

**ФАКУЛЬТЕТ АВТОМАТИКИ, ТЕЛЕМЕХАНІКИ ТА ЗВ'ЯЗКУ**

**Кафедра спеціалізованих комп'ютерних систем**

**МЕТОДИЧНІ ВКАЗІВКИ**

**до виконання лабораторних робіт з дисципліни**

***“АВТОМАТИЗАЦІЯ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ”  
SCHNIDER ELECTRIC***

**Харків – 2015**

Методичні вказівки розглянуто і рекомендовано до друку на засіданні кафедри спеціалізованих комп'ютерних систем 16 лютого 2015 р., протокол № 8.

Призначено для студентів факультету АТЗ з напрямку КІУС – бакалавр 6.050202 денної форми навчання. Можливе застосування цих методичних вказівок і на інших формах навчання та напрямках підготовки студентів і магістрантів за рішенням лектора.

Укладач

доц. В.М. Бутенко

Рецензент

проф. В.І. Мойсеєнко

## МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт з дисципліни

*“АВТОМАТИЗАЦІЯ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ”  
SCHNIDER ELECTRIC*

Відповідальний за випуск Бутенко В.М.

Редактор Ібрагімова Н.В.

---

Підписано до друку 16.03.15 р.

Формат паперу 60x84 1/16. Папір писальний.

Умовн.-друк.арк. 4,50. Тираж 60. Замовлення №

Видавець та виготовлювач Українська державна академія залізничного транспорту,

61050, Харків-50, майдан Фейербаха, 7.

Свідоцтво суб'єкта видавничої справи ДК № 2874 від 12.06.2007 р.

**УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ**

**Факультет автоматики, телемеханіки і зв'язку**

**Кафедра спеціалізованих комп'ютерних систем**

**МЕТОДИЧНІ ВКАЗІВКИ**  
**для виконання лабораторних робіт з дисципліни**  
**«Автоматизація технологічних процесів» Schneider Electric**  
**для студентів з напрямку КІУС – бакалавр 6.050202 денної**  
**форми навчання**

**Харків 2015**

Методичні вказівки розглянуто і рекомендовано до друку на засіданні кафедри "Спеціалізовані комп'ютерні системи" 16 лютого 2015 р., протокол № 8.

Призначено для студентів факультету АТЗ з напрямку КІУС – бакалавр 6.050202 денної форми навчання. Можливе застосування цих методичних вказівок і на інших формах навчання та напрямках підготовки студентів і магістрантів за рішенням лектора.

Укладач

доц. В.М. Бутенко

Рецензент

проф. В.І. Мойсеєнко

## Вступ

Метою цих методичних вказівок є вивчення практичних аспектів імітаційного моделювання технологічних процесів на прикладі програмно-технічних компонентів залізничної автоматики як складової частини систем забезпечення руху поїздів засобами промислових контролерів. Для створення програмних модулів спеціалізованих комп'ютерних систем використовується множина мов програмування, яка відповідає ІЕС 31131-3/EN 31131. Курс розглядається як галузева складова і передбачає ознайомлення з відповідними методиками створення і дослідження моделей і підпрограм у відповідності з робочою програмою дисципліни, затвердженою радою факультету.

Побудова та дослідження моделей і програмного забезпечення при розробленні нових складних технічних систем часто складають значну частину в загальній сукупності робіт, що проводяться на початкових фазах проекту. Модель являє собою систему теоретичних посилок і практичних методик організації машинних обчислень, що дозволяють відобразити в кількісних оцінках і характеристиках динаміку функціонування досліджуваного об'єкта, який існує або проектується.

Дані методичні вказівки допомагають отримати вміння та навички первинного програмування промислових контролерів для автоматизації технологічних об'єктів.

Для виконання лабораторних робіт у лабораторіях кафедри СКС академії встановлено програмне забезпечення (ПЗ) Unity PRO XL v 3.0, надане фірмою Шнайдер Електрик Україна (<http://www.schneider-electric.ua>), що функціонує в навчально-демонстраційному режимі. Для самостійної роботи студентів можливе застосування будь-якої версії легального ПЗ Unity PRO. Однак у випадку невідповідності версії на студентському та лабораторному ПК саме на студента покладається відповідальність за демонстрацію самостійно розроблених програмних модулів при виконанні лабораторної роботи. Практика показує, що краще або застосовувати версію 3.0, або приносити переносний ПК для демонстрації самостійно розроблених програм автоматизації.

# Лабораторна робота 1

## ОЗНАЙОМЛЕННЯ ТА РОБОТА З МЕК, СТВОРЕННЯ ПРОЕКТІВ АВТОМАТИЗАЦІЇ

**Мета роботи:** вивчити структуру та особливості міжнародного стандарту електротехнічної комісії та програмування промислових контролерів.

**Обладнання та ПЗ:** ПЕОМ з необхідними системними вимогами для роботи ПЗ, MS Windows, інтернет Explorer, Open Office Writer, програми Unity PRO.

Unity PRO – це програмне середовище конфігурації, програмування, налагодження і діагностування виконавчої системи та промислових контролерів; є результатом розвитку двох програмних продуктів PL7 PRO – середовище програмування першої групи контролерів і Concept – середовище програмування другої групи контролерів.

Виконавча система Unity – це програмне забезпечення, яке виконується в контролері. Виконавча система базується на ОС Unity, яка вже знаходиться в завантаженні ПЛК та бере участь у всіх операціях контролера.

Головна задача Unity – розроблення виконавчого проекту, який виконується в контролері.

### Послідовність виконання роботи

- 1 Знайти додаткові джерела інформації.
- 2 Вивчити теоретичний і практичний матеріал, необхідний для виконання роботи.
- 3 Відповісти на контрольні питання.
- 4 Зобразити в зошиті програму мовою релейно-контактною логіки за варіантом студента за журналом викладача в підгрупі:

- 1)  $y_1 = x_1 \cap x_2 \cap x_3 \cap (\bar{x}_4 \cup \bar{x}_5) \cap y_{19}, y_{19} = x_6 \cup \bar{x}_7$  ;
- 2)  $y_2 = \bar{x}_1 \cap x_2 \cap \bar{x}_3 \cap (\bar{x}_4 \cup \bar{x}_5) \cap y_{19}, y_{19} = \bar{x}_6 \cup \bar{x}_7$  ;
- 3)  $y_3 = \bar{x}_1 \cap \bar{x}_2 \cap \bar{x}_3 \cap (x_4 \cup \bar{x}_5) \cup y_{19}, y_{19} = \bar{x}_6 \cup x_7 \cap x_9$  ;
- 4)  $y_4 = \bar{x}_1 \cap x_2 \cap \bar{x}_3 \cap \bar{x}_4 \cap x_5 \cup y_{100}, y_{100} = \bar{x}_6 \cup x_7$  ;
- 5)  $y_5 = x_1 \cap x_2 \cap x_3 \cap (\bar{x}_4 \cup \bar{x}_5) \cap y_{19}, y_{19} = x_8 \cup \bar{x}_7$  ;

- 6)  $y_6 = x_1 \cap x_2 \cap x_9 \cap (\bar{x}_4 \cup \bar{x}_6) \cap y_{19}, y_{19} = x_8 \cup \bar{x}_7$  ;
- 7)  $y_7 = x_9 \cap x_2 \cap (\bar{x}_4 \cup \bar{x}_5) \cap y_{19}, y_{19} = x_8 \cup \bar{x}_7 \cup \bar{x}_3$  ;
- 8)  $y_8 = x_9 \cap x_2 \cap x_3 \cap (\bar{x}_4 \cup \bar{x}_5) \cap y_{19}, y_{19} = x_8 \cup \bar{x}_7 \cup \bar{x}_1$  ;
- 9)  $y_9 = x_1 \cap x_9 \cap x_3 \cap (\bar{x}_4 \cup \bar{x}_5) \cap y_{19}, y_{19} = x_8 \cup \bar{x}_7 \cup \bar{x}_2$  ;
- 10)  $y_A = x_1 \cap x_2 \cap x_9 \cap (\bar{x}_4 \cup \bar{x}_3) \cap y_{19}, y_{19} = x_8 \cup \bar{x}_7 \cup \bar{x}_5$  ;
- 11)  $y_B = x_1 \cap x_2 \cap x_3 \cap (\bar{x}_9 \cup \bar{x}_5) \cap y_{19}, y_{19} = x_8 \cup \bar{x}_9 \cup \bar{x}_7$  ;
- 12)  $y_C = x_9 \cap \bar{x}_2 \cap x_3 \cap (\bar{x}_4 \cup \bar{x}_5) \cap y_{19}, y_{19} = x_8 \cup \bar{x}_7 \cup \bar{x}_1$  ;
- 13)  $y_D = \bar{x}_1 \cap \bar{x}_2 \cap x_3 \cap (\bar{x}_4 \cup \bar{x}_5) \cap y_{19}, y_{19} = x_8 \cup \bar{x}_7 \cup \bar{x}_9$  ;
- 14)  $y_E = x_1 \cap \bar{x}_2 \cap x_3 \cap (\bar{x}_4 \cup \bar{x}_5) \cap y_{19}, y_{19} = x_8 \cup \bar{x}_7 \cup \bar{x}_9$  ;
- 15)  $y_F = x_1 \cap \bar{x}_2 \cap x_3 \cap (\bar{x}_4 \cup \bar{x}_9) \cap y_{19}, y_{19} = x_8 \cup \bar{x}_7 \cup \bar{x}_5$  .

### Контрольні питання:

- які етапи «еволюції» ПЗ промислових контролерів?
- як скачати програму Unity PRO?
- як встановити ліцензійну (легальну) програму Unity PRO?
- у який каталог встановилась програма?
- у яких форматах зберігаються файли ПЗ?
- які умови розповсюдження програми?
- яка структура програмованого логічного контролера (ПЛК) Premium TSX (Modicon, Quantum, Momentum)?
  - що таке проект?
  - які види мов програмування ви знаєте?
  - що таке трійник?
  - які особливості програмного замітника трійника?
  - запишіть приклади різних типів булевих змінних;
  - у якої мови графічний інтерфейс?
  - що таке збереження проекту?
  - назвіть засоби, що дозволяють сконфігурувати проект;
  - що таке DIN-рейка?
  - який абсолютний шифр у входу (вхідної локалізованої змінної)?
    - які типи процесорних блоків ви знаєте?
    - що таке блок виводу?
    - з яких підсистем складається контролер Шнайдер Електрик?
      - що таке заземлення?
      - які види «обмоток» в мові РКС/LD ви знаєте?
      - що таке налагодження проекту?

- які типи блоків живлення ви знаєте?
  - який абсолютний шифр у виходу (вихідної локалізованої змінної)?
  - які параметри модуля TSX P57 2634M програмованого логічного контролера?
  - які параметри інтерфейсів зв'язку модуля TSX P57 2634M ПЛК?
  - які бувають модулі введення/виведення дискретних каналів?
  - які бувають модулі введення/виведення аналогових каналів?
  - які основні параметри модулів введення/виведення дискретних каналів?
  - які основні параметри модулів введення/виведення аналогів каналів?
  - як створити проект автоматизації в Unity PRO?
- Домашнє завдання для подальшої підготовки до ЛР: знайти ліцензійну (демонстраційну) версію програми Unity PRO та встановити її на переносному ПК студента.
- Прочитати й систематизувати навчальний матеріал всіх рекомендованих джерел інформації.


## **Лабораторна робота 2**

### **АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНУВАННЯ ПЛК ТА СТВОРЕННЯ ПРОЕКТУ**

**Мета роботи:** навчитись створювати проекти, отримати навички конфігурування проекту автоматизації технологічного процесу зі складанням програм та елементів візуалізації.

**Обладнання та ПЗ:** ПЕОМ з необхідними системними вимогами для роботи ПЗ, MS Windows, Internet Explorer, Open Office Writer, програми Unity PRO.

#### **Послідовність виконання роботи**

На робочому столі знайти ярлик програми Unity PRO , запустити. Потім створюємо новий проект (використавши тільки цифри та латинські літери ASCII таблиці) та обираємо



процесорний модуль TSX P57 2634M, вибравши тип шасі (X-Bus) і скомпонувавши ПЛК модулями введення-виведення, за допомогою панелі, яка знаходиться в лівому нижньому вікні (рисунок 1), наприклад аналоговий TSX AEY 414, дискретні TSX DEY 32D2K TSX та DSY 32T2K.

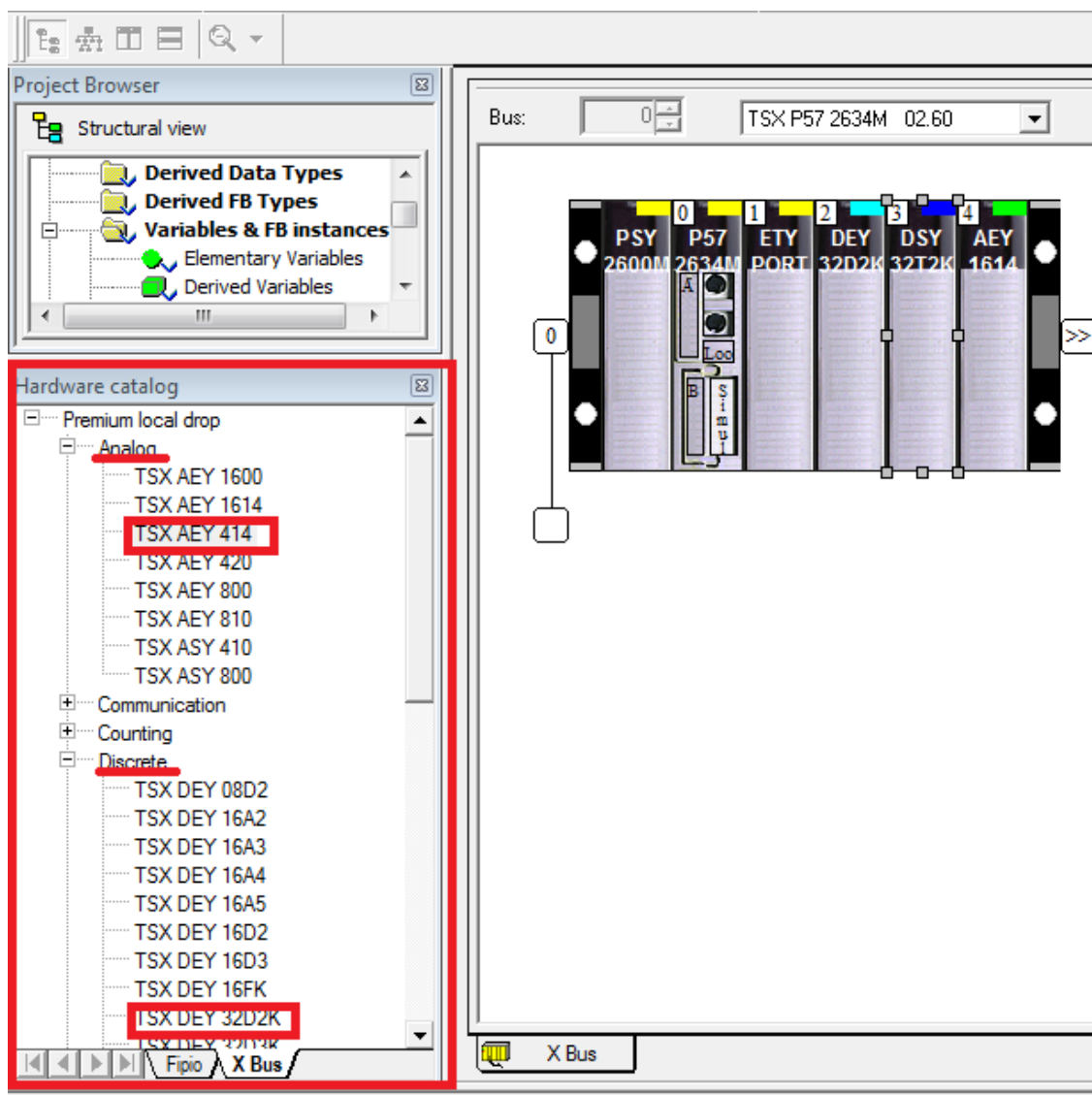


Рисунок 1 – Конфігурування ПЛК в проекті

Після завершення компонування заданих проектом або планом лабораторних робіт (ЛР) модулів (як показано для ЛР на рисунку 2) активізуємо модуль ETY PORT і відкриваємо його для налаштування (рисунок 3). Аналогічно налаштовуються інші модулі ПЛК (за потреби).

