

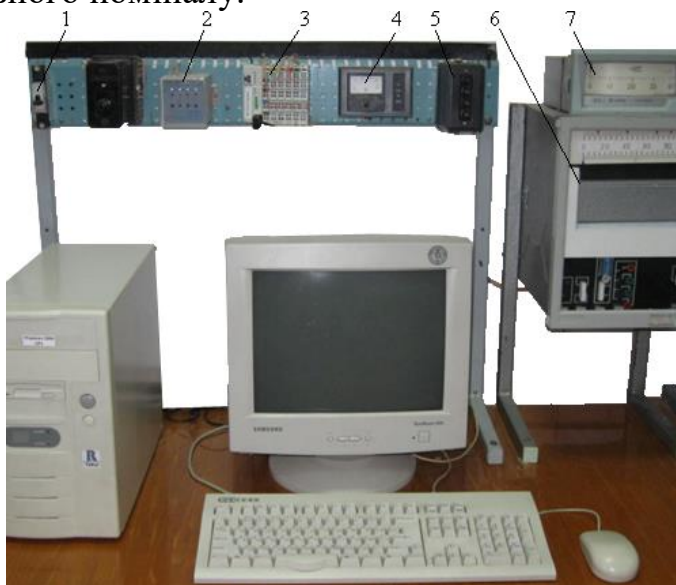
Лабораторна робота №6

Програмування та дослідження роботи контролера WAGO 750-841

Мета: ознайомитися з будовою, придбати навички програмування і роботи з контролером.

Лабораторна установка, рис.8.1, призначена для оволодіння методикою програмування і роботи з контролером 750-841 фірми WAGO.

У лабораторній установці контролер WAGO 750-841 забезпечує рух каретки вторинного приладу шляхом формування керуючого впливу на реле, які згідно з програмою підключаються до входу РП 160 на певний відрізок часу, по черзі, чотири опори різного номіналу.



1 - автомат вхідної напруги; 2 - блок живлення БП 96/36-4; 3 - контролер WAGO 750-841; 4 - міліамперметр вихідного струму контролера; 5 - кнопки управління положенням каретки вторинного приладу РП-160; 6 - мілівольтметр напруги термопари в нагрівальній печі; 7 - вторинний прилад РП-160.

Рисунок 8.1 – Лабораторна установка

Контролер WAGO 750-841

Контролер WAGO 750-841 є оновленою версією Ethernet TCP/IP програмованого контролера сімейства WAGO-I/O-SYSTEM. Оснащений 100-Mbit портом Ethernet, підтримує протоколи ModbusTCP і EtherNet/IP. Зовнішній вигляд WAGO 750-841 приведено на рис.8.2.

Контролер має напругу живлення $\approx 24V$ і USB-порт для підключення до мережі ETHERNET. Експлуатаційний режим контролера відображається за допомогою індикаторів у формі світлодіодів. В деяких випадках, вони є різноколірними (червоними/зеленими або червоними/зеленими/рожевими).

У нижній частині мережного адаптера під захисною ступкою знаходиться конфігураційний і програмний порт, а також операційний перемикач режимів.

У WAGO 750-841 відсутнє традиційне, практично для всіх ПЛК, об'єднувальне шасі. Механічним з'єднувачем для окремих модулів вводу-виводу є стандартна монтажна DIN-рейка.

