режимі (R=18 Ом, L=0,01 Гн). Для врахування опору кола КЗ використовується резистор Rkz.

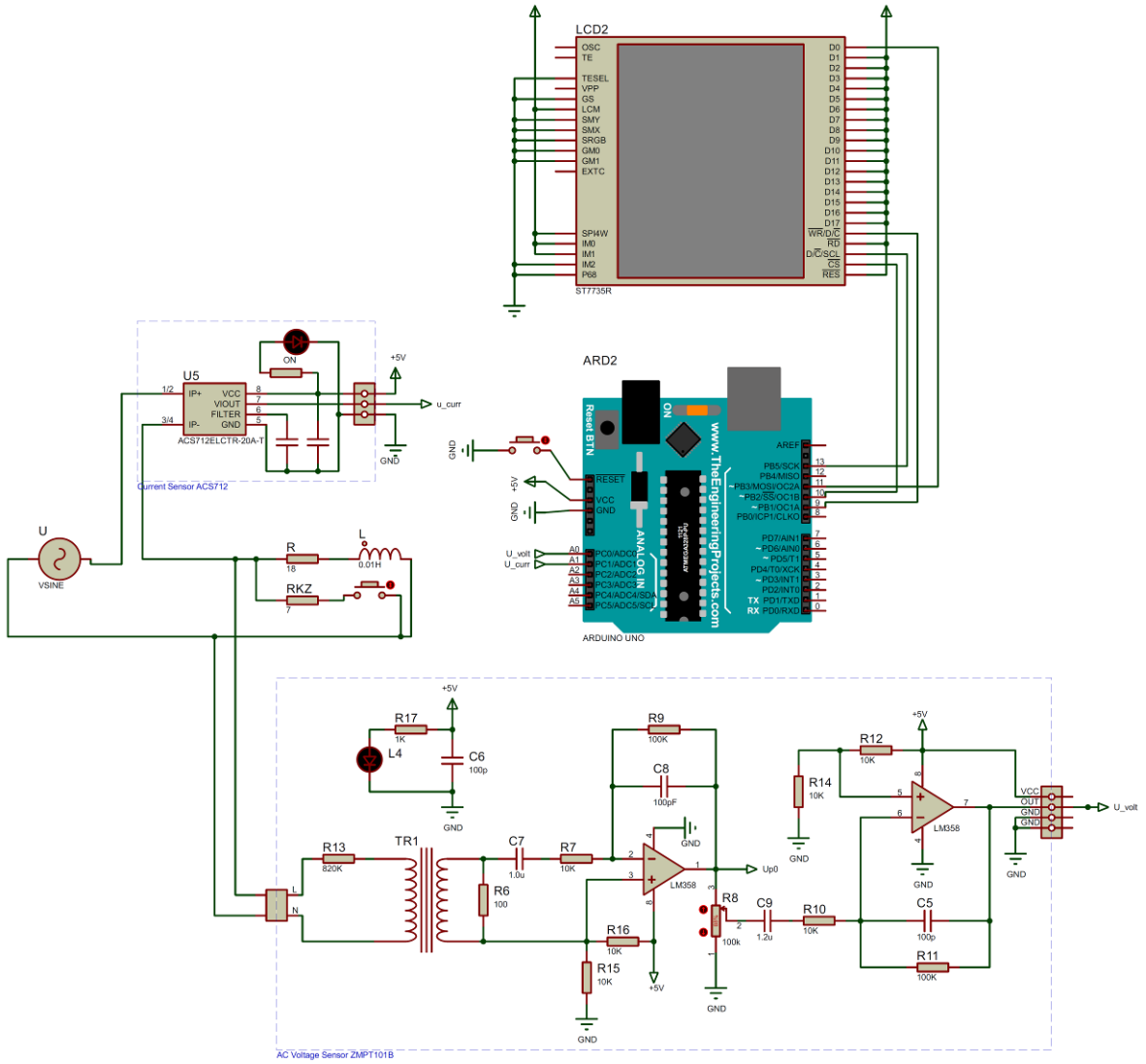
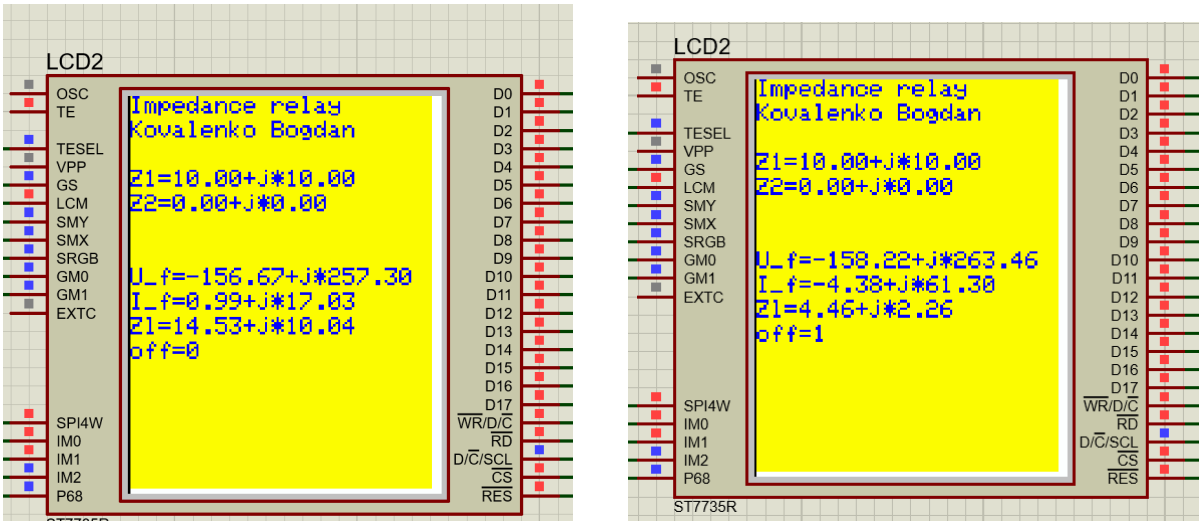