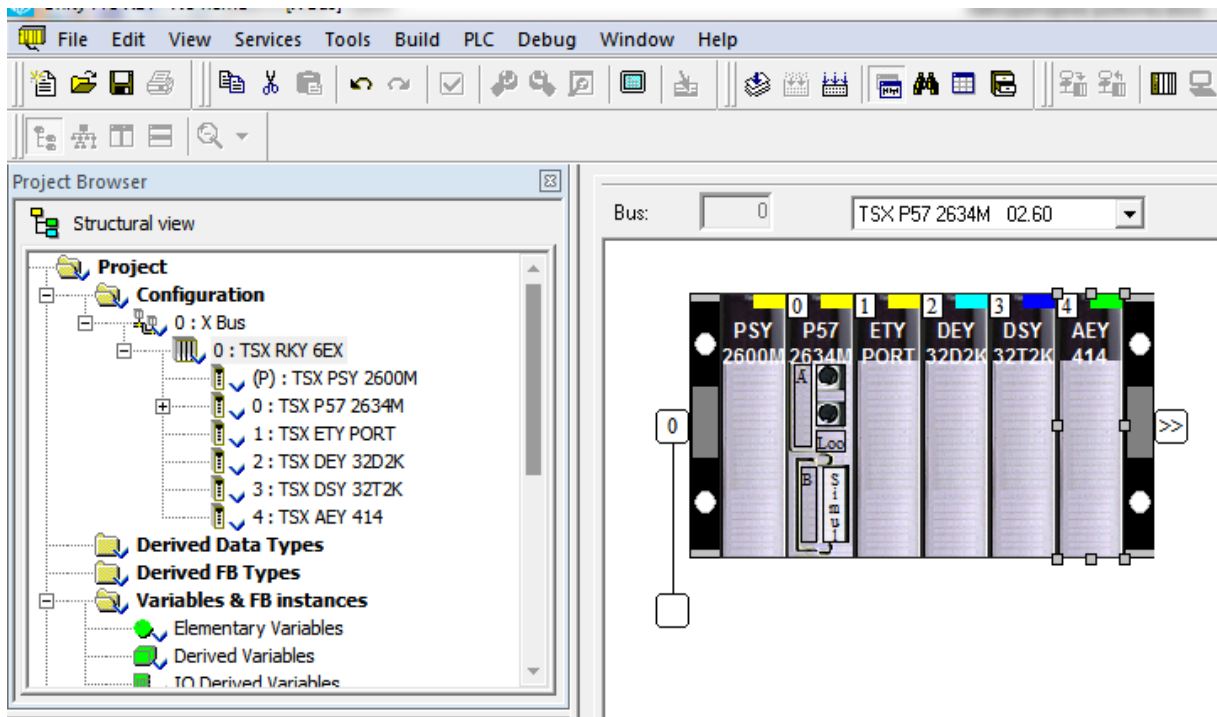


Рисунок 2 – Вибір модуля ПЛК

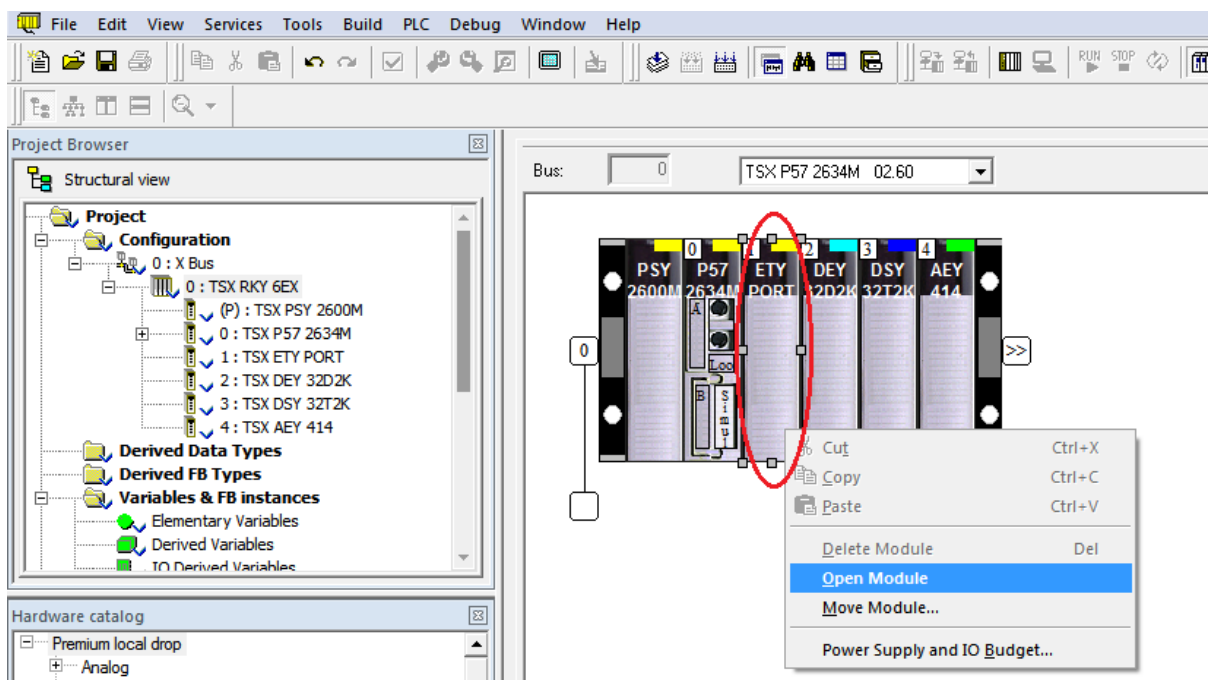
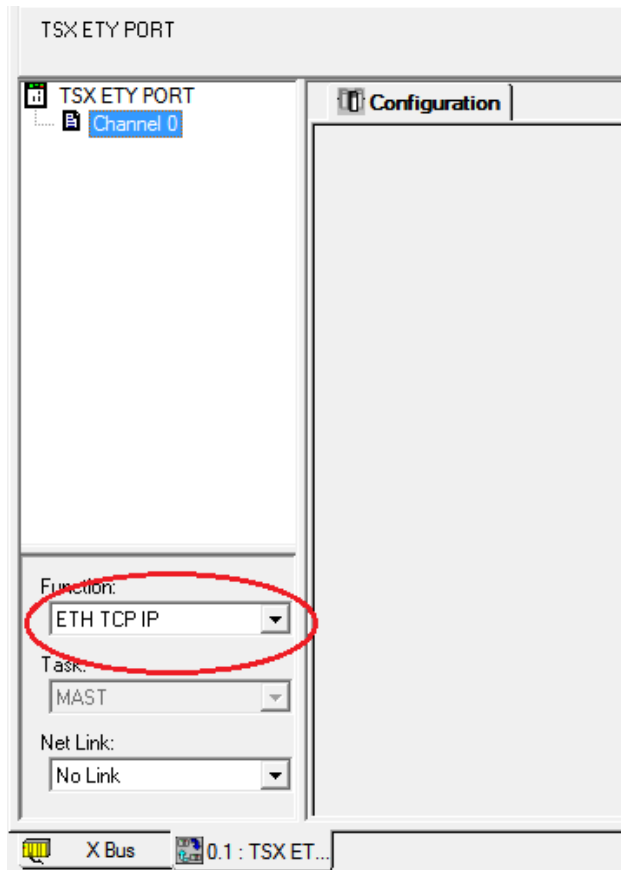
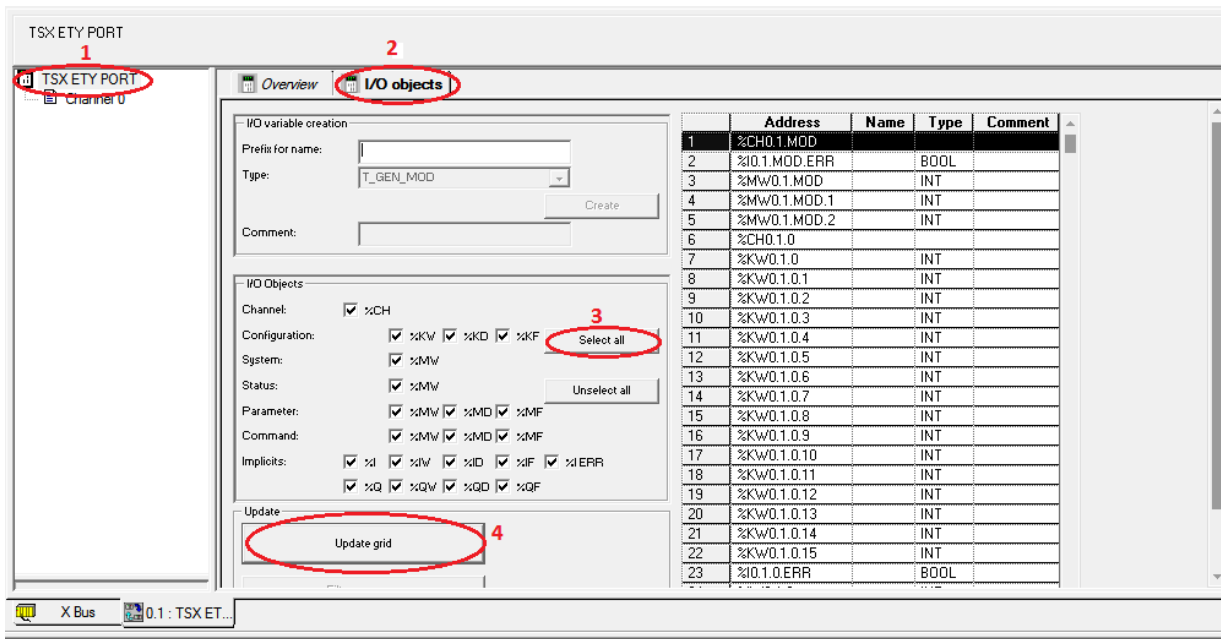


Рисунок 3 – Налаштування модуля TSX ETY PORT

У цьому самому вікні обираємо TSX ETY PORT (рисунок 4, а) (Open Module → Channel 0 → знизу Function) перемикаємо на ETN TCP IP і, зробивши налаштування в закладці I/O objects (рисунок 4, б), зберігаємо налаштування.



а



б

а – вибір протоколу ETH TCP IP;  
 б – налаштування в закладці I/O objects  
 Рисунок 4 – Налаштування модуля TSX ETY PORT

Далі переходимо в меню PLC, як показано на рисунках 5, 6.

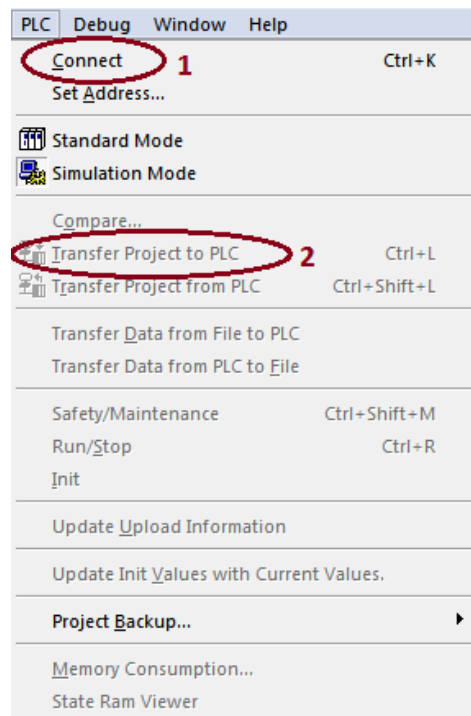


Рисунок 5 – Зв'язок з контролером

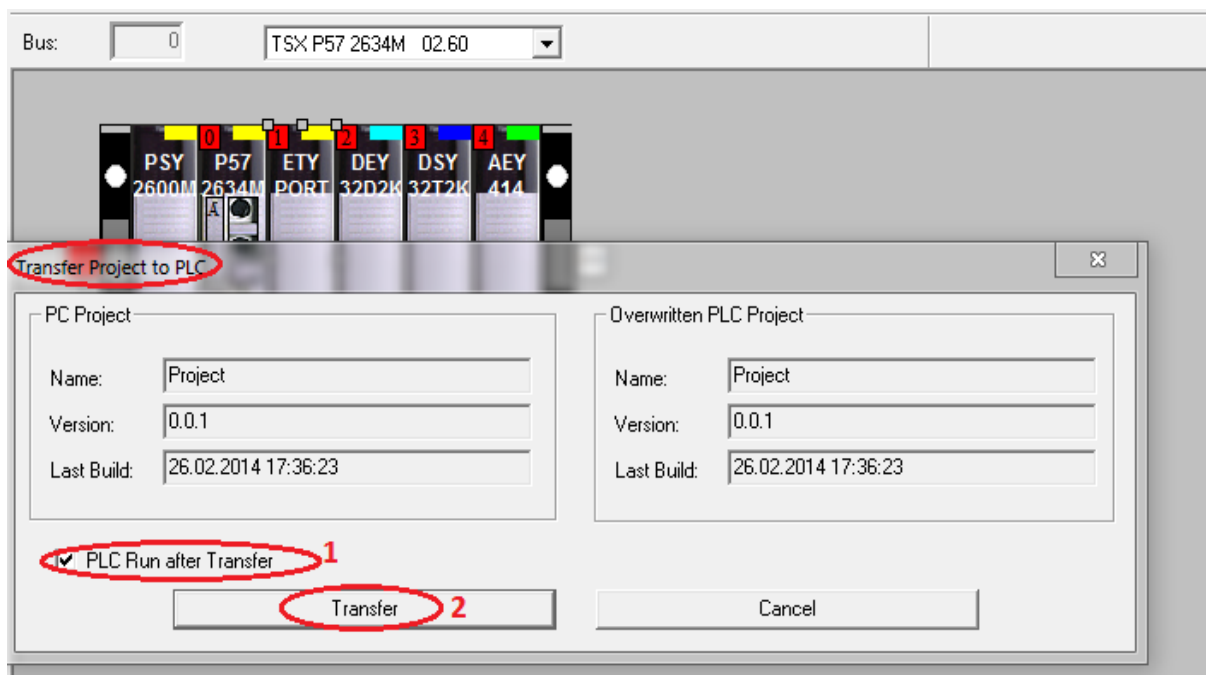


Рисунок 6 – Передача проекту до ПЛК/PLC

У випадку відсутності зв'язку перевірити встановлення опції «Simulation mode», якщо ПЛК не підключений до ПК.

Для того щоб зупинити програму, обираємо меню PLC та натискаємо Stop, потім знову PLC та обираємо Disconnect.

Наступний крок виконання ЛР – на лівій вкладці знайти параметр Program, обрати Tasks, потім MAST, далі Section та правою кнопкою миші викликати контекстне меню й вибрати New Section (рисунок 7).

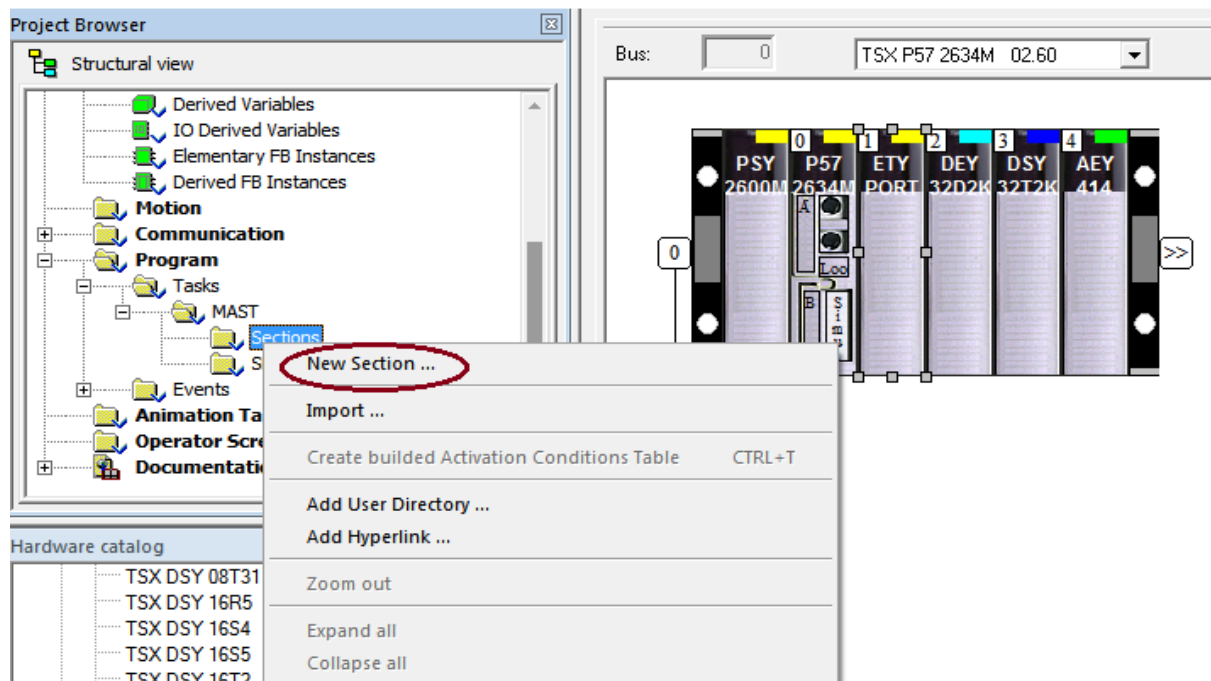


Рисунок 7 – Створення секції програми в групі MAST

Далі вводимо ім'я секції (використовувати тільки латинські літери з можливим додаванням цифр) та обираємо параметр мови програмування (LD) (рисунок 8).

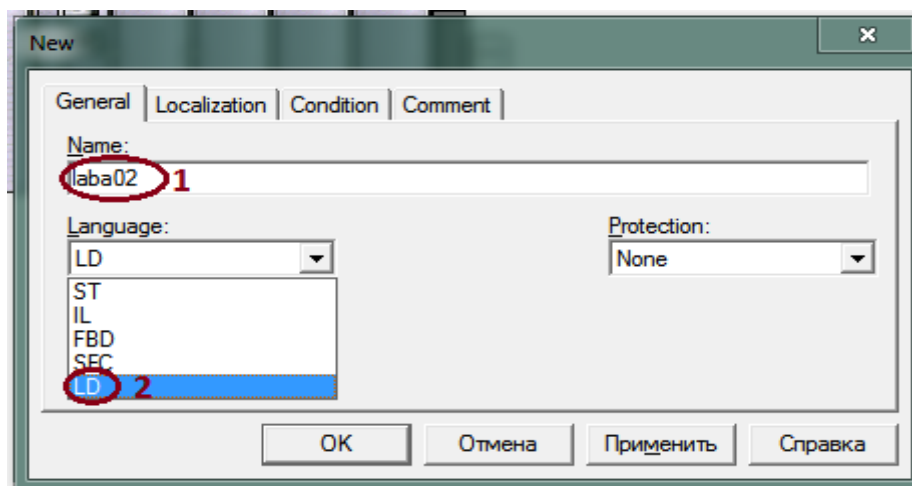


Рисунок 8 – Введення назви секції та вибір мови програмування

Будуємо схему, як показано на рисунку 9, за допомогою кнопок на верхній допоміжній панелі (1).

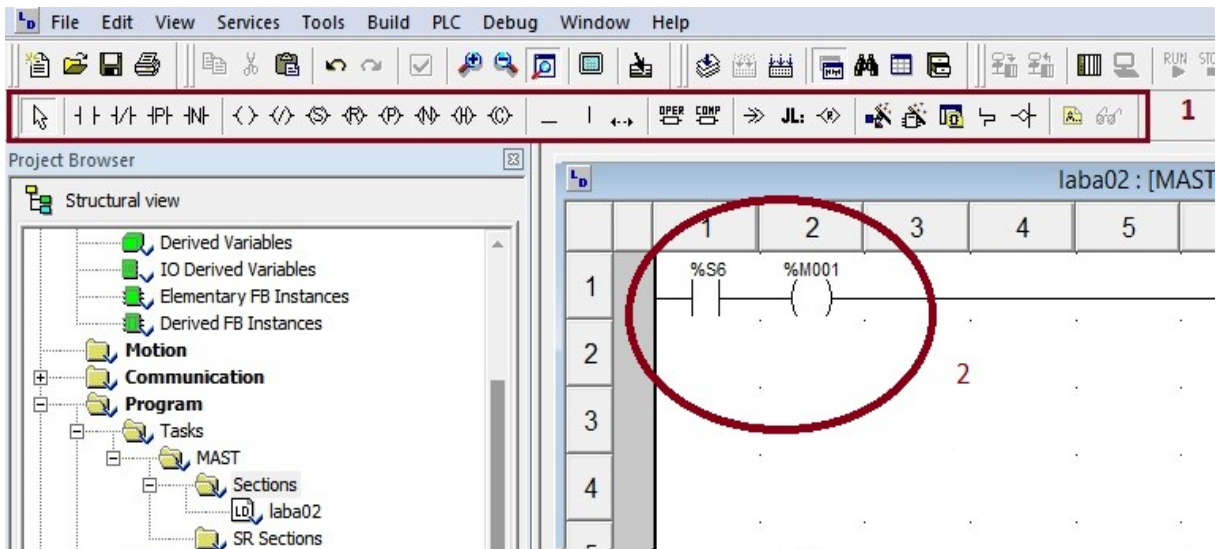


Рисунок 9 – Приклад програми мовою LD

Після компіляції та передачі програми в ПЛК й запуску її там повинно з'явитись зображення обтічних струмами частин схеми/програми мовою LD (рисунок 10).

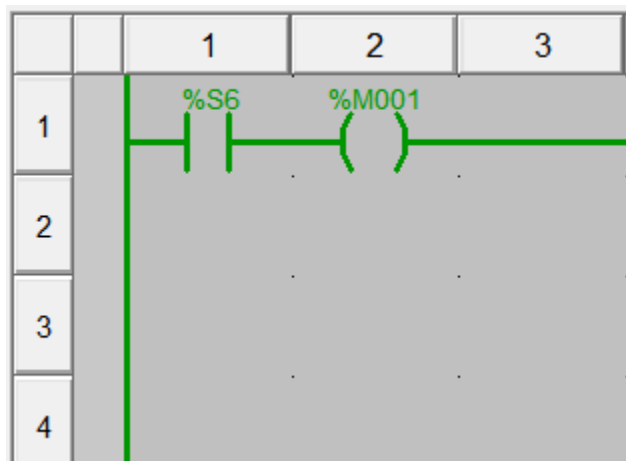


Рисунок 10 – Зображення працюючої програми мовою LD

Далі в лівому вікні (рисунок 11) обираємо «Operation Screen» та правою кнопкою миші викликаємо меню й вибираємо «New Screen».

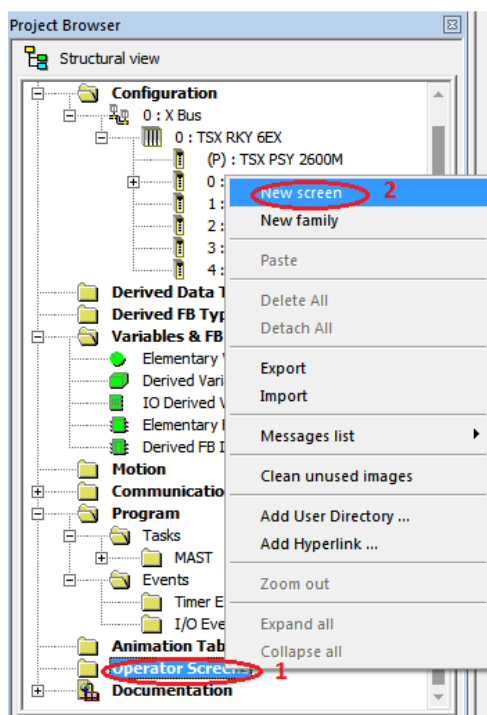


Рисунок 11 – Створення нового елемента екранної візуалізації

У вікні, яке з'явилося, записуємо назву Screen\_laba2 (рисунок 12).

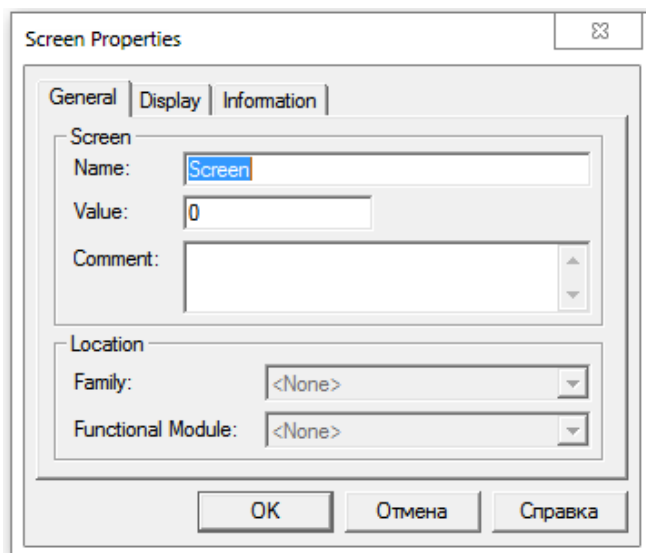


Рисунок 12 – Вікно налаштування візуалізації

Далі в цьому вікні візуалізації за допомогою правої кнопки миші (ПКМ) створюємо новий об'єкт «коло» та, знову натискаючи ПКМ, за допомогою опції Properties задаємо параметри, як на рисунках 13, 14.