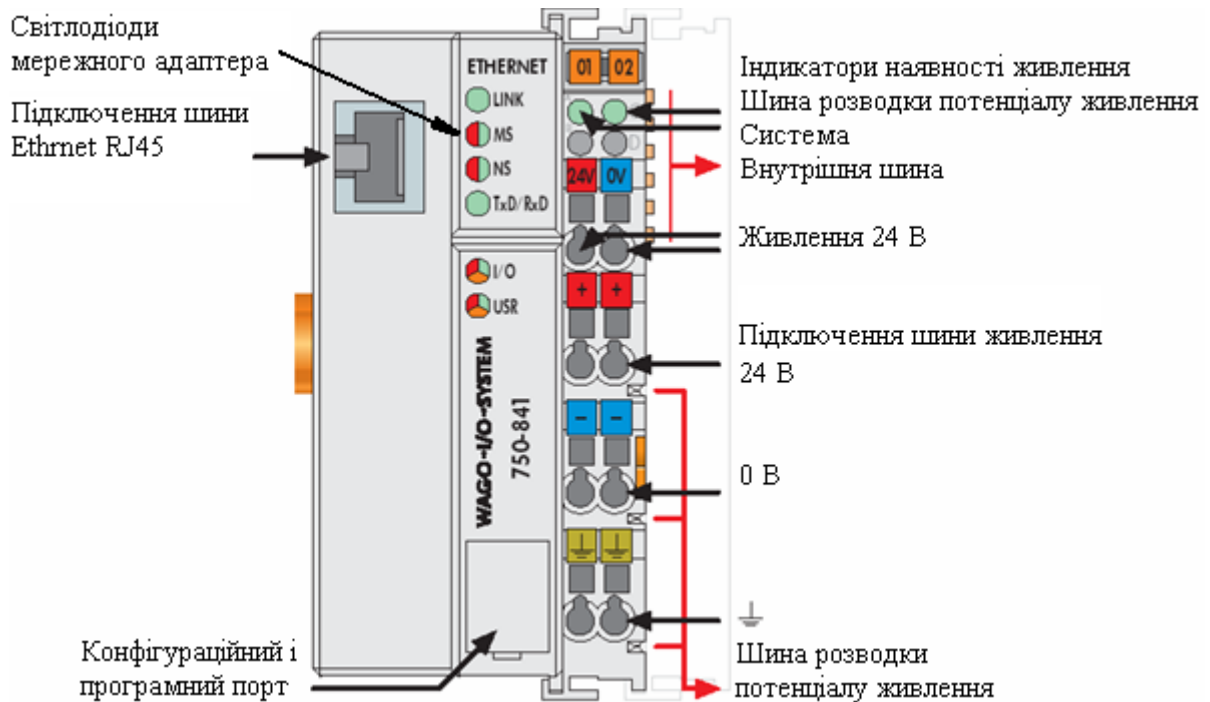


Рисунок 8.2 – Fieldbus контролер ETHERNET TCP/IP

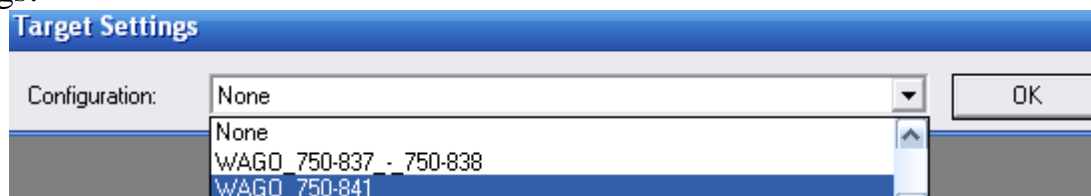
Fildbas-вузол лабораторної установки складається з адаптера WAGO 750-841, модуля дискретного вводу WAGO 750-402, модуля дискретного виводу WAGO 750-504, модуля аналогового вводу WAGO 750-461, модуля аналогового виводу WAGO 750-554 і кінцевого термінального модуля WAGO 750-600.

Виконання лабораторної роботи

Розробити LD-програму управління положенням каретки вторинного приладу РП-160, передбачивши затримку каретки у першому положенні на 7с, у другому на 5с, у третьому на 8с, у четвертому на 4с, а також аварійну зупинку.


Створення проекту

Запустити CoDeSys V2.3 і, використовуючи в меню шлях File-> New, лівою клавішею миші відкрити вікно налаштування цільової платформи Target Settings:



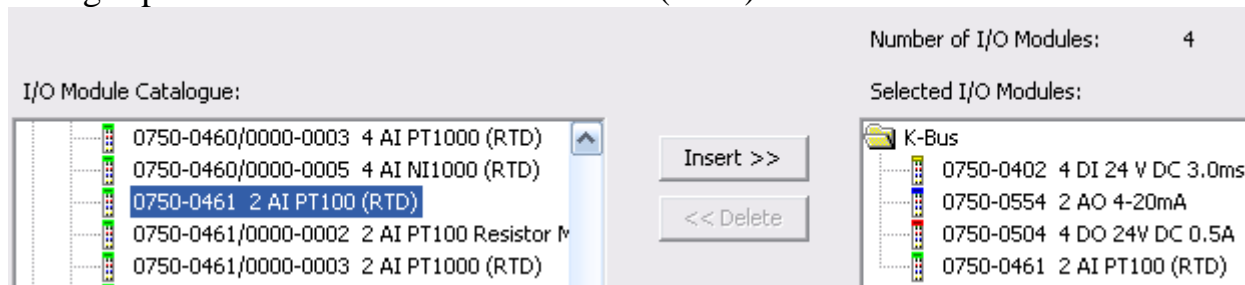
Зі списку Configuration вибрати контролер WAGO-750-841 і натиснути ліву клавішу миші. У діалоговому вікні New POU, що відкриється, позначити LD-мову, вибрати Program, як тип POU, залишити головний шаблон PLC_PRG і натиснути OK.