Рисунок 5.14 - Навчальна модель мікропроцесорного реле мінімального опору

Рис. 5.15 подає результати функціонування пристрою РЗ. На дисплеї відображаються величини характерних точок характеристики спрацювання (Z1, Z2), комплексна величина Uf нерухомого вектора напруги, а також величина If нерухомого вектора струму. Також на дисплей виводиться виміряний повний опір Z*l* лінії і значення сигналу off, що подається на вимкнення вимикача при КЗ («0» – вимикач працює; «1» – відключити вимикач).

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.



а) б)

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

Таблиця 5.1

Вихідні дані

Варіант	Опорні точки характеристики спрацювання				Параметри моделі реле		
	$\underline{Z}_1 = R_1 + jX_1$		$\underline{Z}_2 = R_2 + jX_2$		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). Таке значення часу приймається початковим для інтервалу, за який визначається положення векторів. Для цього використовується функція:

t1=micros();

1.5 Для того, аби виміряти миттєві значення напруги та струму (з одночасним лінійним перетворенням), використовуються команди:

u1=1.04*(analogRead(0)-512);

i1=0.049*(analogRead(1)-512);

Орієнтовна тривалість кожного з вимірювань – 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 виміряній 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 виміряній 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 Відобразити на дисплеї обраховані значення згідно з 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.

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.

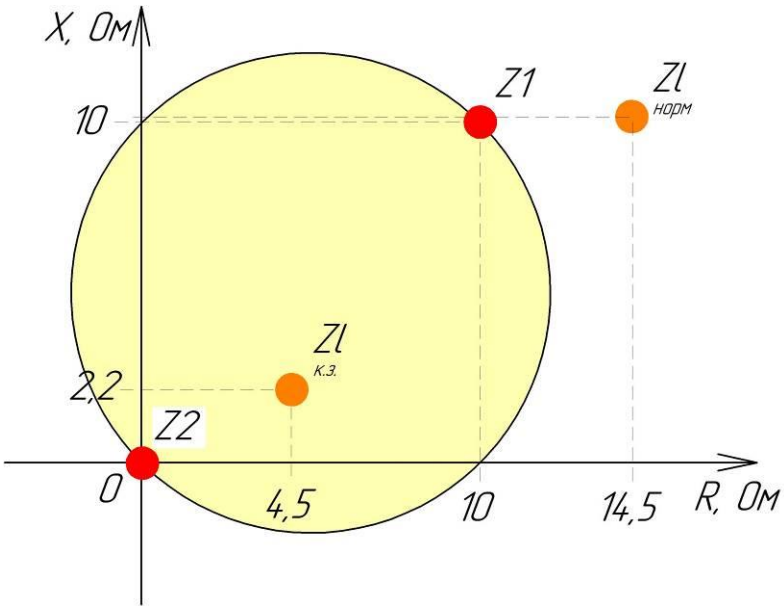


Рисунок 5.16 – Характеристика спрацювання реле мінімального опору та фактичні імпеданси кола в нормальному режимі та при к.з. для параметрів відповідно до рис. 15

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: дискртизовані 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. 361 с.

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.