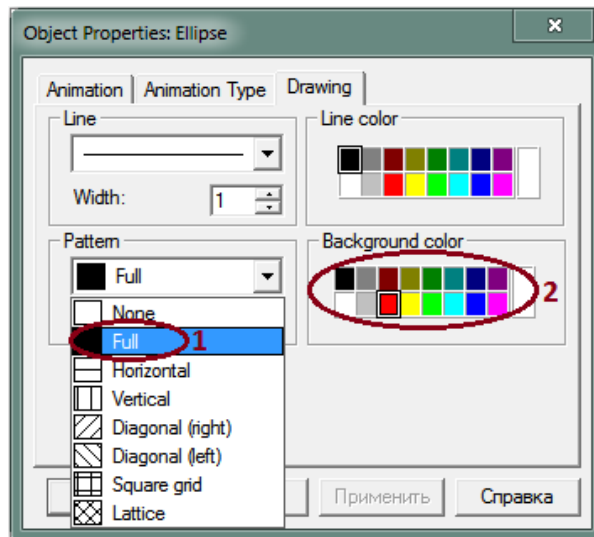


Рисунок 13 – Налаштування форми зображення елемента візуалізації

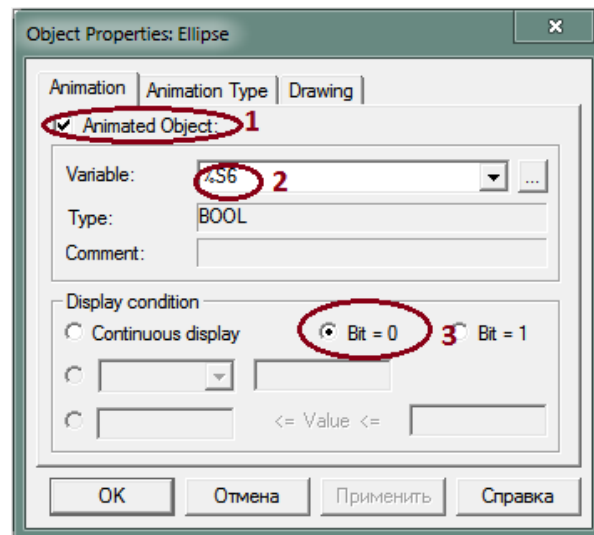


Рисунок 14 – Налаштування форми залежності елемента візуалізації від змінних проекту

Після цього так само створюємо ще один об'єкт «коло», змінюючи тільки колір, і Bit=0 змінюємо на Bit=1.

Запускаємо програму, кола повинні по черзі блимати.

План виконання роботи:

- 1 Знайти всі необхідні додаткові джерела інформації.
- 2 Вивчити теоретичний і практичний матеріал, необхідний для виконання роботи.

3 Відповісти на контрольні питання та розробити план ЛР.

Контрольні питання:



- 1 Як сконфігурувати ПЛК з різними модулями в проекті?
- 2 Яких типів бувають модулі?
- 3 Як активувати модуль EYU PORT?
- 4 Які мови програмування підтримуються Unity PRO?
- 5 У якому вікні можливо зображувати кола?
- 6 Як зробити візуалізацію, залежну від змінних проекту?
- 7 Які розширення мають файли з проектом Unity PRO?

### **Лабораторна робота 3**

## **АПАРАТНО–ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ОБРОБКИ ВХІДНИХ/ВИХІДНИХ ДИСКРЕТНИХ СИГНАЛІВ**

**Мета роботи:** отримати навички конфігурування проекту зі складанням програм введення/виведення дискретних сигналів технологічного процесу та елементів його візуалізації.

**Обладнання та ПЗ:** ПЕОМ з необхідними системними вимогами для роботи ПЗ, MS Windows, Internet Explorer, Open Office Writer, програми Unity PRO.

### **Об'єкт управління для лабораторної роботи**

Для цієї роботи слід викликати проект з попередньої роботи або створити його знову з додаванням програми вмикання семисегментного індикатора (рисунок 15). Кожен з сегментів підключений до виходів модуля DSY 32T2K і через нього будуть виводитись змінні  $y_1 \div y_7$ . А до модуля TSX DEY 32D2K підключені 10 кнопок для зчитування сигналів  $x_0 \div x_9$ . В умові тестової задачі для демонстрації лабораторної роботи необхідно замикає тільки одну кнопку, а на індикаторі має відобразитися цифра від 0 до 9.

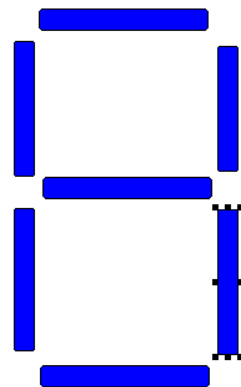


Рисунок 15 – Індикатор

### **Послідовність виконання роботи**

У проекті відкриваємо вікно Variables & FB instances -> Elementary Variables та створюємо 17 змінних типу EBOOL від  $x_0$  до  $x_9$  та від  $y_1$  до  $y_7$  (рисунок 16).

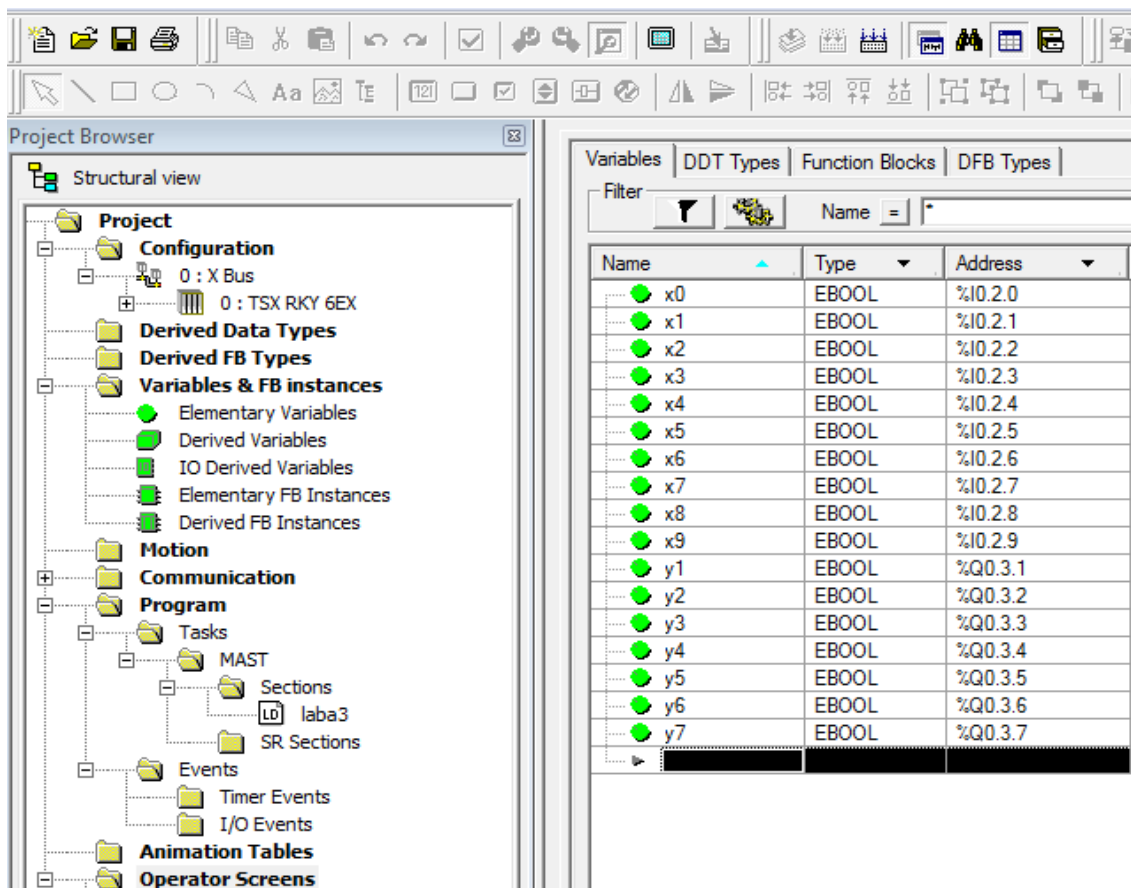


Рисунок 16 – Вхідні та вихідні змінні

Наступний крок – на лівій вкладці знаходимо параметр Program, обираємо Tasks, потім MAST, далі Section та правою кнопкою миші вибираємо New Section (рисунки 8, 17).

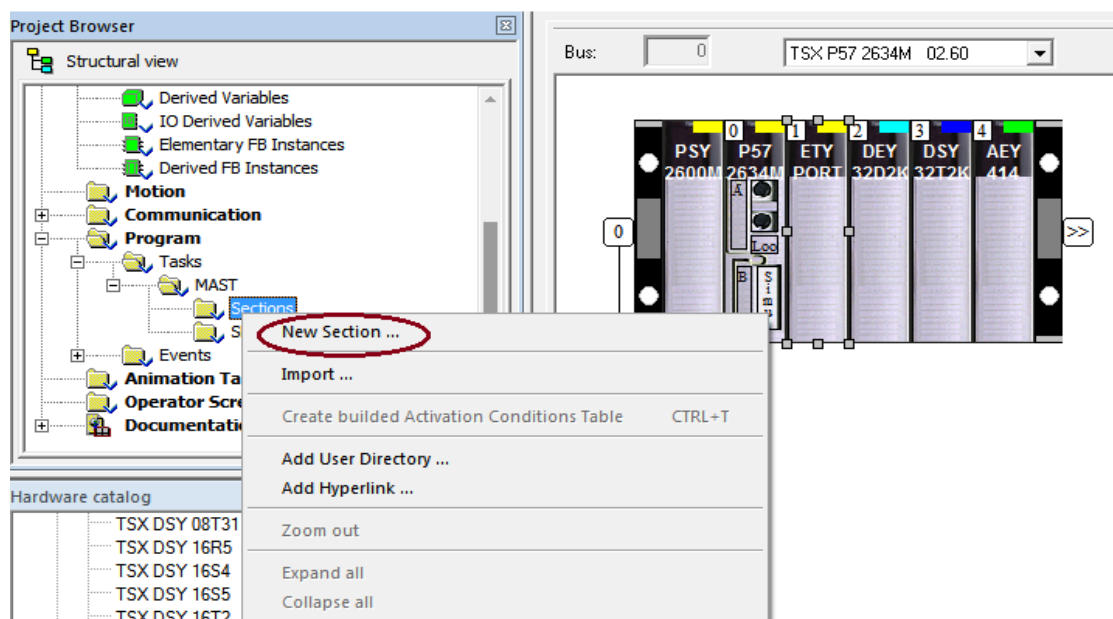


Рисунок 17 – Створення нової секції для ЛРЗ

Створюємо «схеми» для кожної з вихідних змінних  $y_i$ , які залежать від вхідних змінних ( $x_0 \div x_9$ ) (рисунки 18, 19).

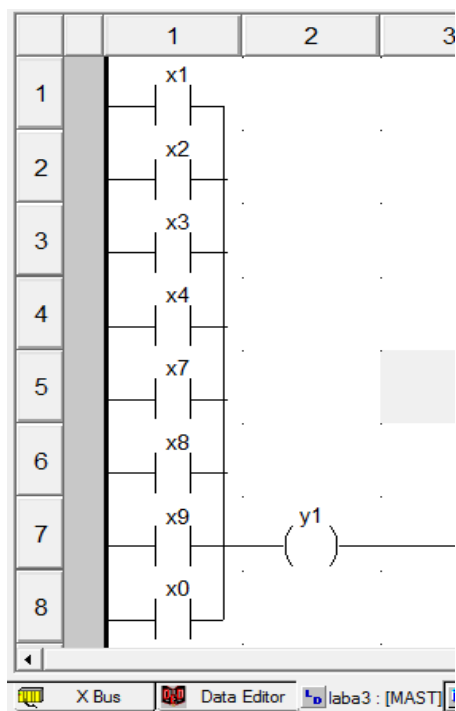


Рисунок 18 – Схема для  $y_1$

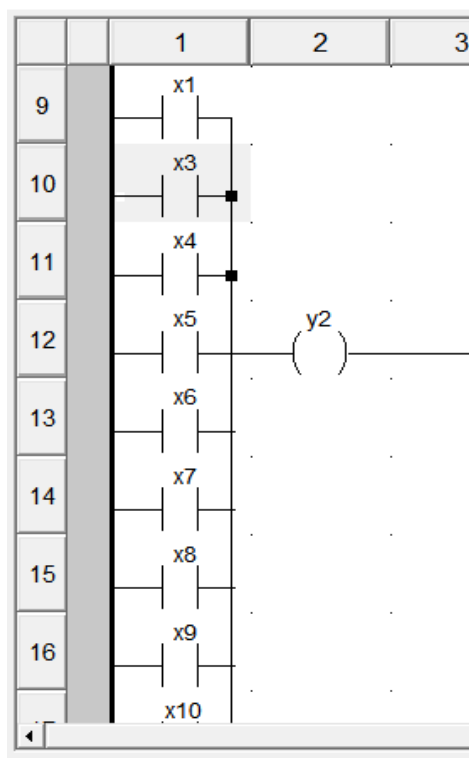


Рисунок 19 – Схема для  $y_2$

Далі відкриваємо Operation Screen → New Screen і з ліній створюємо семисегментний індикатор, для кожної з ліній задаємо відповідні параметри від y1 до y7 (рисунок 20).

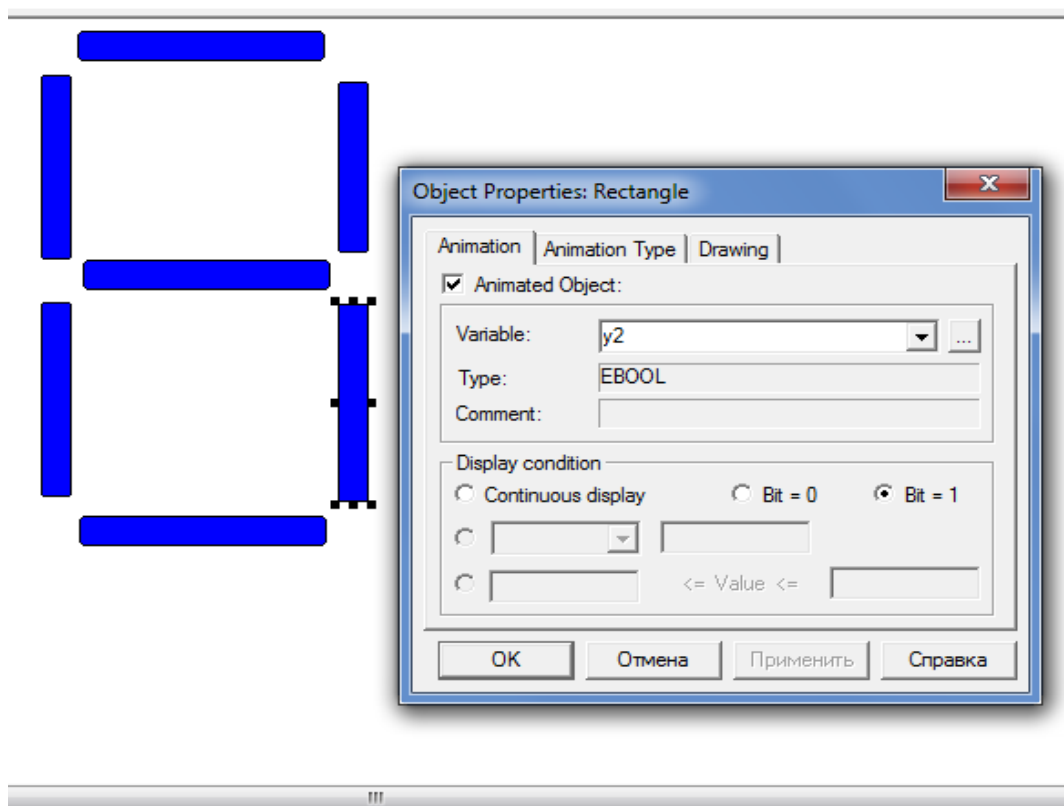


Рисунок 20 – Відтворення індикатора на екрані

Запускаємо програму (див. лабораторну роботу 2).

Для імітації натискання кнопки «2» при роботі в імітаційному режимі відкриваємо вікно Section та натискаємо ПКМ, наприклад на x2, і в меню обираємо Force Value → Force 1 (рисунок 21) потім відкриваємо вікно Screen і дивимось на результат (рисунок 22).

План виконання роботи:

- 1 Знайти додаткові джерела інформації.
- 2 Вивчити теоретичний і практичний матеріал, необхідний для виконання роботи.
- 3 Відповісти на контрольні питання.
- 4 Виконати роботу та індивідуальне завдання з розширеною складністю.

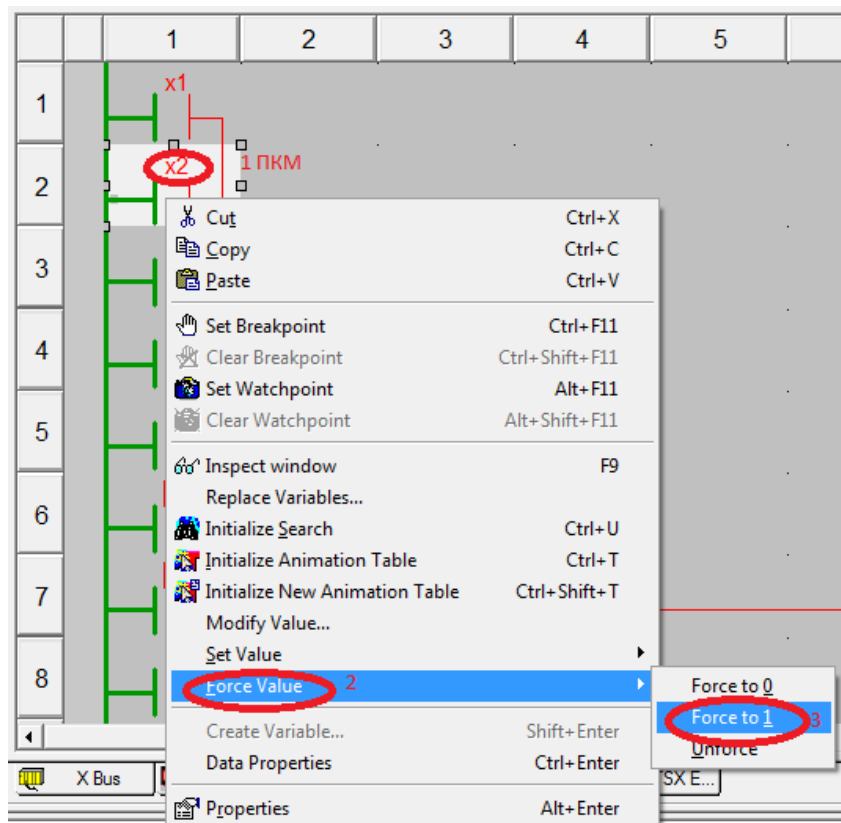


Рисунок 21 – Емуляції натискання кнопки «2»

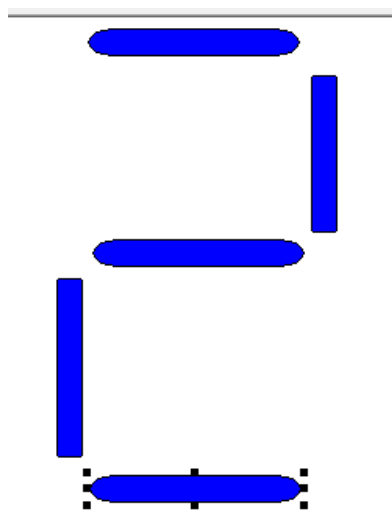


Рисунок 22 – Результат зміни зображення на екрані візуалізації об'єкта управління

Для того щоб змінити значення змінної  $x_2$ , знову натискаємо ПКМ та обираємо «Force 0», а на іншій потрібній цифрі (x) обираємо «Force 1».

Контрольні питання для зарахування ЛР:

1 Як створити семисегментний індикатор на модулях виводу дискретних сигналів?

2 Що зробити, щоб індикатор показував потрібну (будь-яку) цифру або літеру?

3 Наведіть приклади використання таких індикаторів у побуті, рекламі, транспорті.

4 За допомогою яких програмних мов Unity PRO можна управляти семисегментним індикатором?

Домашнє завдання для поглибленого засвоєння: зробити семисегментний індикатор для шістнадцяткової системи, щоб при натисканні двох кнопок і більше відображалась помилка у вигляді символу «=» (де «=» – Error), а якщо ніякого сигналу немає – символ «-» (відсутні вхідні дані).

Домашнє завдання підвищеної складності: зробити семисегментний індикатор для шістнадцяткової системи, щоб при натисканні двох кнопок і більше відображалась помилка у вигляді символу «=» (де «=» – Error), а якщо ніякого сигналу немає – символ «-» (відсутні вхідні дані), а у випадку зупинки або відсутності програми виводився символ «\_» (де «\_» – Stop).

#### **Лабораторна робота 4**

### **ПРОГРАМУВАННЯ МОДЕЛЕЙ ДИНАМІЧНИХ ПРОЦЕСІВ ТИПУ «БІЖУЧІ ВОГНІ»**

**Мета роботи:** навчитись створювати проекти, отримати навички створення програм управління динамічними процесами зі складанням програми та елементів візуалізації типу «біжучі вогні».

**Обладнання та ПЗ:** ПЕОМ з необхідними системними вимогами для роботи ПЗ, MS Windows, Internet Explorer, Open Office Writer, Unity PRO.

#### **Послідовність виконання роботи**


Як зазначалось у лекціях, візуальний ефект «біжучі вогні» реалізується шляхом послідовного вмикання не менше трьох світлодіодів на час 0,5 с кожний. Така динамічна модель реалізується послідовним перебором електронних «реле»  $У_i$

шляхом вимикання попереднього  $y_{i-1}$  та вмикання наступного  $y_{i+1}$  з затримкою на спрацювання 0,5 с (FFB–об’єкт TON з параметром  $PT=t\#500ms$ ).

Для виконання роботи необхідно створити новий проект з такими самими апаратними модулями, як і в другій лабораторній роботі, і налаштувати модуль ETU PORT. Далі відкриваємо вікно Variables & FB instances -> Elementary Variables і створюємо 5 змінних (рисунок 23).