На екрані з'являться два вікна, ліворуч вікно організатора проекту POU's, а праворуч робоче поле для створення програмного коду.

Щоб здійснити конфігурування контролера, у нижній частині організатора проекту POU's треба активізувати вкладку  (Resources), і лівою клавішею миші позначити PLC Configuration, натиснути праву клавішу і вибрати Edit object. У діалоговому вікні конфігурації PLC, що з'явиться, розкрити Hardware configuration і вибрати K-Bus[FIX]. Натиснути праву клавішу миші, щоб відкрити контекстне меню і вибрати Append Subelement.

У вікні I/O-Configuration, що відкривається, знаходиться список усіх існуючих аналогових (Analog) і дискретних (Digital) модулів вводу (Input) і модулів виводу (Output).

Для реалізації алгоритму управління положенням каретки вторинного приладу використовуються тільки дискретні виходи, проте необхідно при конфігуруванні PLC вказати всі модулі, що встановлені в слотах контролера. У каталозі модулів вводу – виводу, що в лівій частині вікна, двічі натиснути на “Digital Input”, щоб вибрати дискретний модуль DI 0750-0402 4 DI 24V DC 3.0ms, а потім пересунути його в праву частину вікна конфігурації, використовуючи кнопку “Insert”. Після цього, вибрати аналоговий модуль AO 0750-0554 2 AO 4-20mA з каталогу модулів вводу - виводу, натискаючи на “Analog Output”. І знову перемістити його в праву частину вікна конфігурації, використовуючи кнопку “Insert”. Аналогічні дії необхідно виконати з модулем дискретного виводу (Digital Output -> DO 0750-0504 4 DO 24V DC 0.5mA) і аналогового вводу (Analog Input -> AI 0750-0461 2 AI PT 100 (RTD)).

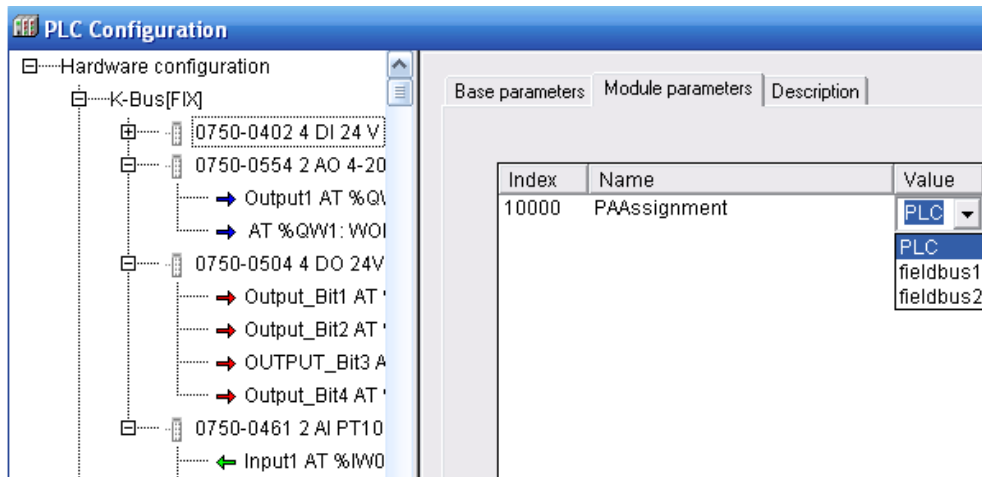


Якщо необхідно видалити деякі модулі, треба вибрати відповідний модуль в правій частині вікна конфігурації і натиснути кнопку “Delete”. Можна змінити позицію модуля, вибираючи і переміщаючи модулі, використовуючи кнопку , що знаходиться у правій частині вікна конфігурації. Кінцевий модуль 750-600 не показують у вікні конфігурації вводу - виводу. При завершенні конфігурування – натиснути кнопку “OK”.


Відкрити K-Bus[FIX] і натиснути ліву клавішу миші. Через Module parameters визначити, як здійснюється управління входами-виходами - за допомогою контролера чи протоколу Ethernet (fieldbus 1 = Modbus/TCP або fieldbus 2 = Ethernet/IP). Оскільки входами і виходами управлятиме контролер, вибрати PLC.

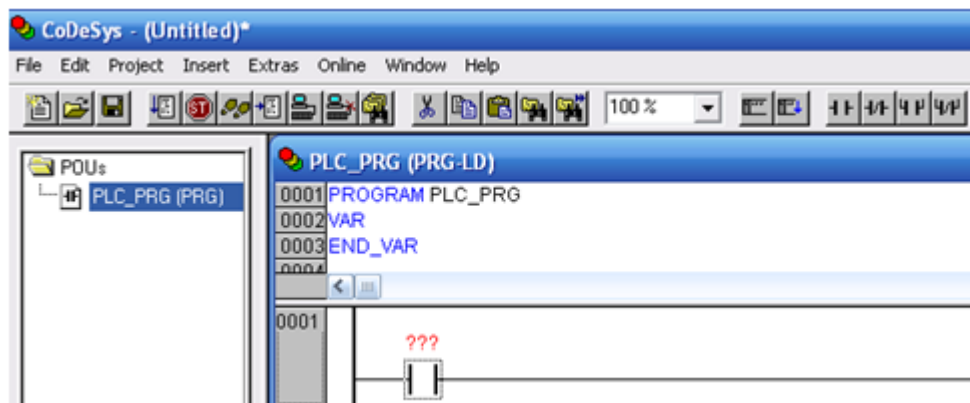
Після цього необхідно присвоїти імена вхідним і вихідним модулям контролера. Оскільки при керуванні використовується модуль дискретного виводу 0750-0504 двічі натиснути лівою клавішею по його назві.

Натискання стрілки, що поряд "AT" відкриє текстове поле, в якому ввести ім'я змінної для дискретного виходу - Output_Bit1 (для адреси %QX2.0), Output_Bit2 (для адреси %QX2.1), Output_Bit3 (для адреси %QX2.2), Output_Bit4 (для адреси %QX2.3), а також для аналогового виходу - Output_1 (для адреси %QW0) і аналогового входу - Input_1 (для адреси %IW0):





Розроблення програмного коду

Для створення програми управління кареткою вторинного приладу у вікні організатора об'єктів перейти на вкладку POU's і подвійно клацнути лівою клав'яшею миші на PLC_PRG (PRG). Відкриється вікно з першою LD-мережею і відповідною панеллю інструментів. Щоб встановити в мережу перший контакт, натиснути лівою клав'яшею миші на кнопку  в панелі інструментів:






Замість знаків питань надрукувати ім'я контакту Start. Натиснути на клавіатурі Enter. У діалоговому вікні, що з'явиться, зберегти найменування (Name: Start) і логічний тип (Type: BOOL). Клас змінної - глобальний (Class:VAR_GLOBAL), натиснути OK. Змінна Start з'явиться у вікні глобальних змінних проекту.

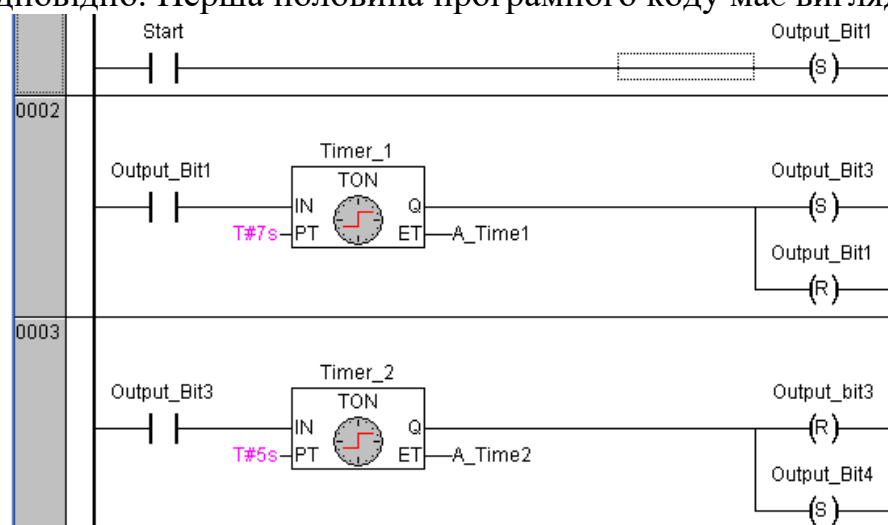
Щоб додати Set-котушку, позначити лівою клав'яшею миші LD-мережу і натиснути лівою клав'яшею миші на кнопку  (coil), яка знаходиться на панелі інструментів. Змінити ім'я котушки на Output_Bit1 (котушка буде зв'язана з першим каналом модуля дискретного виводу).