Name	Type	Address
y1	EBOOL	%Q0.3.1
y2	EBOOL	%Q0.3.2
y3	EBOOL	%Q0.3.3
y4	EBOOL	%Q0.3.4
start	EBOOL	%I0.2.1

Рисунок 23 – Змінні проекту *laba4*

Наступним кроком на вкладці меню Structural view знаходимо параметр Program, обираємо Tasks, потім MAST, далі Section і правою кнопкою миші вибираємо New Section. Нову секцію краще назвати *laba4\_1*. Елемент схеми TON можна знайти на панелі закладок, він буде зображений у такому вигляді , або натиснути ПКМ у місці створення екземпляра FVB – об’єкта класу TON з іменем екземпляра FBI\_1 та обрати FVB Input Assistant (рисунок 24), де в FVB-тип ви запишете TON, і натиснути «ОК».

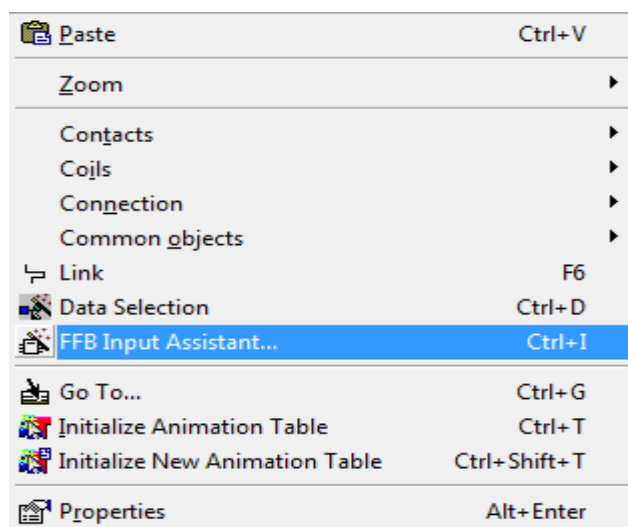


Рисунок 24 – Вставлення FVB – об’єкта через меню

«Схеми» вмикання  $y_1$ ,  $y_2$ ,  $y_3$  та  $y_4$  зображені на рисунках 25, 26.

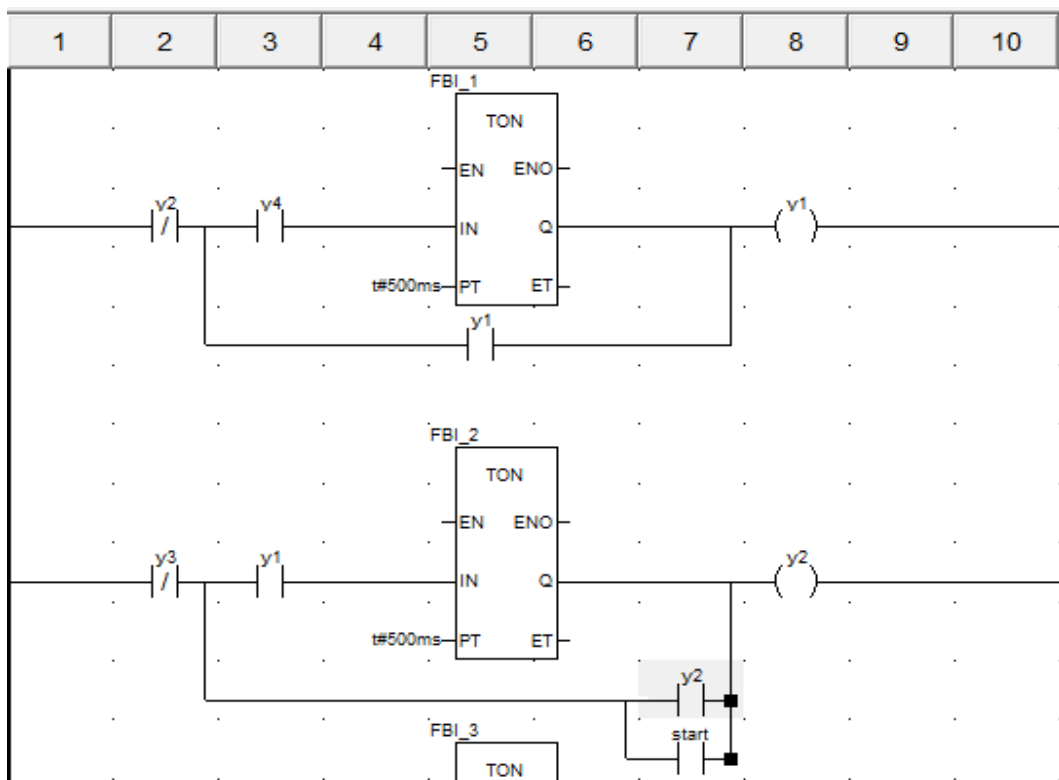


Рисунок 25 – «Схеми» вмикання  $y_1$  та  $y_2$

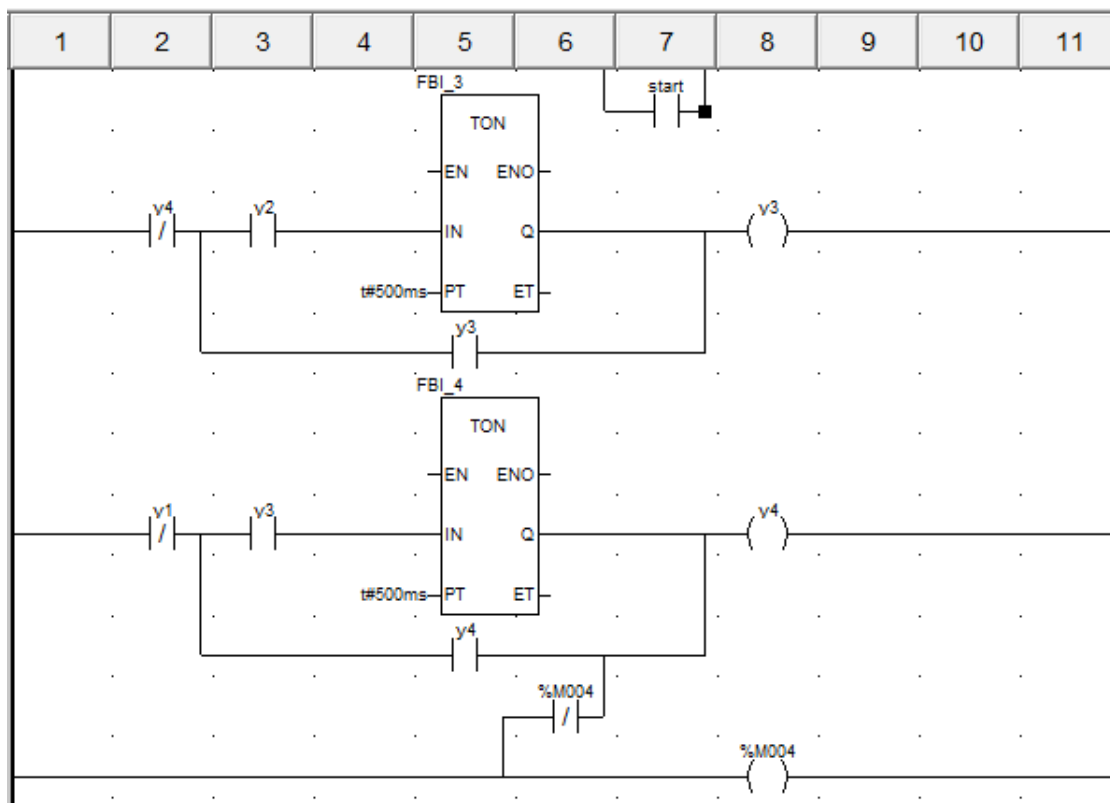


Рисунок 26 – «Схеми» вмикання  $y_3$  та  $y_4$



**Увага!** Під час виконання роботи у вас може виникнути така помилка, як на рисунку 27. Уникнути її можна побудовою елементів на відстані один від одного.

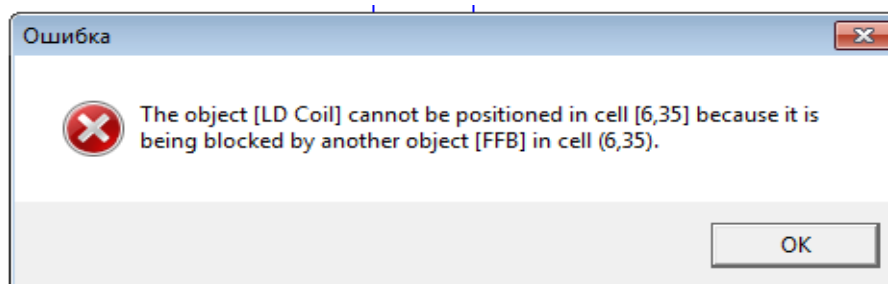


Рисунок 27 – Повідомлення про помилку розташування «реле»

Далі відкриваємо Operation Screen -> New Screen і створюємо чотири квадрати (рисунок 28), вибираємо їх колір і налаштовуємо до кожної із змінних  $y_1$ ,  $y_2$ ,  $y_3$  та  $y_4$  відповідно, ставимо в секції «Display condition» опцію біта Bit=1.

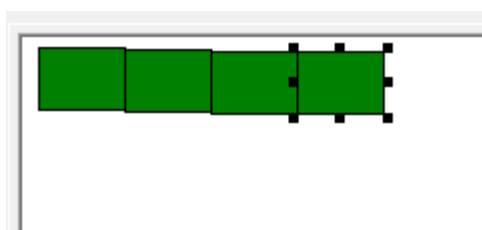


Рисунок 28 – Один набір квадратів для візуалізації роботи програми

Копіюємо ці 4 квадрати і вставляємо поряд декілька разів у лінію. Запускаємо програму.

Контрольні питання:

- 1 Де можна знайти елемент TON?
- 2 Для чого потрібен модуль TOF?
- 3 Які параметри вхідних даних елемента TON?
- 4 Які типи даних елемента TON?
- 5 Які параметри вихідних даних елемента TON?
- 6 Як необхідно змінити програму з чотирьох до трьох «реле»?
- 7 Як необхідно змінити програму з чотирьох до шести «реле»?

Домашнє завдання:

- 1 Створити біжучі вогні з автозапуском.
- 2 За власною ідеєю зробити миготливий рисунок.
- 3 Створити вогні з автозапуском, без допоміжних «реле».

## Лабораторна робота 5

### СТВОРЕННЯ FBD-МОДУЛІВ ДЛЯ МОДЕЛЮВАННЯ ТРИЗНАЧНОГО АВТОБЛОКУВАННЯ

**Мета роботи:** навчитись створювати FBD-модулі, отримати навички автоматизації технологічного процесу зі складанням програм мовою FBD з елементами візуалізації.

**Обладнання та ПЗ:** ПЕОМ з необхідними системними вимогами для роботи ПЗ, MS Windows, Internet Explorer, Open Office Writer, програми Unity PRO.

#### Послідовність виконання роботи

Як і в попередній роботі, компонуємо всі необхідні модулі та активуємо їх. Наступний крок – відкриваємо вікно Derived FB Types, щоб створити новий FBD-тип, на основі якого потім будуть створюватися екземпляри. Крім унікальної назви типу (за правилами створення ідентифікаторів), необхідно створити локальні вхідні та вихідні змінні в FBD-модулі та логіку «перетворення» вхідних змінних у вихідні та записати параметри, як на рисунках 29, 30. У випадку імітації маятникового трансмітера (MT) створимо екземпляр даних функціонального блока DFB-типу з назвою D\_MT, який буде відповідати за миготливий вихід.

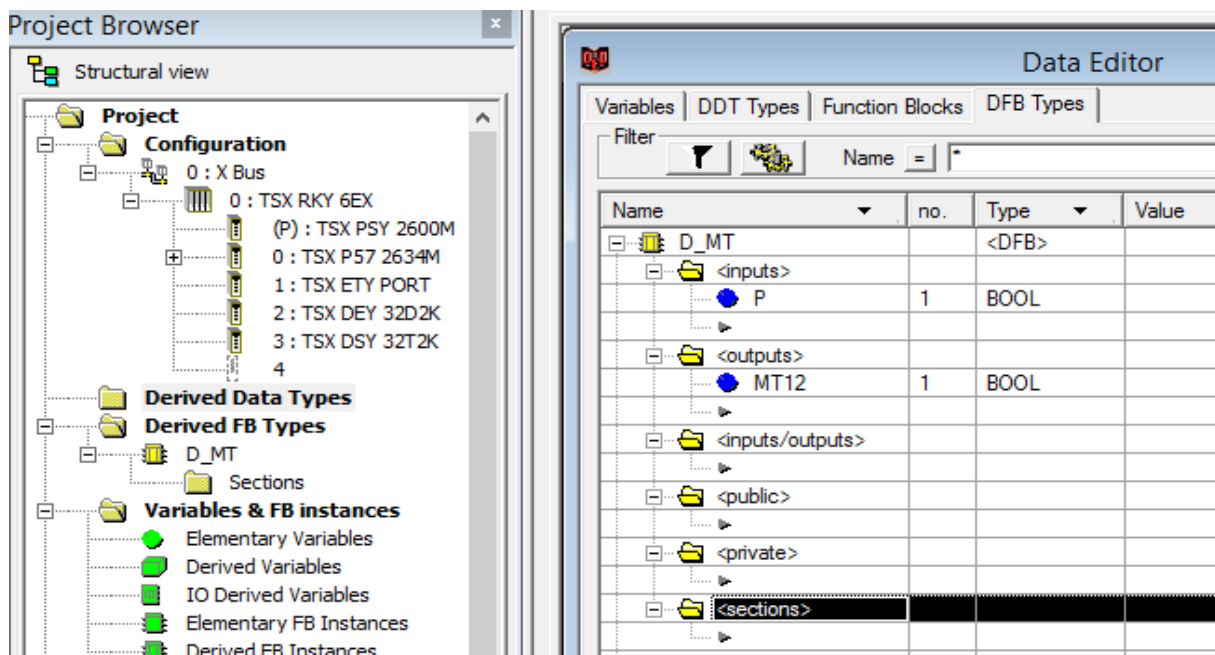


Рисунок 29 – Створення в розділі проекту Derived FB Types DFB-типу з назвою «D\_MT»

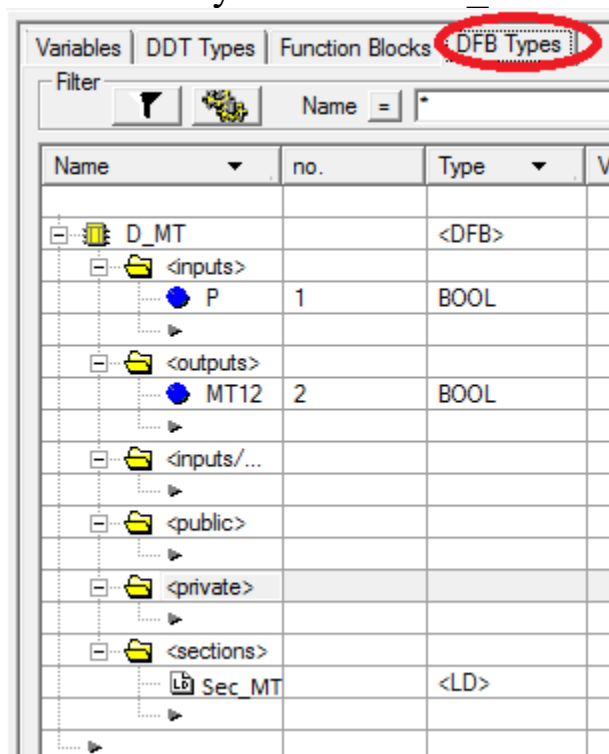


Рисунок 30 – Розроблення DFB-типу «D\_MT» з назвою змінних і визначенням їх типів

Потім будемо схему (рисунок 31), натискаючи на об'єкт Sec\_MT 2 рази у вікні (рисунок 30).

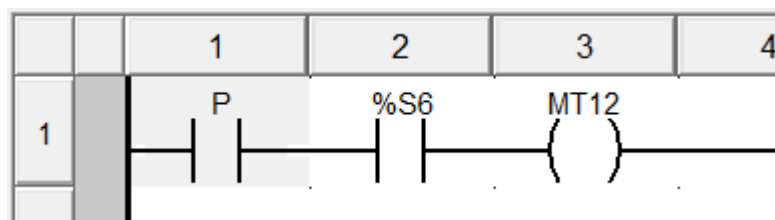


Рисунок 31 – Внутрішня схема DFB блока типу D\_MT

Відкриваємо операторський екран Screen\_laba5 і зображуємо світлофор з розташуванням показань тризначного АБ, як на рисунку 32, і задіємо кожен з кружків для трьох кольорів показань, де будуть наступні параметри з можливістю задавати в полі Variable локалізовані (рисунок 33) і нелокалізовані змінні.

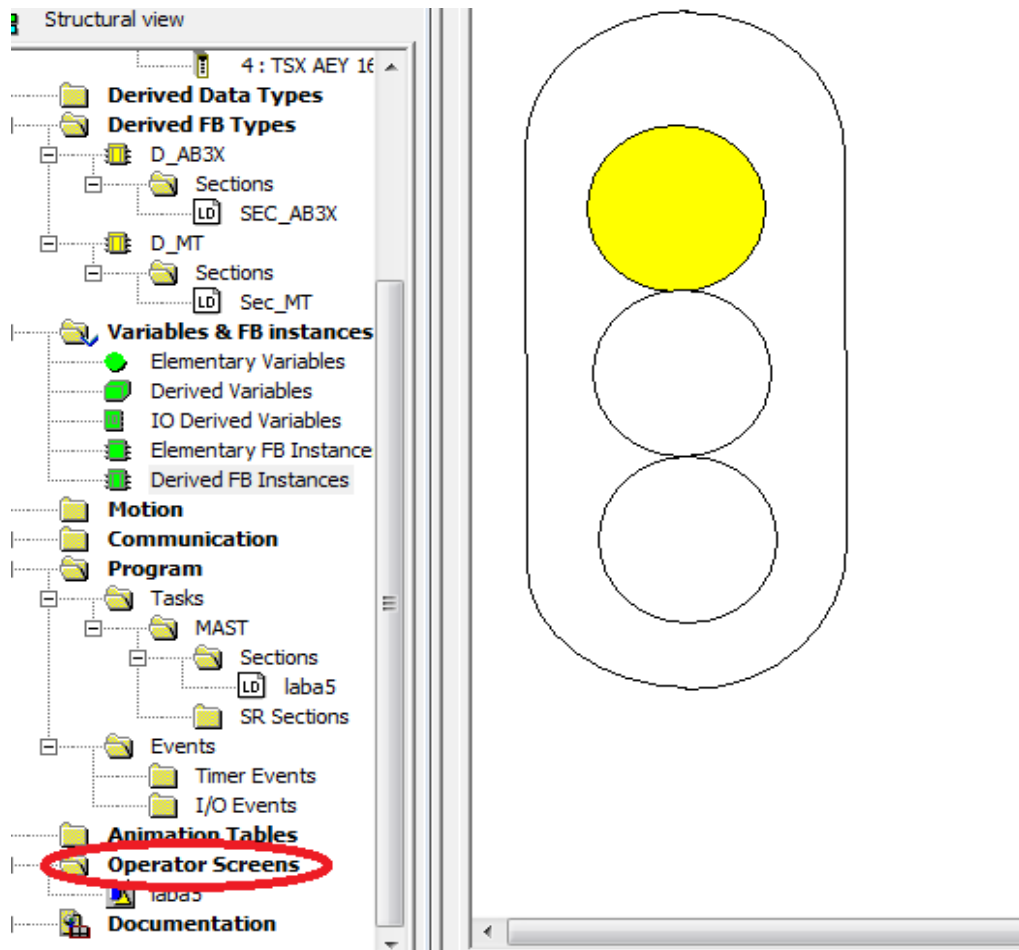


Рисунок 32 – Зображення показань тризначного АБ

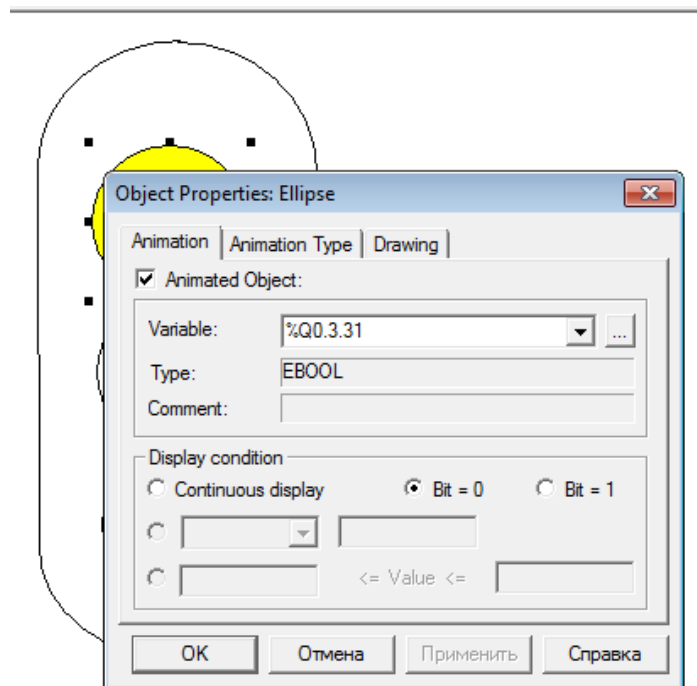


Рисунок 33 – Налаштування верхнього показання

Компілюємо підпрограму Sec\_MT. Випробовуємо програму з екземпляром DFB блока типу D\_MT, як зі стандартним блоком.

Після цього будуємо функціональну модель сигнальної точки для тризначного автоблокування (АБ) за схемами двоколійного автоблокування постійного струму при автономній тязі для дільниць з одностороннім рухом поїздів [5].