Створити нову мережу кнопкою  Network (after). Додати контакт (contact) і призначити йому ім'я Output_Bit1. Далі додати функціональний блок. Для цього в панелі інструментів увійти в меню Insert->Function Blocks. У діалоговому вікні (зліва), що відкриється, вибрати категорію: стандартні функціональні блоки (Standard Function Blocks). З таймерів (timer) стандартної бібліотеки (standard.lib) вибрати TON (таймер із затримкою вмикання) і натиснути OK. Щоб визначити ім'я нового екземпляра функціонального блока TON, клацнути лівою клав'яшею миші над зображенням таймера і надрукувати ім'я Timer_1, натиснути Enter і закрити вікно оголошень кнопкою OK. Замінити

три знаки питання на вході РТ константою T#7s, яка відповідає 7с. Цей час можна змінювати в процесі налагодження. Біля виходу ЕТ задати змінну A_time1, натиснути Enter і у вікні оголошення, що відкриється вибрати її тип (Type) – TIME.

На виході функціонального блока додати RESET-котушку кнопкою  і оголосити її, як Output_Bit1. Позначити цю котушку лівою клавішею миші і в панелі інструментів натиснути кнопку , щоб додати паралельну SET-котушку для переміщення каретки в інше положення, і оголосити її змінну, як Output_Bit3.

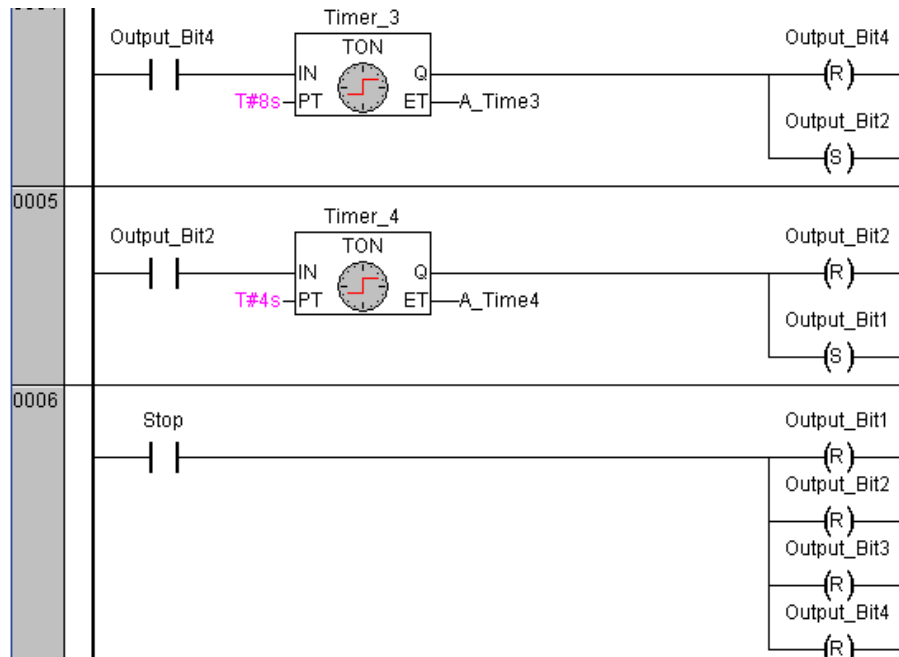
Створити ще одну мережу кнопкою . Додати контакт (contact) Output_Bit3, встановити таймер TON з ім'ям Timer_2. Замінити три знаки питання на вході РТ константою T#5s, а на виході ЕТ встановити змінну A_time2 типу TIME. Розмістити на цій мережі Reset- і Set-котушки з іменами Output_Bit3 і Output_Bit4 відповідно. Перша половина програмного коду має вигляд:



Створити мережу 0004. Додати контакт (contact) з ім'ям Output_Bit4. Встановити таймер TON і змінити його ім'я на Timer_3. На вході РТ надрукувати T#8s, а на виході ЕТ - змінну A_time4 типу TIME. Встановити Reset - і Set-котушки з іменами Output_Bit4 і Output_Bit2.

Створити мережу 0005, яка містить контакт Output_Bit2, таймер TON - Timer_4 з константою T#4s на вході РТ і змінною A_time5 типу TIME на виході ЕТ, Reset- котушку (Output_Bit2) і Set-котушку (Output_Bit1).

Для аварійного зупинення програми створити мережу 0006, додати контакт Stop, який розмикатиме всі дискретні виходи, а також чотири Reset-котушки з іменами Output_Bit1, Output_Bit2, Output_Bit3, Output_Bit4. Друга половина програмного коду має вигляд:



Протягом створення програмного коду сформувався список оголошених змінних:


```

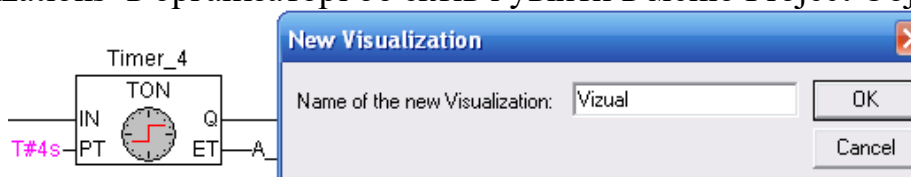
PLC_PRG (PRG-LD)
0001 PROGRAM PLC_PRG
0002 VAR
0003   Start: BOOL;
0004   Timer_1: TON;
0005   A_Time1: TIME;
0006   A_Time2: TIME;
0007   Output_bit1 AT %QX2.0: BOOL;
0008   Output_bit2 AT %QX2.1: BOOL;
0009   Output_bit3 AT %QX2.2: BOOL;
0010   Output_bit4 AT %QX2.3: BOOL;
0011   Timer_3: TON;
0012   Timer_2: TON;
0013   A_Time3: TIME;
0014   Timer_4: TON;
0015   A_Time4: TIME;
0016   Stop : BOOL;
0017 END_VAR

```


Для перевірки програми виконати компіляцію проекту командою меню Project -> Build або клавішею F11. CoDeSys виведе результати компіляції в нижній частині робочого екрану. Якщо є помилки, то перейти у вікно повідомлень і клацнути кнопкою миші на тексті повідомлення. Це призведе до переміщення курсору в місце, де виникла помилка.

Візуалізація

Щоб створити візуалізацію у проекті, необхідно кнопкою  вибрати вкладку Visualizations в організаторі об'єктів і увійти в меню Project Object Add:



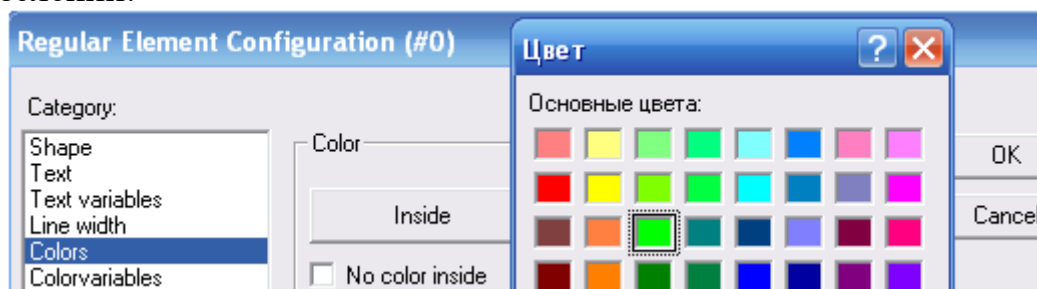
Ввести ім'я візуалізації, наприклад Visual, і натиснути кнопку ОК. Відкриється вікно, в якому створити візуалізацію.

Для візуалізації кнопки Start, в панелі інструментів вибрати прямокутник (Rectangle) . У вікні редактора візуалізації натиснути ліву клавішу миші і розтягнути прямокутник до потрібної висоти і ширини, відпустити клавішу.

Двічі клацнути лівою клавішею миші по прямокутнику. З'явиться діалогове вікно для настройки елементу візуалізації. У Category вибрати альтернативу Text і заповнити поле Contents ім'ям змінної Start:

Перейти в категорію змінних (Variable), клацнути лівою клавішею миші в полі зміни кольору (Change color) і скористатися асистентом вводу - натиснути на клавіатурі <F2> і в діалоговому вікні, що відкриється, зі списку глобальних змінних (Global Variables) вибрати змінну Start:


Перейти в категорію колір (Colors). Натиснути кнопку Inside і задати сірий колір фарбування елементу (колір елементу при хибному значенні змінної Start). За допомогою кнопки Inside визначити колір активного стану змінної, наприклад зелений:



У категорії Input активізувати Toggle variable і ввести змінну Start.