Виокремлюємо схему сигнальної точки в окрему «типову» схему вмикання ламп світлофора. Сигнали залежать від стану рейкового кола, яке огорожене світлофором, і стану попередньої сигнальної точки, які представлені змінними. Створюємо функціональний блок користувача для сигнальної точки тризначної АБ «D\_AB3X» (рисунок 34).

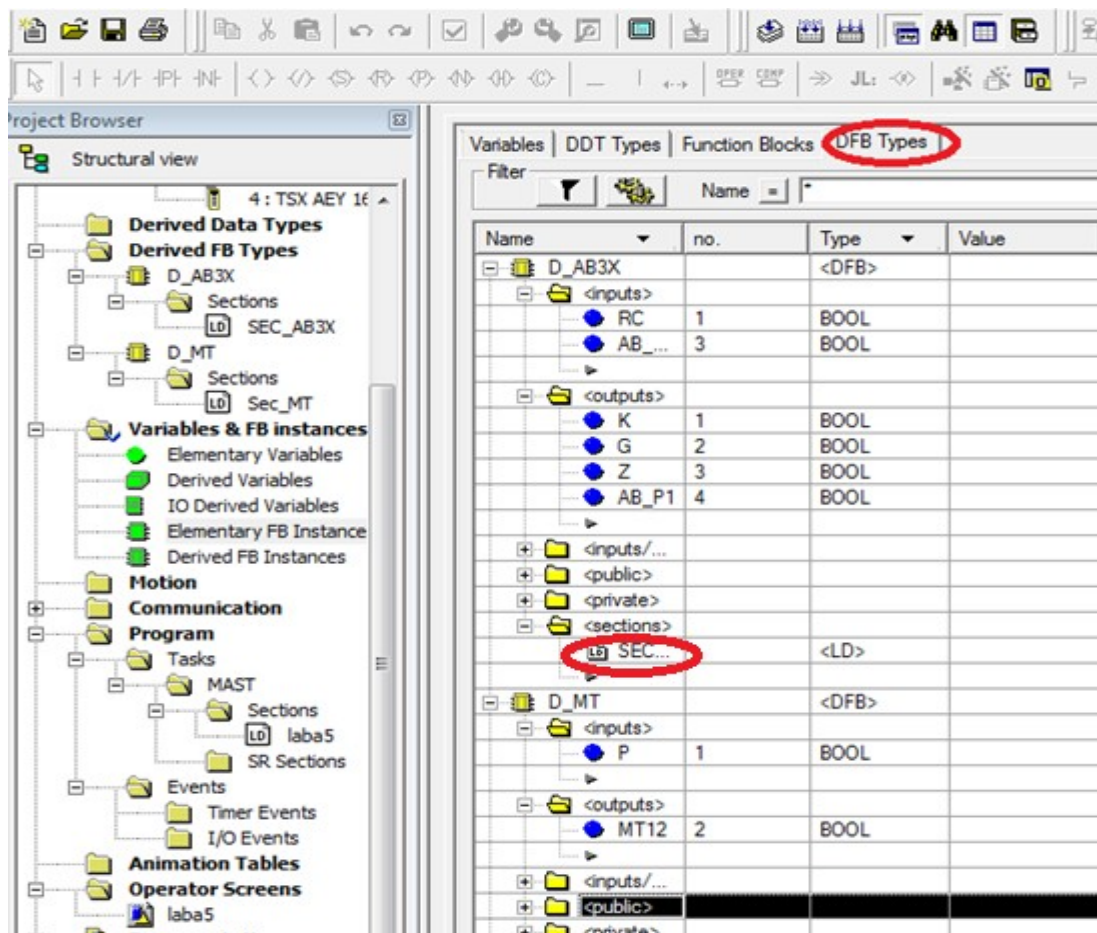


Рисунок 34 – Структура DFB типу з ім'ям D\_AB3X

Створивши DFB типи для тризначного АБ, додаємо програму поєднання вхідних і вихідних змінних будь-якою з п'яти мов. Для цього в секції D\_AB3X та D\_MT відкриваємо вкладку Section і створюємо SEC\_AB3X та SEC\_MT, відповідно

натискаючи на кожну 2 рази мишкою (має з'явитись вікно, як на рисунку 35).

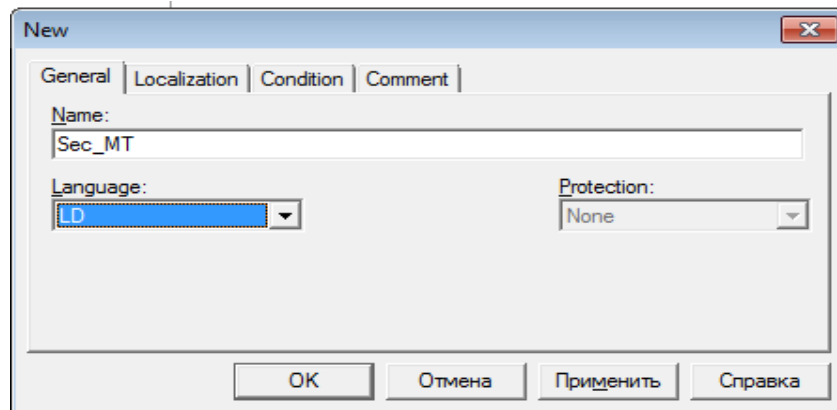
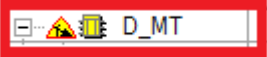



Рисунок 35 – Вибір мови програмування для SEC\_MT

У процесі створення цих елементів буде відобразитись такий знак , це означає, що елемент у процесі обробки. Після завершення створення треба скомпілювати програму, щоб не було помилок, за допомогою символу . У секціях, які ми відкрили, створюємо такі схеми як на рисунках 36, 37.

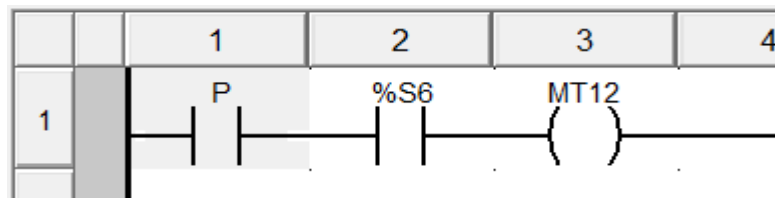


Рисунок 36 – Схема SEC\_MT миготливого виходу

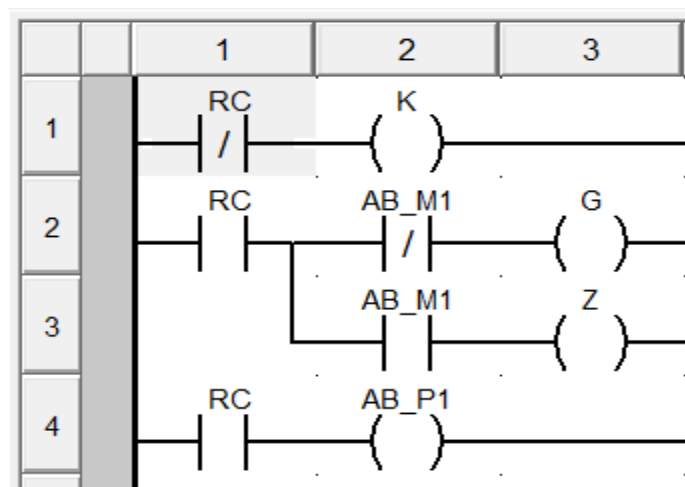


Рисунок 37 – Схема SEC\_AB3X для тризначного світлофора

Знову компілюємо програму. Далі відкриваємо вже знайому нам вкладку Program і відкриваємо створені нами схеми (так, як ми створювали TON в попередній роботі), вписуючи їхні назви (рисунок 38).

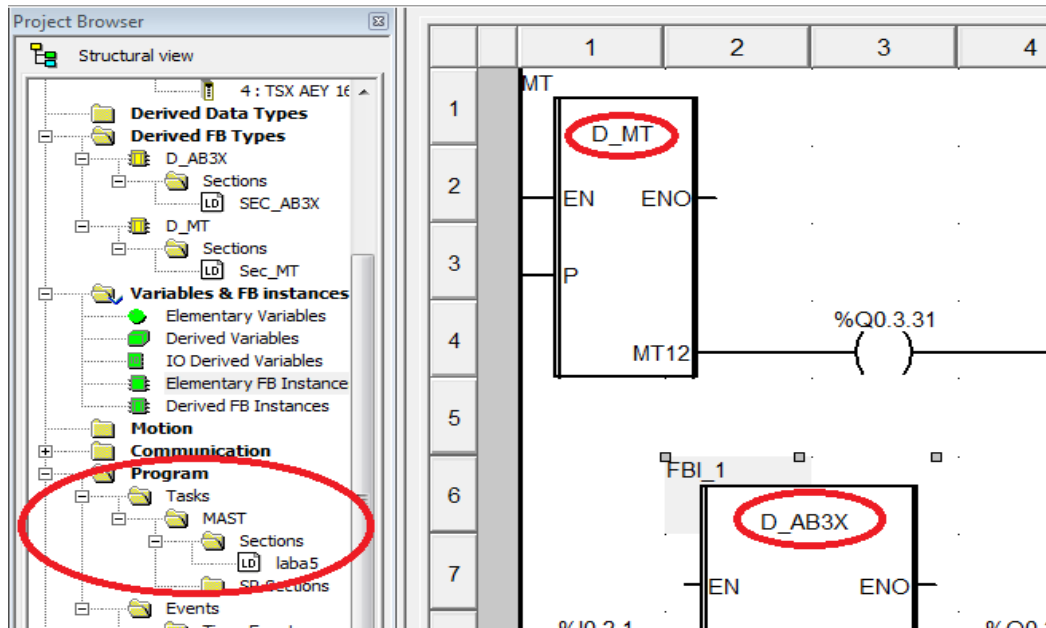
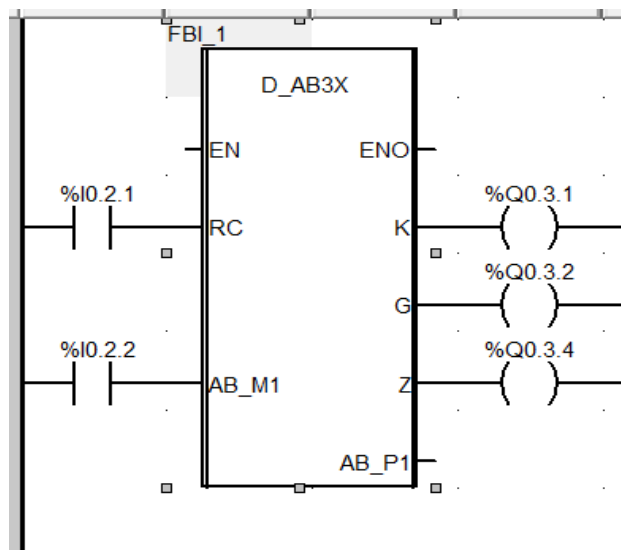


Рисунок 38 – Схема для перевірки екземпляра MT

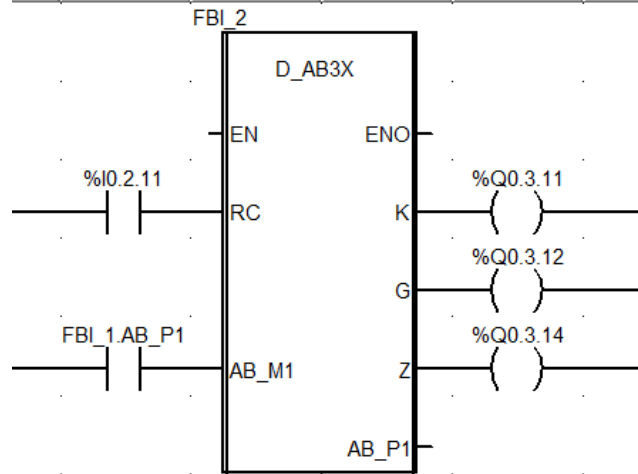
Створивши екземпляри FBI\_1, FBI\_2, FBI\_3 на базі функціонального блока користувача (DFB-типу з ім'ям D\_AB3X), підключаємо до їхніх виходів «реле» з локалізованими змінними (рисунок 39 а, б, в).



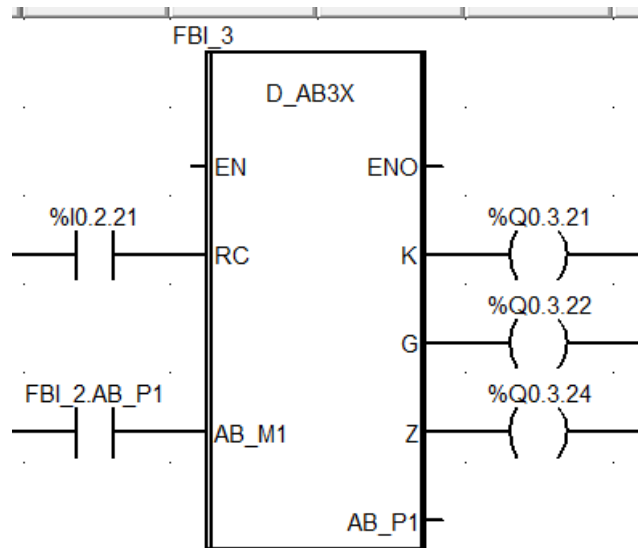
а

Рисунок 39 – Схеми підключення екземплярів FBI\_1 (а),

## FBI\_2 (б), FBI\_3 (в) (аркуш 1)



б



в

Рисунок 39 – Схеми підключення екземплярів FBI\_1 (а),  
FBI\_2 (б), FBI\_3 (в) (аркуш 2)

Вихідну змінну AB\_P1 блока FBI\_1 допускається контролювати в інших колах як контакт з ім'ям **FBI\_1.AB\_P1**. Блоки будуюмо один під одним. Запускаємо програму, як наведено на рисунках 5, 6. У режимі емуляції роботи ПЛК відображаються червоним кольором розімкнені кола, а зеленим кольором замкнуті.



Щоб перевірити, чи правильно працює програма, треба встановити в «1» %I0.2.21 ПКМ Force Value 1 біт (рисунки 40, 41).

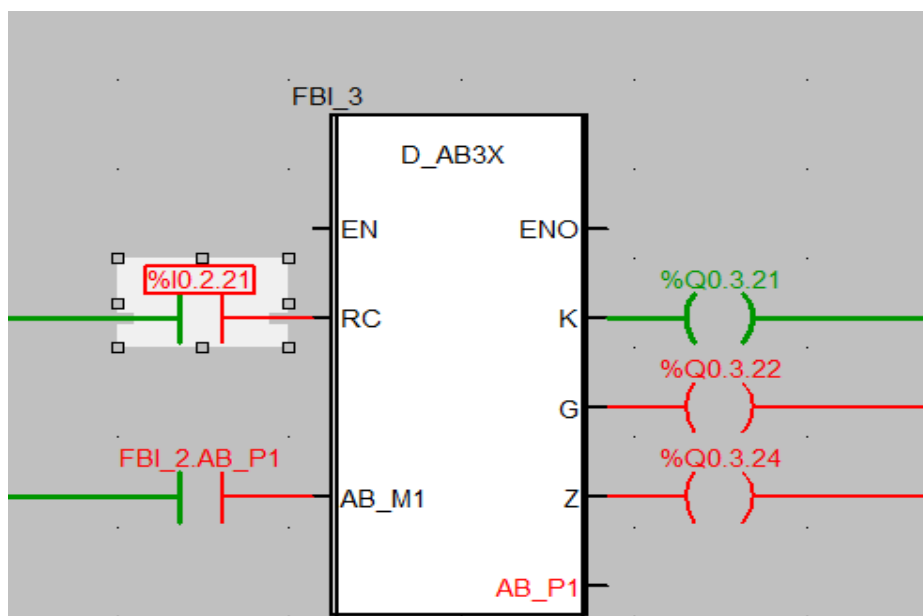


Рисунок 40 – Перевірка роботи FBI\_3

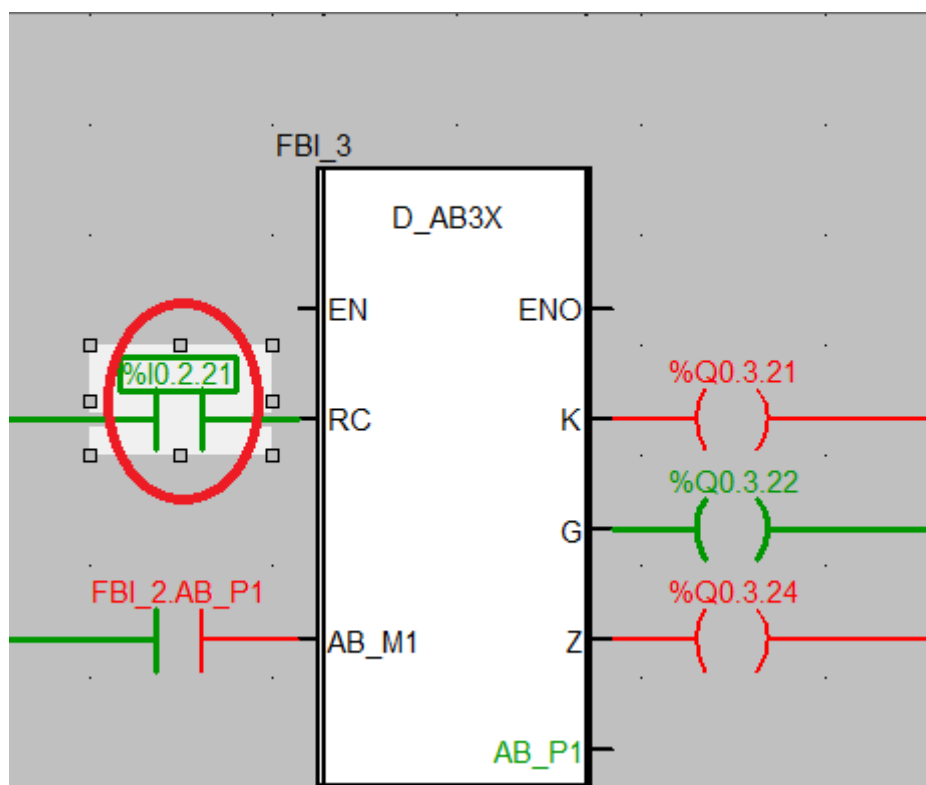


Рисунок 41 – Зміна стану входу «RC» блока FBI\_3

План виконання роботи:

1 Знайти додаткові джерела інформації.

2 Вивчити теоретичний і практичний матеріал, необхідний для виконання роботи.

3 Відповісти на контрольні питання та виконати домашнє завдання.

4 Спланувати лабораторну роботу згідно з домашнім завданням відповідного рівня.

Контрольні питання

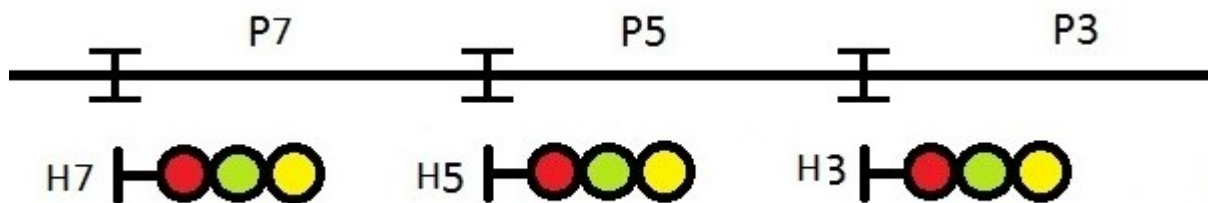
1 Як створити новий блок?

2 Як працює прохідна та передвхідна сигнальна точка?

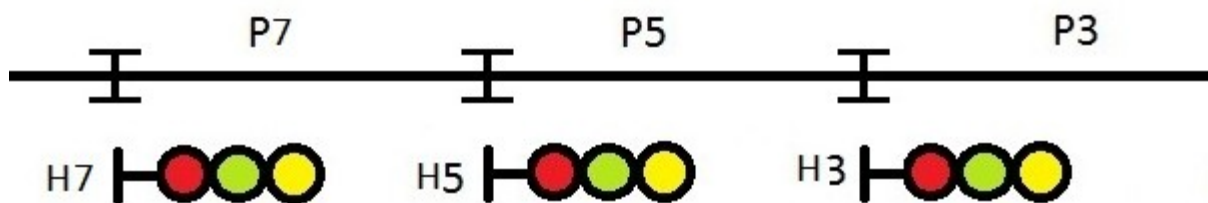
3 Як підключити три лампи світлофора до блока D\_AB3X?

Домашнє завдання зі створенням власних DFB:

1 Схематично зобразити у вікні Screen\_laba5\_dzE ділянку колії з трьох рейкових кіл P7, P5, P3 і трьох світлофорів H7, H5, H3, з'єднавши їх у єдину технологічну модель роботи за схемою перегону, наведеною нижче.

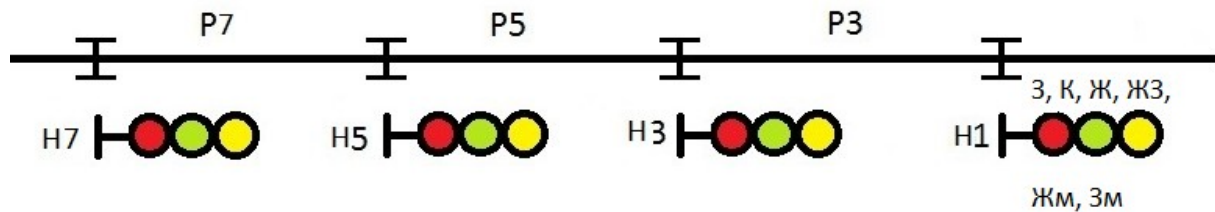


2 Схематично зобразити у вікні Screen\_laba5\_dzD ділянку колії з трьох рейкових кіл P7, P5, P3 і трьох світлофорів H7, H5, H3, з'єднавши їх у єдину технологічну модель роботи за схемою перегону, наведеною нижче з переносом червоного показання у випадку перегорання червоної лампи.

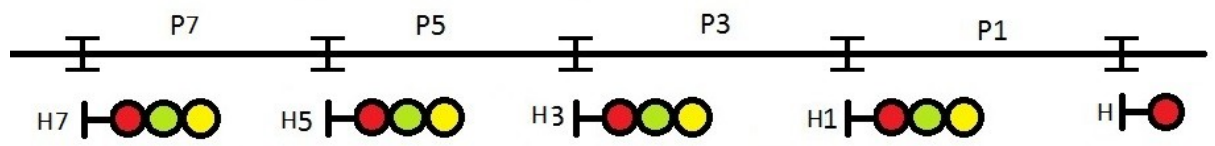


3 Схематично зобразити у вікні Screen\_laba5\_dzC ділянку колії з трьох рейкових кіл P7, P5, P3 і чотирьох світлофорів H7, H5, H3, H1, з'єднавши їх у єдину технологічну модель роботи за схемою чотиризначного автоблокування перегону, наведеною нижче, з перенесенням червоного показання у випадку

перегорання червоної лампи та вимикання зеленого вогню при показанні «жовтий з зеленим» у випадку перегорання лампи жовтого вогню й інтерактивним введенням зайнятості рейкових кіл Р7, Р5, Р3 і роботи світлофора Н1 (шість показань).



4 Схематично зобразити у вікні Screen\_laba5\_dzA ділянку колії з чотирьох рейкових кіл Р7, Р5, Р3, Р1 і п'яти світлофорів Н7, Н5, Н3, Н1, Н, з'єднавши їх у єдину технологічну модель роботи за схемою чотиризначного автоблокування перегону, наведеною нижче, з перенесенням червоного показання у випадку перегорання червоної лампи та вимикання зеленого вогню при показанні «жовтий з зеленим» у випадку перегорання лампи жовтого вогню з роботою Н1 в режимі передвхідної сигнальної установки з інтерактивним введенням зайнятості рейкових кіл Р7, Р5, Р3, Р1 і роботи світлофора Н (шість показань – К, Жв, З, Жн+Жв, Жн+Жв+Зп, К+Бмиг).



## Лабораторна робота 6 ПОБУДОВА ШИФРАТОРІВ КОДОВИХ СИГНАЛІВ

**Мета роботи:** отримати навички конфігурування проекту для процесу динамічної генерації технологічних послідовностей імпульсів зі складанням програм та елементів візуалізації.

**Обладнання та ПЗ:** ПЕОМ з необхідними системними вимогами для роботи ПЗ, MS Windows, Internet Explorer, Open Office Writer, програми Unity PRO.

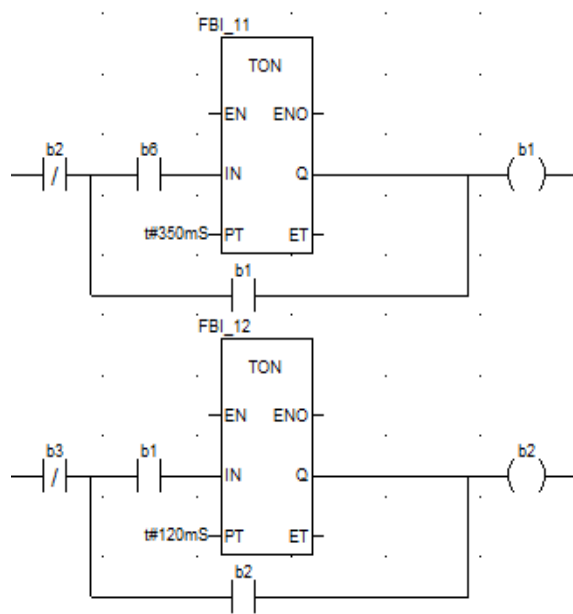
### Послідовність виконання роботи

Так, як і в попередній роботі, встановлюємо всі необхідні модулі та активуємо їх. Наступний крок – відкриваємо вікно Variables & FB instances -> Elementary Variables та створюємо нові змінні (рисунок 42).

Name	Type	Address
b1	BOOL	%MW5.1
b2	BOOL	%MW5.2
b3	BOOL	%MW5.3
b4	BOOL	%MW5.4
b5	BOOL	%MW5.5
b6	BOOL	%MW5.6
G5	BOOL	%MW5.11
K5	BOOL	%MW5.12
y1	BOOL	%MW5.7
y2	BOOL	%MW5.8
y3	BOOL	%MW5.9
y4	BOOL	%MW5.10
Z5	BOOL	%MW5.0

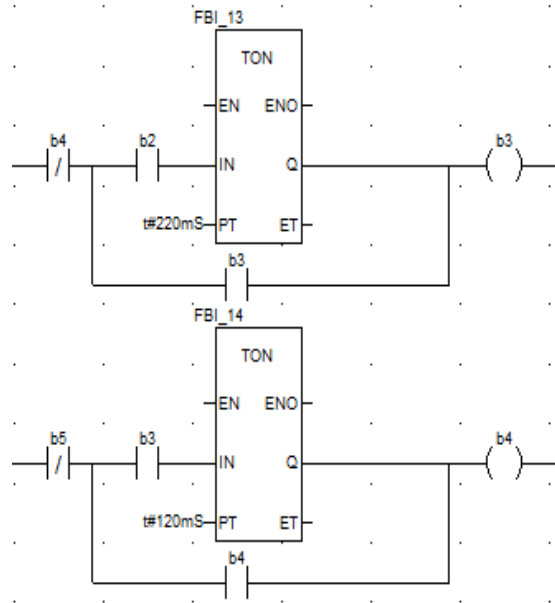
Рисунок 42 – Змінні для моделювання КПТШ 515

Потім відкриваємо Derived FB Types, щоб створити нові елементи схеми пристрою КПТШ515, і записуємо параметри подачі коду «З», «Ж», «КЖ», як на рисунках 43, 44, 45. Для коду «З» - три різні імпульси й три різні паузи; для коду «Ж» - два різні імпульси, дві різні паузи; «КЖ» - один імпульс, одна пауза.

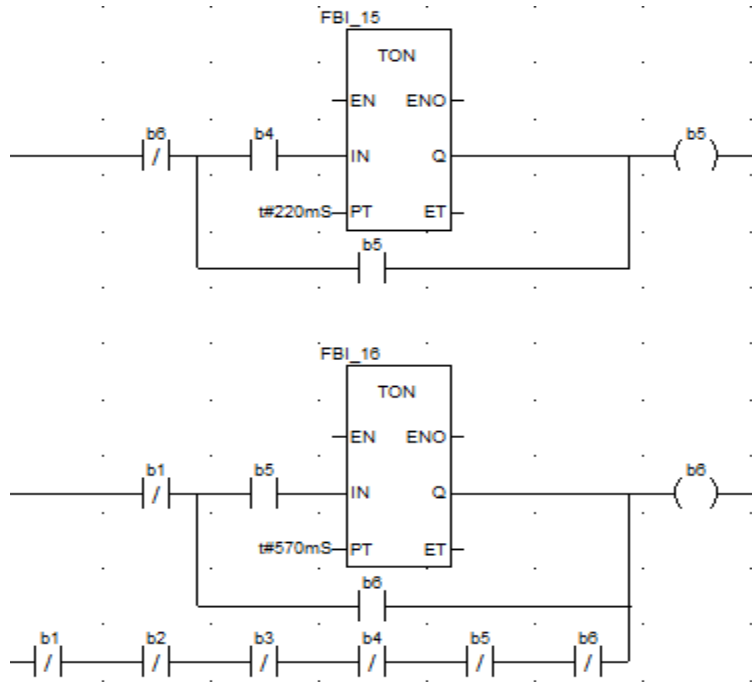


а

Рисунок 43 – Перша (а), друга (б) і третя (в) пари «імпульс-пауза» коду «3» (аркуш 1)

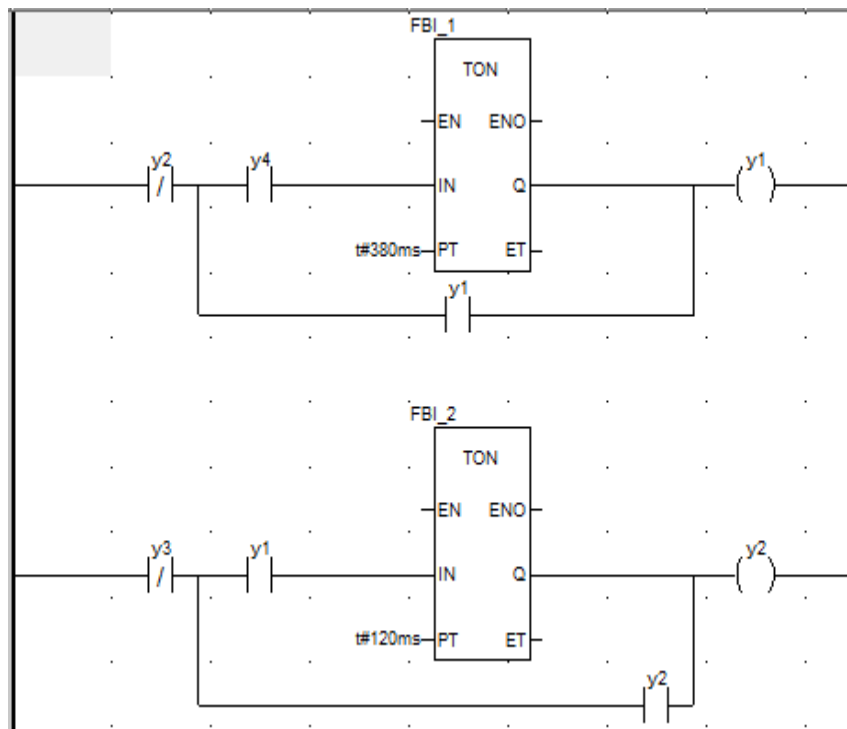


б

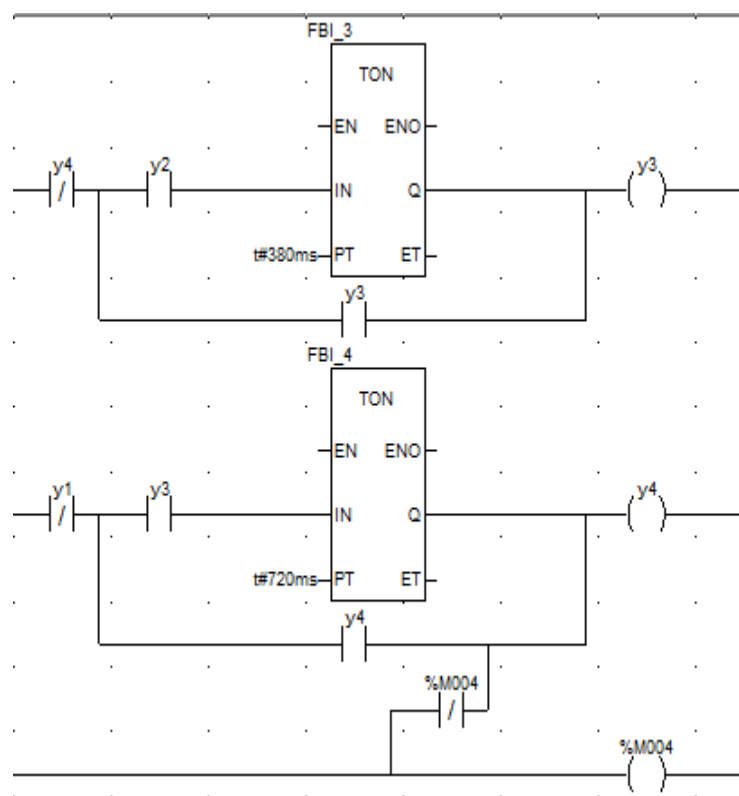


в

Рисунок 43 – Перша (а), друга (б) і третя (в) пари «імпульс-пауза» коду «3» (аркуш 2)



а



б

Рисунок 44 – Перша (а) і друга (б) пари «імпульс-пауза» коду «Ж»

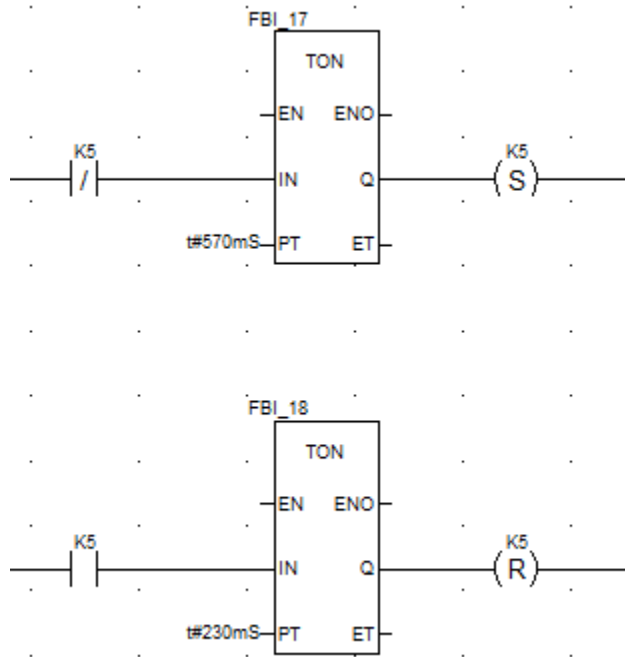


Рисунок 45 – Пара «імпульс-пауза» коду «КЖ»

Далі запишемо загальну схему подачі інвертованих імпульсів КПТШ515 (рисунок 46).

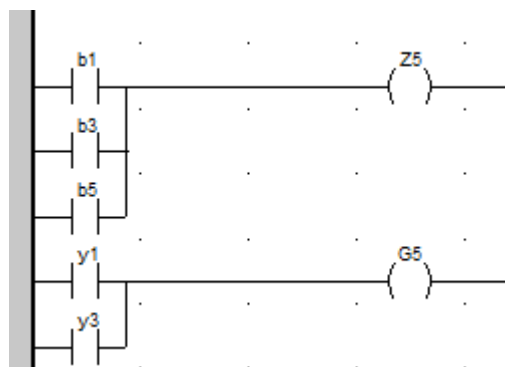


Рисунок 46 – Схема інвертованих виходів «З» та «Ж» КПТШ 515

Скомпілюємо та запусимо програму.

Запустивши програму, ви маєте помітити, що імпульси для кожного коду працюють згідно з заданими в лабораторній роботі значеннями (рисунок 47).

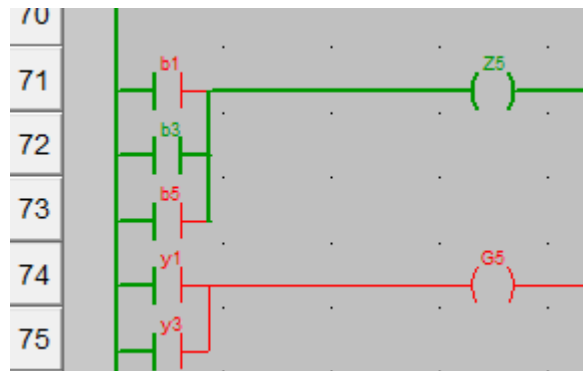


Рисунок 47 – Генерація другої паузи коду «З»

Наступним кроком побудуємо шифратор для КПТШ515. Відкриваємо вікно Variables & FB instances -> Elementary Variables і записуємо дані з рисунка 48.

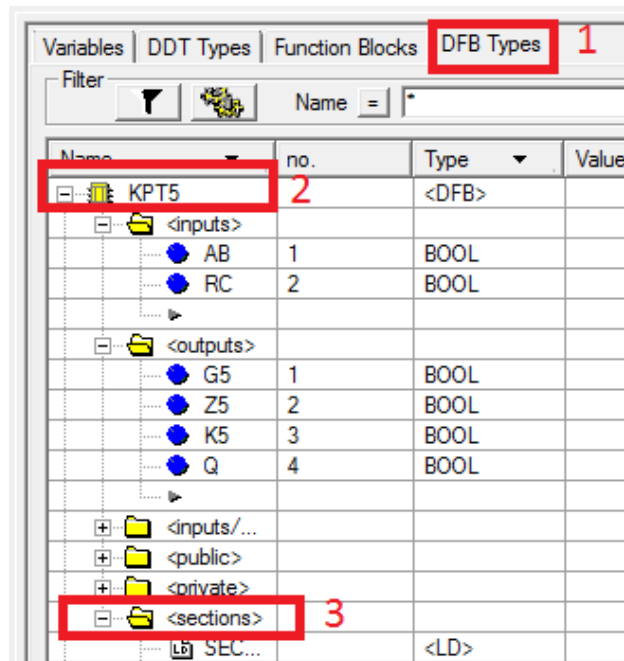


Рисунок 48 – Створення DFB типу КПТШ 515

У рядку Sections створюємо нову секцію мовою LD (рисунок 49).



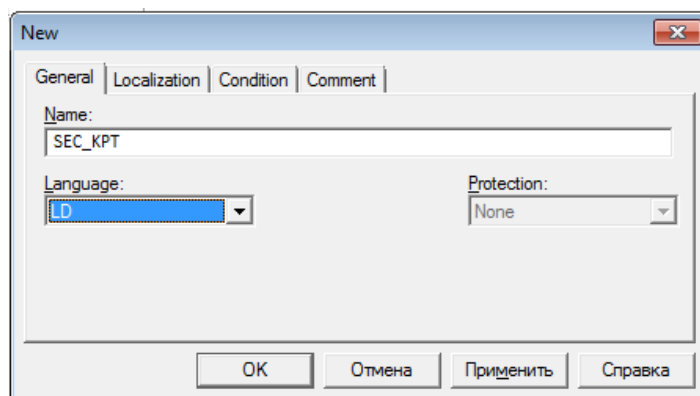


Рисунок 49 – Створення секції мовою LD

У даній секції реалізуємо таку схему, як на рисунках 43, 44, 45, 46 з незначними змінами для генерації кодів КПТШ 515.

Компілюємо програму та знову повертаємося до вікна, у якому ми будували схеми КПТШ 515, там створюємо екземпляр класу, як показано на рисунках 50, 51.

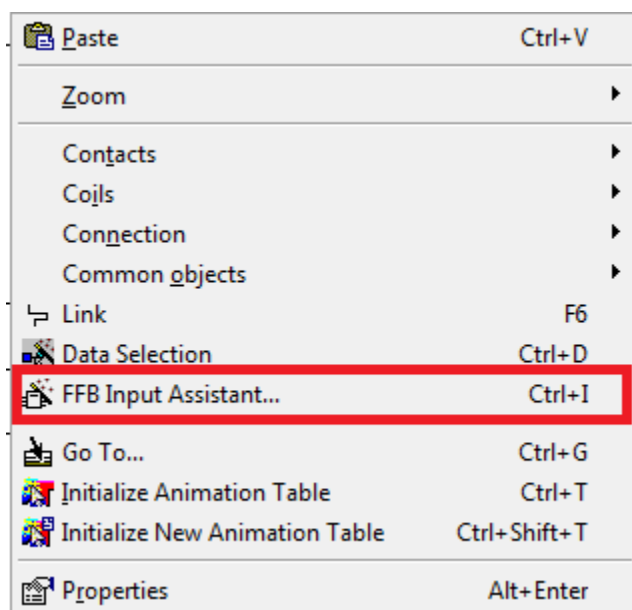


Рисунок 50 – Вставлення екземпляра FFB/DFB-типу

Записуємо назву FFB/DFB-типу (рисунок 51) у полі FFB type та коригуємо або підтверджуємо генеровану назву (FBI\_21).