В результаті, при значенні FALSE змінної Start прямокутник відображатиметься сірим, а при значенні TRUE – зеленим. Значення змінюватиметься при кожному «натисненні» кнопки.

Решта елементів вікна візуалізації розробляється за аналогією з методикою створення кнопки Start.

За допомогою кнопки  в панелі інструментів намалювати коло. У настройках Text списку Category у вікні Contents надрукувати Out_1, у настройці - Variable у рядку Change color надрукувати змінну .Output_Bit1, у настройці - Colors кнопкою Inside визначити блакитний колір, а у настройці Alarm color кнопкою Inside - рожевий. Скопіювати створене коло командою в меню Edit-> Copy і вставити його кілька разів командою Edit->Paste для наступних положень механізму Out_2, Out_3, Out_4.

Для кожного кола скоригувати настройки:

- Category-Text-> Contents для другого положення – Out_2, для третього – Out_3, для четвертого – Out_4;
- Category-Variable ->Change color для другого положення - змінна .Output_Bit2, для третього – .Output_Bit3, для четвертого – .Output_Bit4.

За допомогою кнопки  нарисувати прямокутник для клавіші Stop, з такими настройками:

- Category-Text , Contents - текст Stop;
- Category-Variable , Change color - змінна .Stop;
- Category-Input - прапорець Toggle variable активізований, змінна -.Stop;
- Category-Colors , Color - зафарбування Inside сірим кольором, Alarm color червоним кольором.

Намалювати білий прямокутник для таймерів з наступними настройками для положень Out_1, Out_2, Out_3, Out_4:

- Category-Text, Contents- текст Timer_1:%s, Timer_2:%s, Timer_3:%s, Timer_4:%s;

- Category-Variable, Textdisplay- змінні PLC_PRG.A_Time1, PLC_PRG.A_Time2, PLC_PRG.A_Time3, PLC_PRG.A_Time4.

Налагодження програми

Якщо проект без помилок, запустити його в режимі симуляції. Цей режим дозволяє перевірити роботу програми, не записуючи її в контролер. Для цього встановити прапорець в меню Online -> Simulation. Активізувати проект командою Online->Login або натисненням на панелі інструментів кнопку із зображення EOM і контролера. При цьому CoDeSys відразу виділить яскравим синім кольором активні з'єднання і контакти, тобто ті, які мають значення TRUE. Для безпосереднього запуску програми виконати команду Online -> Run або натиснути клавішу F5. Перейти у вікно візуалізації і натиснути кнопку Start. Кола Out_1, Out_2, Out_3, Out_4 почнуть змінювати колір відповідно до роботи програми, а у прямокутниках показується поточний час затримки таймерів. Вимкнення всіх котушок здійснюється кнопкою Stop.

У меню Online відкрити діалог Communication parameters. Натиснути кнопку New для налаштування нового сполучення, вибрати драйвер RS232 і перевірити параметри вікна конфігурації, які містять наступні дані установки: Baudrate=19200, Parity=Even, Зупинка bits=1, Моторола byteorder=No. В меню Online вибрати Communication Parameters, а в меню Gatenay -> Local.

Переконатися в правильності настройки каналу сполучення. Зняти прапорець в меню Online -> Simulation. З'єднання з контролером здійснюється з середовища програмування CoDeSys командою Online->Login в панелі інструментів. Команда Online -> Run запускає проект. Перейти у вікно візуалізації і натиснути кнопку Start. Каретка вторинного приладу РП-160 почне переміщатися згідно з запрограмованим алгоритмом, затримуючись в певному положенні визначений час. Фрагмент робочого поля під час перевірки роботи програми приведен на рис.8.3.

Завдання для самостійної роботи

Модифікуйте програму керування рухом каретки вторинного приладу, таким чином, щоб кожна зупинка каретки підраховувалась, а після 5 зупинок каретка поверталася на початок шкали.

Зміст звіту

Звіт повинен містити назву роботи, короткі відомості про контролер, програму керування кареткою, результати самостійної роботи і висновки.

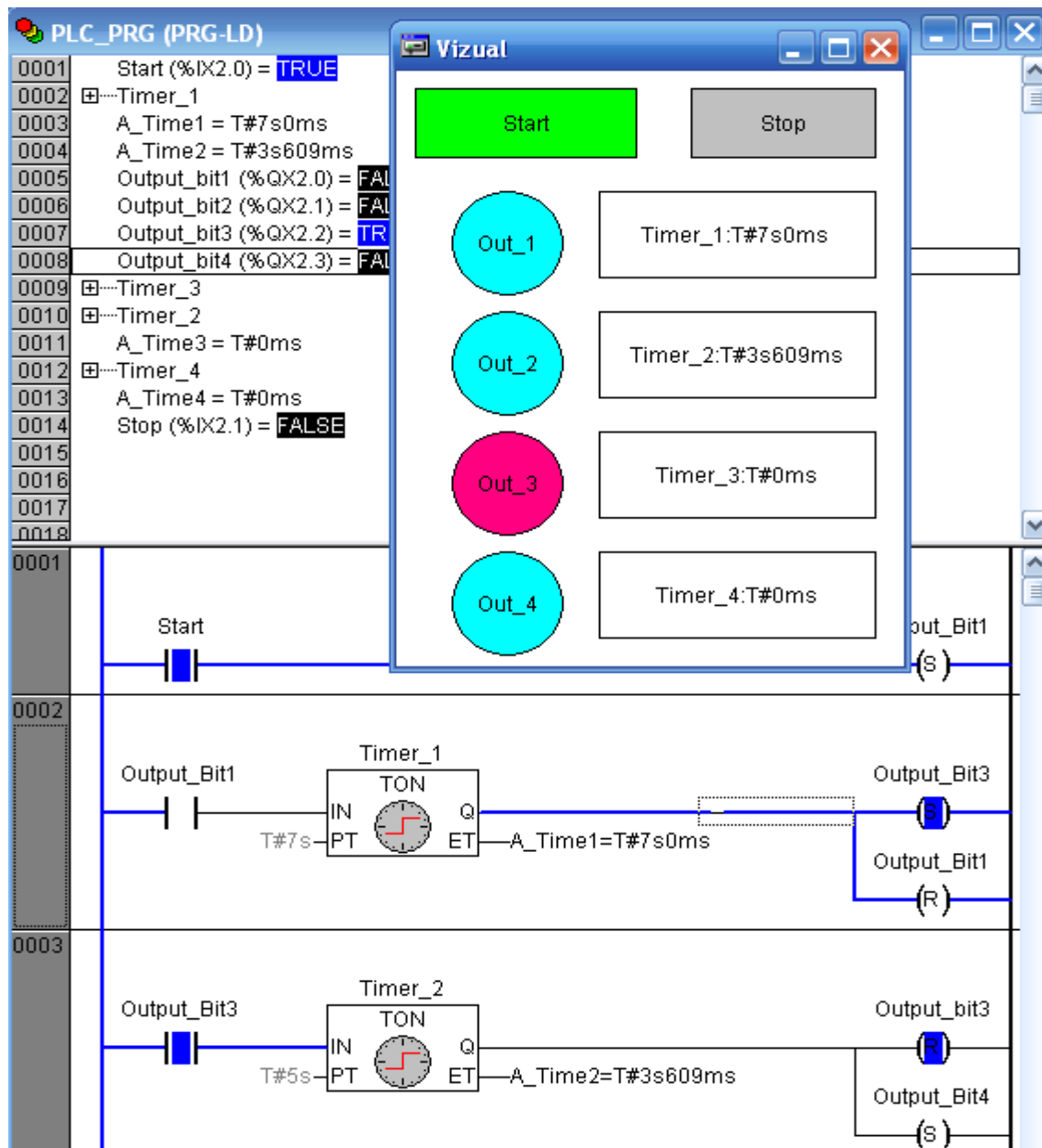


Рисунок 8.3 – Фрагмент робочого поля під час перевірки роботи програми

Контрольні запитання

1. Що являє собою контролер WAGO 750-841?
2. Які модулі використовуються в лабораторній роботі?
3. Як здійснюється з'єднання контролера з ПК і настроювання каналу з'єднання?
4. Чим характеризується інструментальна система програмування CoDeSys?
5. Для чого призначене трасування в CoDeSys?
6. Як здійснюється візуалізація в CoDeSys і, яке її призначення?
7. З яких етапів складається програмування контролера?
8. Як здійснюється зв'язок написаної програми з фізичними входами-виходами контролера?
9. Яке призначення має кінцевий модуль?
10. Як здійснюється налагодження програми?