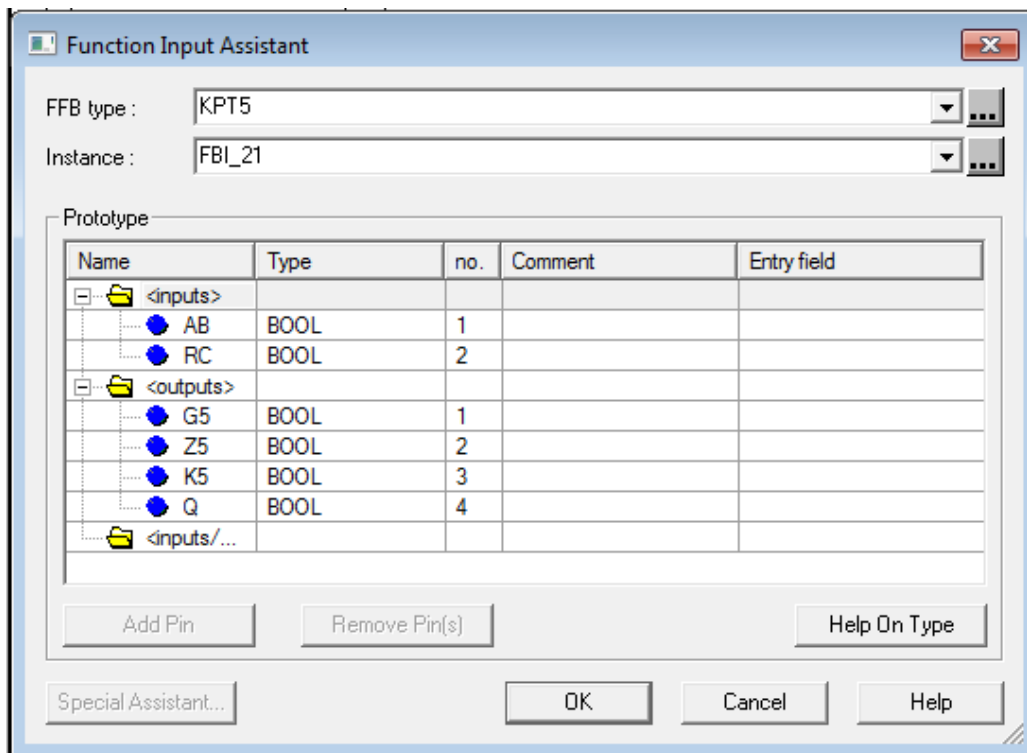


Рисунок 51 – Корекція назви екземпляра FFB/DFB-типу

На пустому вікні натискаємо ліву кнопку миші, з'явиться екземпляр розробленого DFB-модуля, для якого необхідно доробити схему під'єднання, як на рисунку 52.

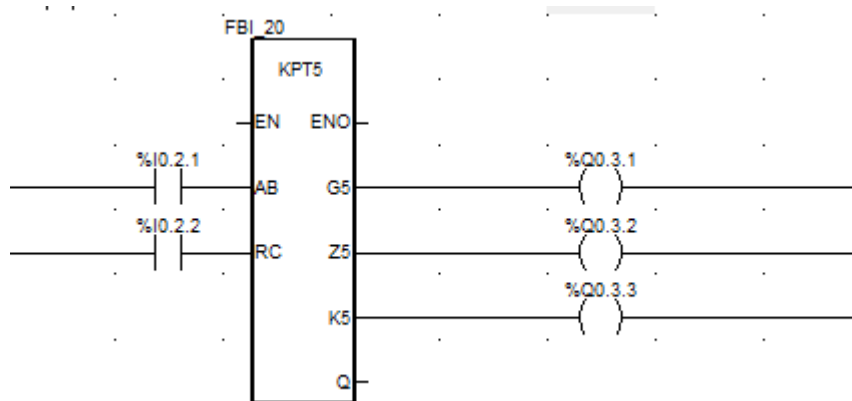


Рисунок 52 – Схема перевірки DFB-модуля КПТШ 515

Запускаємо програму й інтерпретуємо результат, використовуючи рисунок 53.

Контрольні питання:

1 Скільки імпульсів у кожного з кодів КПТШ 515, КПТШ 715, КПТШ 815, КПТШ 1015, КПТШ 1115?

2 Які ще бувають коди КПТШ 1015, КПТШ 1115?

3 Для чого потрібен КПТШ?

4 Як зробити кодер МТ-1, МТ-1М, МТ-2, МТ-2М?

5 Як зробити кодер ТП-24, ТП-24М?

Домашнє завдання з застосуванням DFB:

1 Створити DFB-модулі окремо для генерації КПТШ 515-«КЖ», КПТШ 515-«Ж», КПТШ 515-«З».

2 Створити DFB-модуль для генерації КПТШ 715.

3 Створити DFB-модуль для генерації КПТШ 1015.

4 Створити DFB-модуль для генерації МТ-2М.

5 Створити DFB-модуль для генерації КПТШ 1115.

6 Створити DFB-модулі окремо для генерацій КПТШ 1015-«А1», КПТШ 1015-«А2».

7 Створити DFB-модулі окремо для генерацій КПТШ 715-«КЖ», КПТШ 715-«Ж», КПТШ 715-«З».

Тип трансмитера	Обозначение кода	Продолжительность импульсов и интервалов
КПТШ-5 (КПТШ-515) КПТШ-8 (КПТШ-815)	-З-	
	-Ж-	
	-ЮК-	
КПТШ-7 (КПТШ-715) КПТШ-9 (КПТШ-915)	-З-	
	-Ж-	
	-ЮК-	
КПТШ-10 (КПТШ-1015) КПТШ-13 (КПТШ-1315)	-А1-	
	-А2-	
КПТШ-11 (КПТШ-1115)	-З-	
	-Ж-	
	-ЮК-	

Рисунок 53 – Часові характеристики трансмітерів

## Лабораторна робота 7 ПОБУДОВА ДЕШИФРАТОРІВ КОДОВИХ СИГНАЛІВ

**Мета роботи:** отримати навички конфігурування проекту для процесу динамічного, циклічного дешифрування технологічних послідовностей імпульсів зі складанням програм та елементів візуалізації.

**Обладнання та ПЗ:** ПЕОМ з необхідними системними вимогами для роботи ПЗ, MS Windows, Internet Explorer, Open Office Writer, програми Unity PRO.

### Послідовність виконання роботи

Так, як і в попередній роботі, встановлюємо всі необхідні модулі та активуємо їх. Наступний крок – відкриваємо вікно Variables & FB instances -> Elementary Variables та створюємо нові змінні.

Щоб створити нові елементи схеми пристрою дешифрування необхідно проаналізувати роботу всіх типів КПТШ (рисунок 53). Простий аналіз дозволяє створити схему вмикання котушки на вимикання S1, яка фіксуватиме наявність великої кодової паузи будь-якого з режимів шифраторів КПТ (рисунок 54).

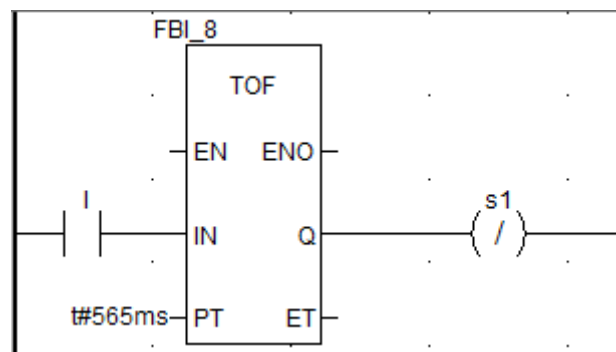
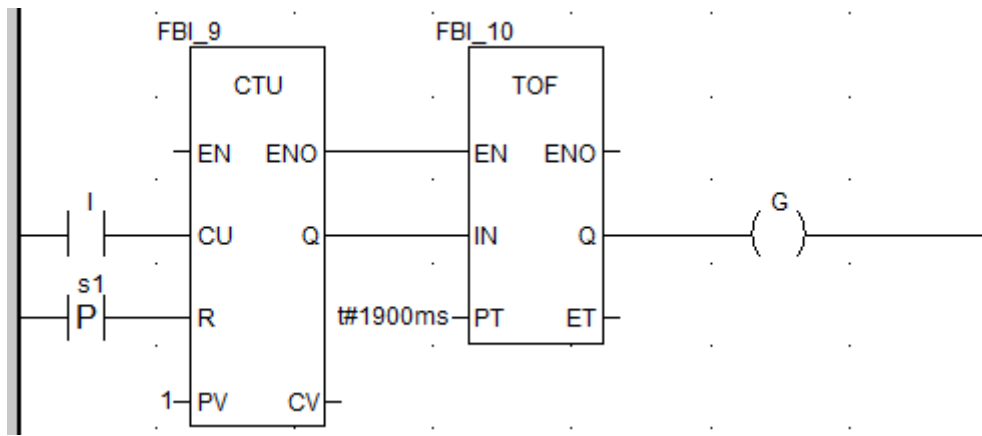
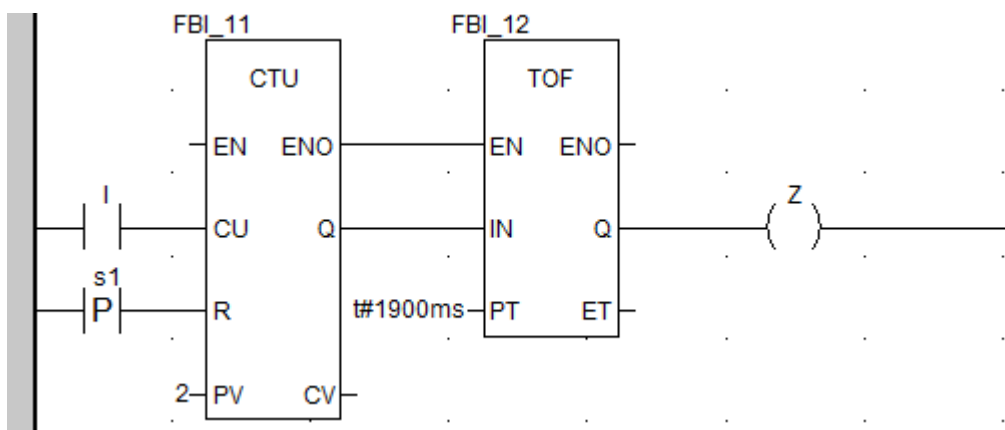


Рисунок 54 – Схема фіксації великої кодової паузи КПТ

Наступним кроком стало можливим створити схему підрахунку імпульсів у кодовій послідовності з застосуванням блока STU. Котушки «G» і «Z» будуть, за аналогією з класичним автоблокуванням, фіксувати наявність відповідно першого (рисунок 55, а) і другого (рисунок 55, б) імпульсів кодової послідовності.



а



б

Рисунок 55 – Схеми фіксації першого (а) і другого (б) імпульсів будь-якого коду КПТ

Скомпілюємо та запустимо програму. Однак для технологічного налагодження модуля дешифратора необхідно змусити працювати реле «I» в режимі шифратора, як наводилось у лабораторній роботі 6. Запустивши програму, ви маєте помітити, що імпульси для кожного коду генеруються згідно з заданими в лабораторній роботі 6 значеннями (рисунок 53), а розроблена в даній лабораторній роботі схема декодує їх у стани реле «G» і «Z».

Контрольні питання:

1 Які реле контролюють імпульси кожного з кодів КПТШ 515, КПТШ 715, КПТШ 815, КПТШ 915, КПТШ 1015, КПТШ 1315?

2 Які коди КПТШ використовують для кодування рейкових кіл і яке інформаційне навантаження вони виконують?

3 Для чого потрібен КПТШ на станції, у релейній шафі перегону, релейній шафі переїзду?

4 Як зробити декодер МТ-1, МТ-1М, МТ-2, МТ-2М?

5 Як зробити декодер ТП-24, ТП-24М?

Домашнє завдання з застосуванням DFB:

1 Сформулювати умови створення DFB-модулів окремо для декодування КПТШ 515-«КЖ», КПТШ 515-«Ж», КПТШ 515-«З».

2 Сформулювати умови створення DFB-модулів окремо для декодування КПТШ 715.

3 Сформулювати умови створення DFB-модулів окремо для декодування КПТШ 1015.

4 Сформулювати умови створення DFB-модулів окремо для декодування МТ-2М.

5 Сформулювати умови створення DFB-модулів окремо для декодування КПТШ 1115.

6 Створити DFB-модулі окремо для декодування КПТШ 1015-«А1», КПТШ 1015-«А2».

7 Сформулювати умови створення DFB-модулів окремо для декодування КПТШ 715-«КЖ», КПТШ 715-«Ж», КПТШ 715-«З».

## **Лабораторна робота 8 СТВОРЕННЯ FBD-МОДУЛІВ ДЛЯ МОДЕЛЮВАННЯ ТРИЗНАЧНОГО КОДОВОГО АВТОБЛОКУВАННЯ**

**Мета роботи:** навчитись створювати FBD-модулі, отримати навички автоматизації технологічного процесу зі складанням програм мовою FBD з елементами візуалізації сигнальних точок кодового автоматичного блокування.

**Обладнання та ПЗ:** ПЕОМ з необхідними системними вимогами для роботи ПЗ, MS Windows, Internet Explorer, Open Office Writer, програма Unity PRO.

### **Послідовність виконання роботи**

Як і в попередній роботі, komponуємо всі необхідні модулі та активуємо їх. Наступний крок – відкриваємо вікно Derived FB Types, щоб створити новий FBD-тип (КАВЗ\_5 для тризначного

автоблокування з КПТШ 515, або КАВ4\_5 для чотиризначного автоблокування з КПТШ 515), на основі якого (рисунок 56) потім будуть створюватися екземпляри блоків, таких як N3, N5, N7, з номерами сигнальних точок у непарному напрямку. Крім унікальної назви типу (за правилами створення ідентифікаторів), необхідно створити локальні вхідні (I – імпульсний вхід рейкового кола, КО – вхід контролю лампи червоного вогню та GO – жовтого вогню) і вихідні змінні (LG – вихід на лампу жовтого, LK – червоного та LZ – зеленого кольору й T – вихід трансмітера наступного рейкового кола) у FBD-модулі та логіку «перетворення» вхідних змінних у вихідні та записати параметри, як на рисунках 29, 30, 56.

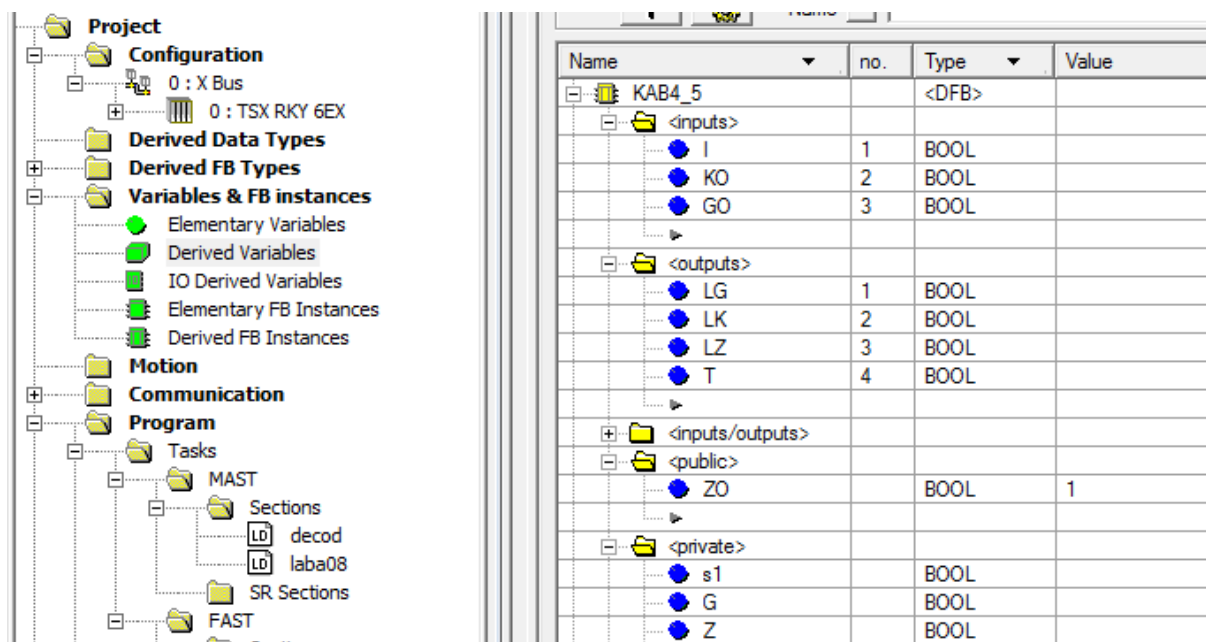


Рисунок 56 – Розроблення DFB-типу «КАВ4\_5» з зазначенням частини змінних і визначенням їхніх типів

Потім komponуємо схему (рисунок 57) з уже відомих компонентів, натискаючи на об'єкт Sec\_KAB4\_5 2 рази, як у вікні на рисунку 30.

Створюємо операторський екран Screen\_laba8 і зображуємо світлофор з розташуванням показань тризначного/чотиризначного автоблокування, аналогічно, як на рисунку 32, і задіємо кожен з кружків для трьох кольорів показань, де будуть наступні



параметри з можливістю задавати в полі Variable локалізовані (рисунок 33) і нелокалізовані змінні.

Після цього студент самостійно будує функціональну модель сигнальної точки для тризначного або чотиризначного автоблокування за схемами двоколійного кодового автоблокування для дільниць з одностороннім рухом поїздів.

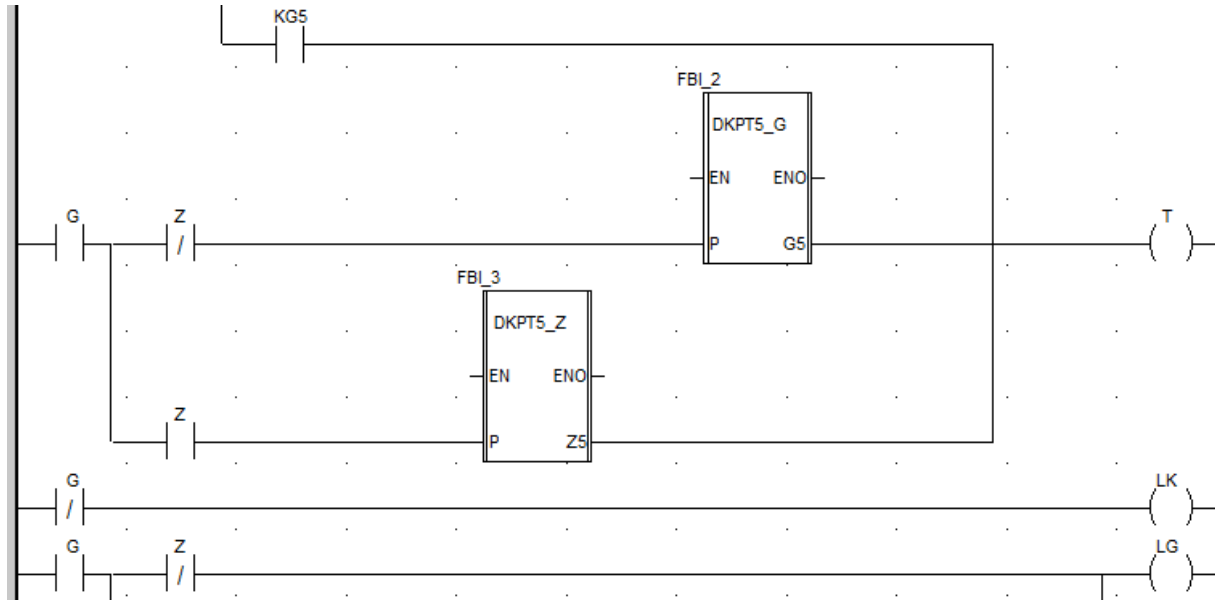
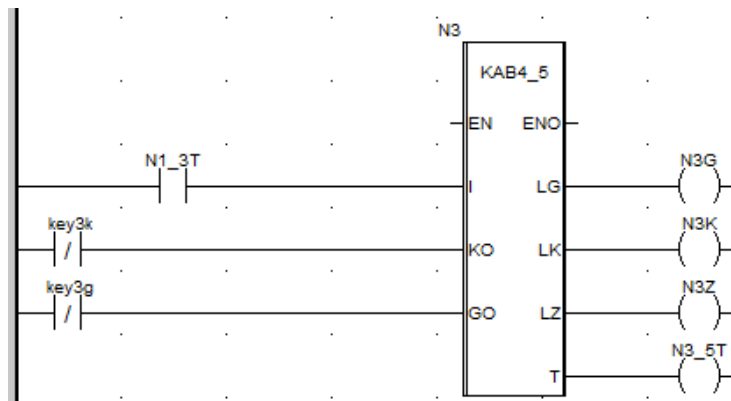


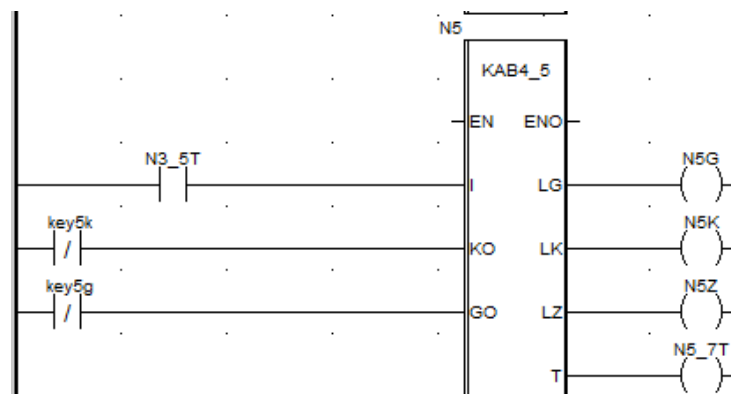
Рисунок 57 – Фрагмент внутрішньої схеми DFB блока типу Sec\_KAB3\_5

Знову компілюємо програму модуля і перевіряємо роботу всіх елементів проекту.

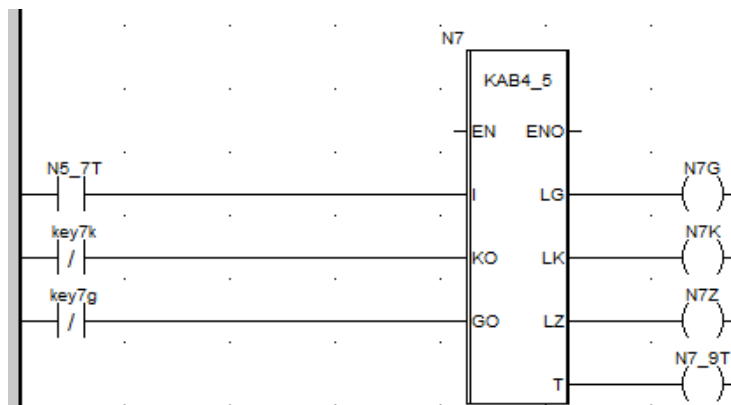
Створивши екземпляри N3, N5, N7 на базі функціонального блока користувача (DFB-типу з ім'ям KAB3\_5), підключаємо до їхніх виходів «реле» з локалізованими змінними (рисунок 58 а, б, в). Змінна N3\_5T символізує вихідний елемент «Т» блока N3 для відправки кодів до рейкового кола 5П.



а



б



в

Рисунок 58 – Схеми підключення екземплярів N3(а), N5 (б), N7 (в)

Блоки будуємо один під одним.

Запускаємо програму, як наведено на рисунках 5, 6. У режимі емуляції роботи ПЛК відображаються червоним кольором розімкнені кола, а зеленим кольором замкнуті (рисунок 59).

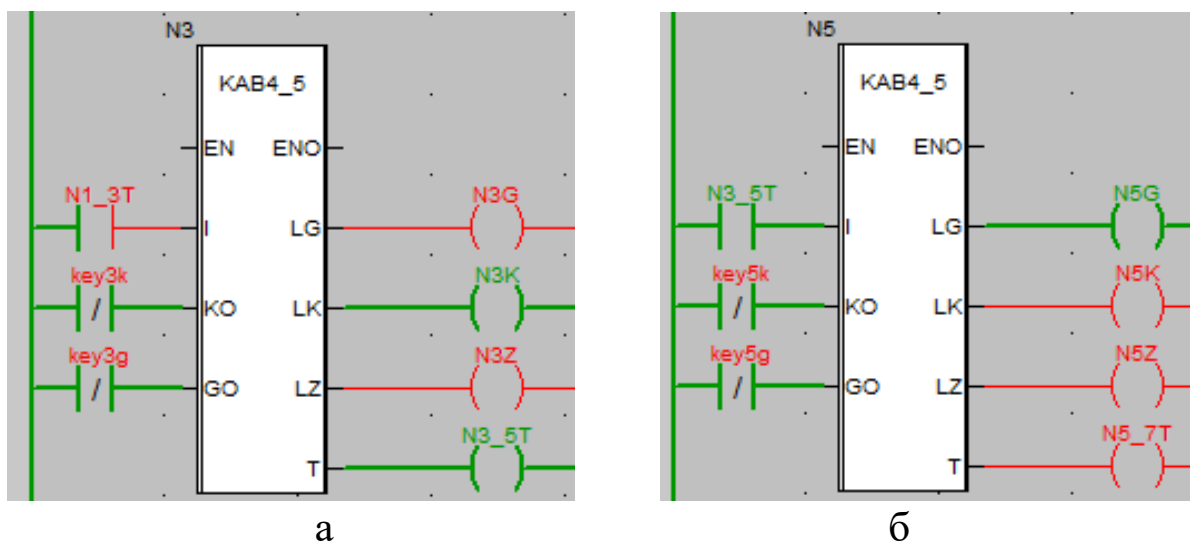


Рисунок 59 – Перевірка роботи блоків АБ N3 (а) і N5 (б)

Студенти, що претендують на рейтинг більше 70 балів, мають реалізувати до кінця лабораторної роботи домашнє завдання рівня від 2 до 4. Для створення чотиризначного кодового автоблокування в прикладах створеного функціонального блока користувача (DFB-типу з ім'ям KAB4\_5) цифра 5 ставиться при генерації кодів з інтервалами КПТШ 5/515/8/815. При встановленні в сигнальній установці, для генерації кодів, КПТШ 7/715/9/915 доцільно ставити у відповідному функціональному блоці користувача DFB-типу з ім'ям KAB4\_7. Типи DFB блоків KAB4\_5 та KAB4\_7 відрізняються тільки зображеними на рисунку 57 модулями DKPT5\_Z та DKPT5\_G параметром вмикання та вимикання реле KG5. Таким чином, у DFB блоці KAB4\_7 мають стояти модулі DKPT7\_Z та DKPT7\_G і змінені параметри вмикання та вимикання реле KG7 (аналог KG5).

Локалізовані змінні «key3k» та «key3g» створені для змін інформації про перегорання червоної та жовтої ламп відповідно на третій, а «key5k» та «key5g» на п'ятій сигнальних точках. Щоб перевірити, чи правильно працює програма, треба встановити в

«1» на локальованій змінній «key3k» ПКМ Force Value 1 біт (рисунок 60).

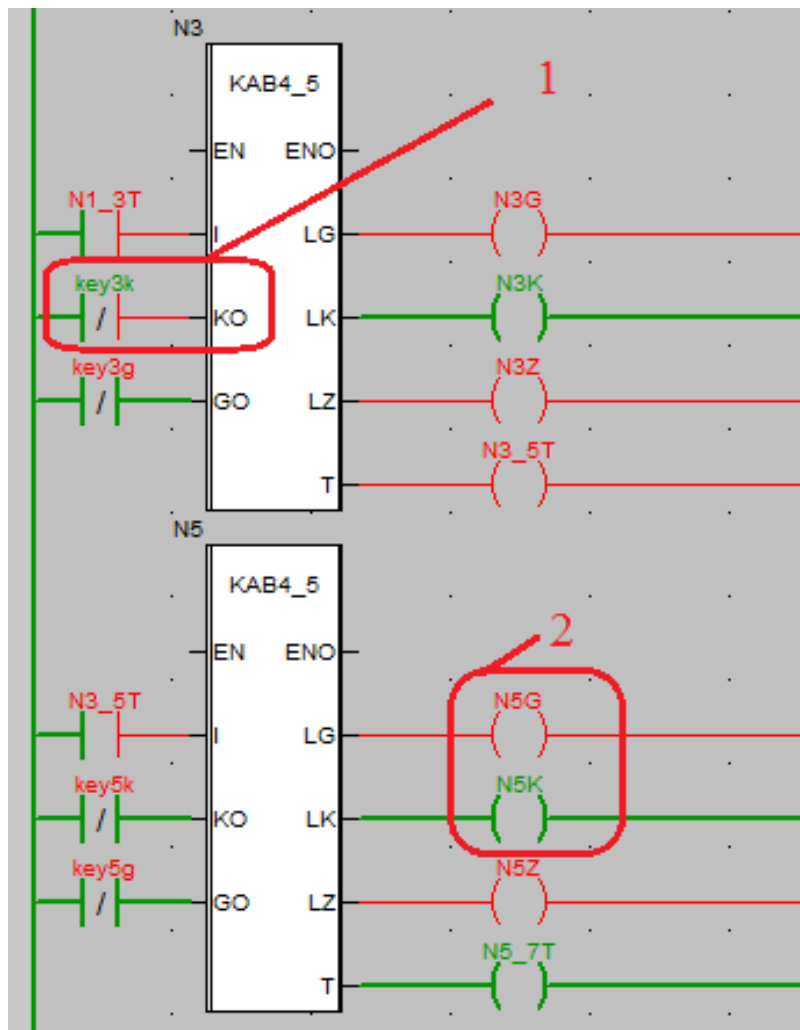


Рисунок 60 – Зміна стану входу «КО» блока N3 та контроль виходів «LG» і «LK» блока N5

При виконанні домашнього завдання найвищого рівня складності буде доцільно переглянути підказки, зображені на рисунках 61, 62.

Студентам, які справились із домашніми завданнями від 1 до 4, індивідуальну роботу може надати викладач.

План виконання роботи:

- 1 Знайти додаткові джерела інформації.
- 2 Вивчити теоретичний і практичний матеріал, необхідний для виконання роботи.
- 3 Відповісти на контрольні питання та виконати домашнє завдання.

4 Спланувати лабораторну роботу згідно з домашнім завданням відповідного рівня.

Контрольні питання:

- 1 Як створити новий блок?
- 2 Як працює прохідна та передвхідна сигнальна точка кодового автоблокування?
- 3 Як підключити три лампи світлофора до блока КАВ3\_5?

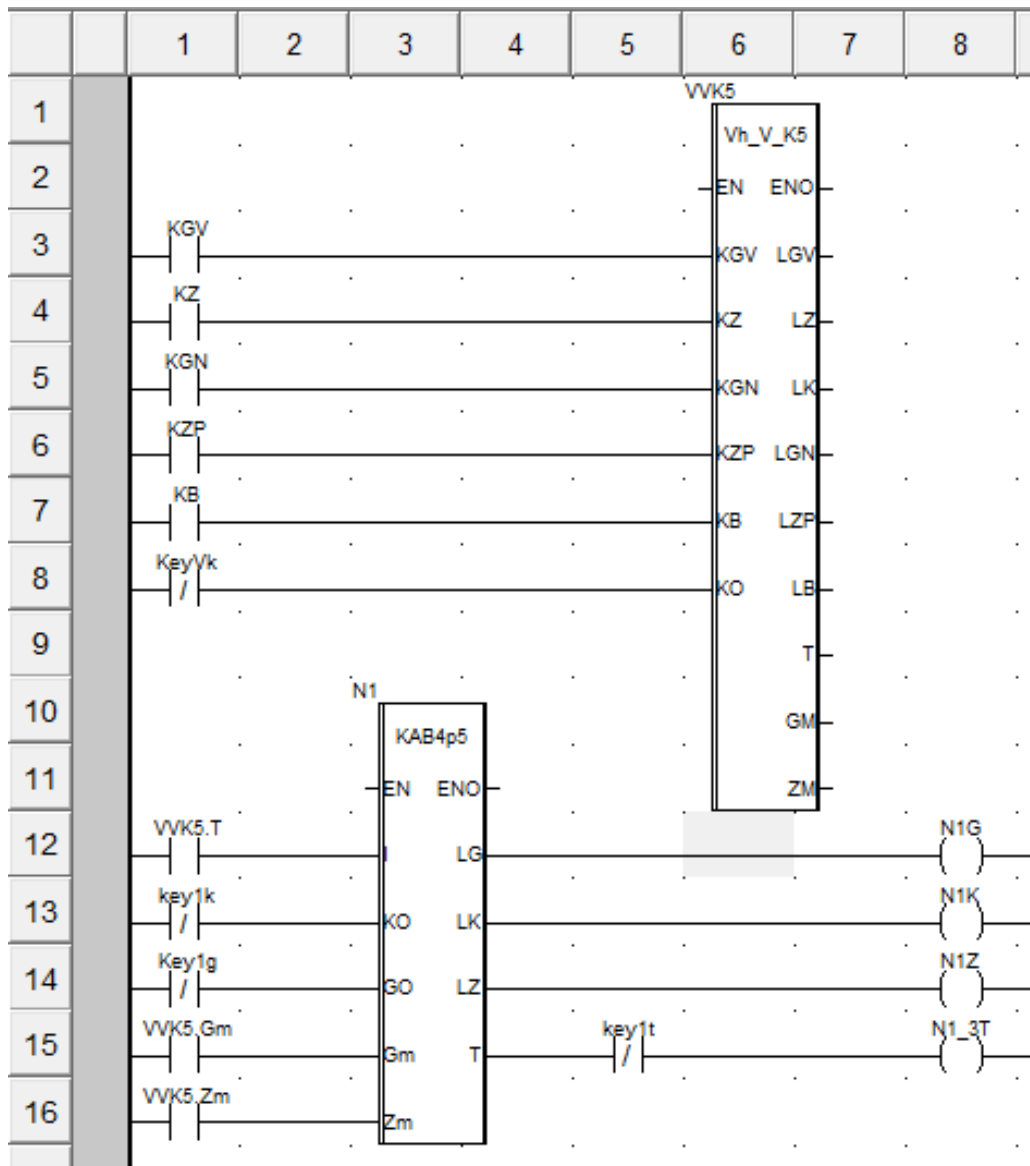


Рисунок 61 – Схема зв'язку екземпляра блока «VVK5» DFB-типу Vh\_V\_K5 з екземпляром блока N1 DFB-типу КАВ4р5 (КАБ 4 – значна, р – передвхідна сигнальна точка з КІТШ 5)

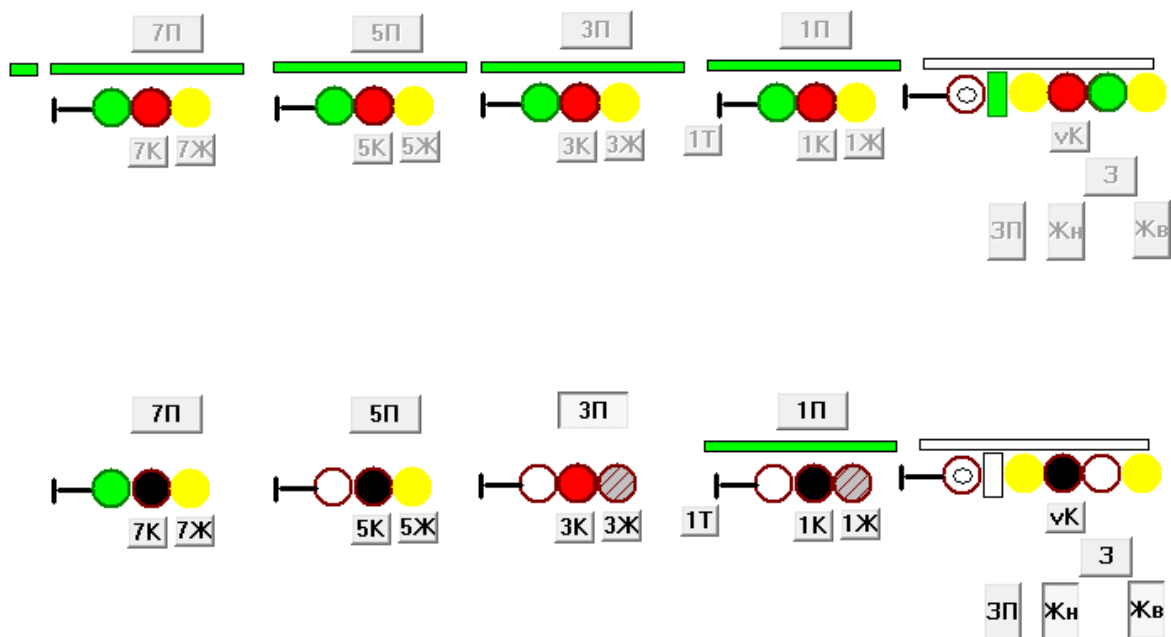
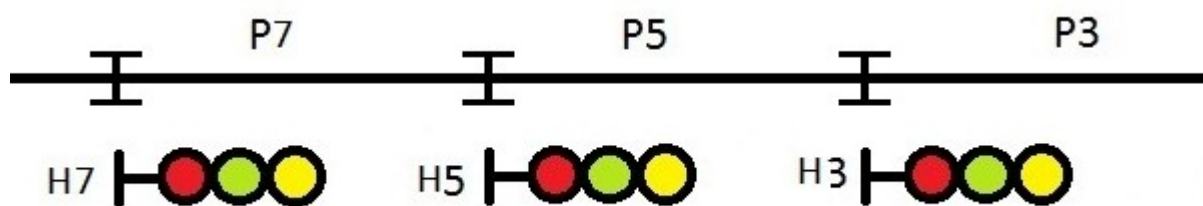


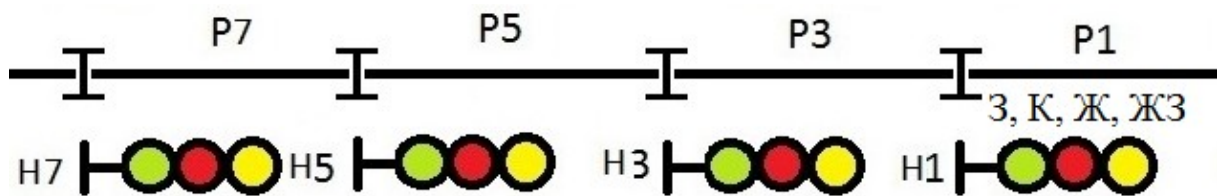
Рисунок 62 – Схема перегону з імітатором вхідного світлофора

Домашнє завдання зі створенням власних DFB:

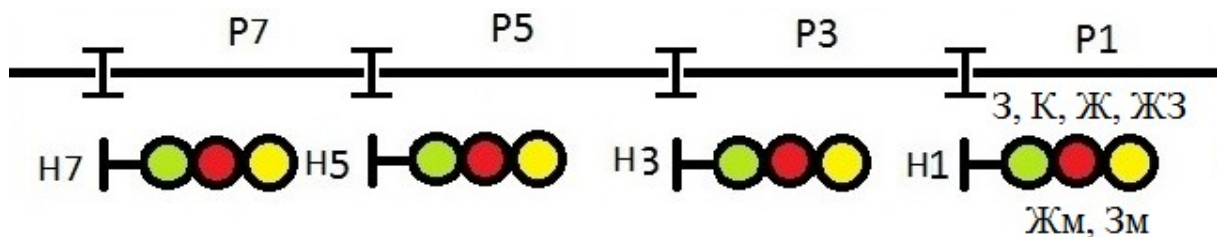
1 Схематично зобразити у вікні Screen\_laba8\_dzE ділянку колії з трьох рейкових кіл Р7, Р5, Р3 і трьох світлофорів Н7, Н5, Н3, з'єднавши їх у єдину технологічну модель роботи тризначного кодового автоблокування за схемою перегону, наведеною нижче.



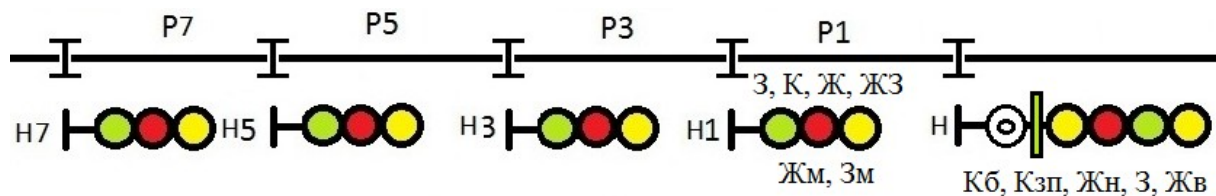
2 Схематично зобразити у вікні Screen\_laba8\_dzD ділянку колії з трьох рейкових кіл Р7, Р5, Р3 і трьох світлофорів Н7, Н5, Н3, з'єднавши їх у єдину технологічну модель роботи чотиризначного кодового автоблокування за схемою перегону, наведеною нижче, з перенесенням червоного показання у випадку перегорання червоної лампи.



3 Схематично зобразити у вікні Screen\_laba5\_dzC ділянку колії з трьох рейкових кіл P7, P5, P3 і чотирьох світлофорів H7, H5, H3, H1, з'єднавши їх у єдину технологічну модель роботи чотиризначного кодового автоблокування за схемою одностороннього перегону, наведеною нижче, з перенесенням червоного показання у випадку перегорання червоної лампи та вимиканням зеленого вогню при показанні «жовтий з зеленим» у випадку перегорання лампи жовтого вогню з динамічною індикацією кодування та інтерактивним введенням зайнятості рейкових кіл P7, P5, P3 і роботи світлофора H1 (шість показань).



4 Схематично зобразити у вікні Screen\_laba5\_dzA ділянку колії з чотирьох рейкових кіл P7, P5, P3, P1 і п'яти світлофорів H7, H5, H3, H1, H, з'єднавши їх у єдину технологічну модель роботи за схемою чотиризначного кодового автоблокування перегону, наведеною нижче з переносом червоного показання у випадку перегорання червоної лампи та вимиканням зеленого вогню при показанні «жовтий з зеленим» у випадку перегорання лампи жовтого вогню з роботою світлофора H1 в режимі передвхідної сигнальної установки з динамічною індикацією кодових сигналів й інтерактивним введенням зайнятості рейкових кіл P7, P5, P3, P1 і роботи світлофора H (шість показань – К, Жв, З, Жн+Жв, Жн+Жв+ЗП, К+Бмиг).



## СПИСОК ЛІТЕРАТУРИ

1 Пупена О.М., Ельперін І.В. Програмування промислових контролерів у середовищі Unity Pro: Навч. посібник.– К.: Ліра-К, 2013. – 376 с.

2 Казаков А.А., Бубнов В.Д., Казаков Е.А. Автоматизированные системы интервального регулирования движения поездов. – М.: Транспорт, 1995. – 320 с.

3 Бойник А.Б., Кошевой С.В., Панченко С.В., Сотник В.А. Системы интервального регулирования движения поездов на перегонах: Учеб. пособие. – Харьков: УкрГАЗТ, 2005. – 256 с.

4 Федоров Н.Е. Современные системы автоблокировки с тональными рельсовыми цепями. – Самара: СамГАПС, 2004. – 132 с.

5 Виноградова В.Ю. Перегонные системы автоматики. – М.: Маршрут, 2005. – 235 с.

6 Данько М.І., Меркулов В.С., Гончаров В.О. та ін. Математичні методи та моделі в розрахунках на ЕОМ: Навч. посібник / За заг. ред. М.І. Данька. — Харків: УкрДАЗТ, 2008. – 172 с.

7 Бутенко В.М., Мойсеенко В.И., Кузьменко Д.М. Компьютерная система управления движением поездов // Залізнич. трансп. України.– 2000.– № 5 – 6. – С. 80 – 82.

8 Поддубняк В. И., Радковский С.А., Бутенко В.М., Мойсеенко В. И. Моделирование состояний объектов систем железнодорожной автоматики // Информационно-управляющие системы на железнодорожном транспорте. – Харьков, 2001. –№ 4. – С. 40 – 44.