

Мікропроцесорний контролер ADAM 5510KW

ADAM 5510KW – це базовий контролер групи контролерів фірми Advantech, які відповідають усім вимогам міжнародного стандарту IEC-61131-3 і можуть програмуватися п'ятьма технологічними мовами. Контролер має збільшений відносно ADAM 5510 розмір пам'яті для програмування та збереження проектів, багатозадачне ядро реального часу, вільно визначаєму бібліотеку, потужні налагоджувальні можливості, відкриту стандартну комунікаційну систему – Modbus, редагування у реальному масштабі часу, RS – 232/485 – комунікаційні можливості, вбудовані годинники реального часу та засоби безпеки, зручний інструментальний засіб конфігурування. Габаритні розміри базового блока складають 110×231× 75мм (В×Ш×Г).

Мікроконтролер ADAM 5510KW, як і всі контролери серії ADAM 5000 складається з двох основних частин - базового блока та модулів вводу – виводу, які є загальними для всіх контролерів серії ADAM 5000, рис.5.6.

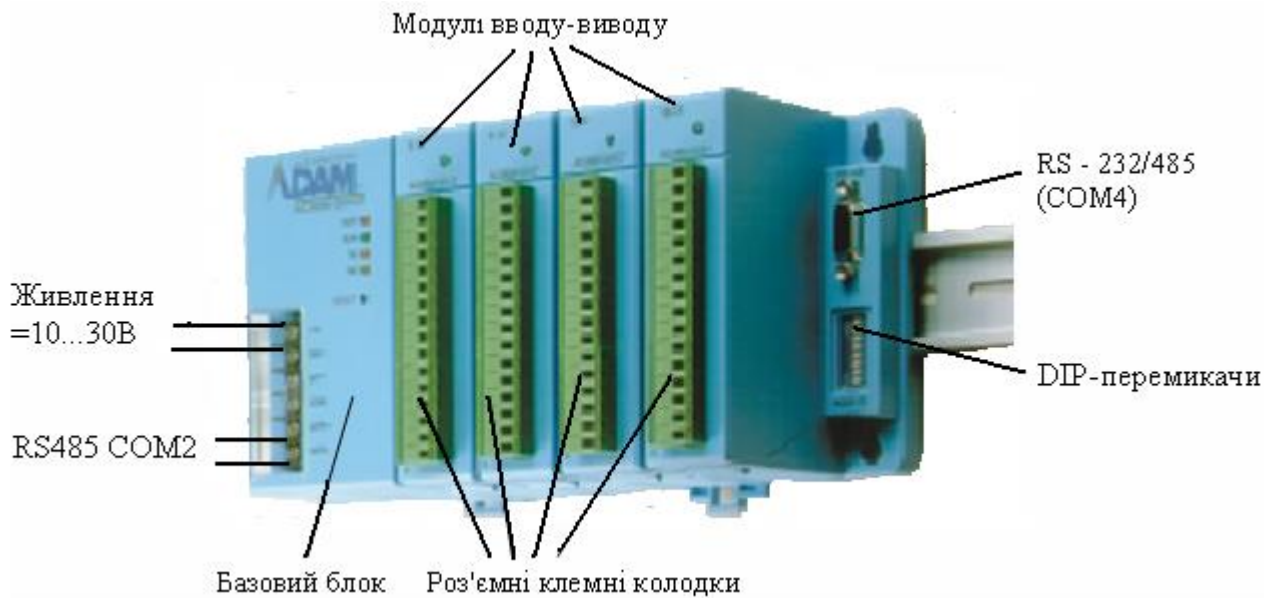


Рисунок 5.6 – Мікропроцесорний контролер ADAM 5510KW

Базовий блок має у своєму складі процесорний модуль, перетворювач напруги, 4-слотову пасивну об'єднувальну панель та серійні комунікаційні порти RS-232, RS-485.

Тип центрального процесора CPU - 80188-40, 16-розрядний. ADAM 5510KW має вбудовану Flash-пам'ять на 1,5MB і статичну оперативну пам'ять (SRAM) на 640KB для завантаження операційної системи та збереження даних.

Це більше ніж у традиційних PC-сумісних контролерів, тому дає змогу збільшувати розмір програми.

Операційна система – ROM-DOS.

ADAM 5510KW, в якості системи виконання реального часу, має програму KW PROCONOS, яка забезпечує виконання одночасно до 16 задач.

ADAM 5510KW має чотири серійні комунікаційні порти, три з яких розташовані так само, як і в ADAM 5510, а четвертий знаходиться з права над DIP-перемикачами.

Ці порти дозволяють програмувати та конфігурувати контролер, контролювати і об'єднувати віддалені пристрої вводу – виводу. Порт COM1 підтримує роботу в режимі інтерфейсу RS-232. До нього приєднується головний комп'ютер за допомогою з'єднувача DB-9.

Порт COM2 підтримує роботу в режимі інтерфейсу RS-485, приєднується до головного комп'ютера за допомогою проводів і клемника через RS-232/RS-485-перетворювач. Завдяки протоколу Modbus контролер можна інтегрувати в АСКТП і об'єднати з більшістю існуючих SCADA-систем.

Порт COM3 – порт налагодження, використовується тільки для системного обслуговування, його інтерфейс - RS-232, приєднувач DB-9. Цей порт дозволяє програмувати, конфігурувати та шукати несправності у контролері з головного комп'ютера.

Порт COM4 підтримує RS-232/485 і призначений для роботи з віддаленими модулями вводу-виводу серії ADAM-4000 (до 255 точок), рис.5.7.

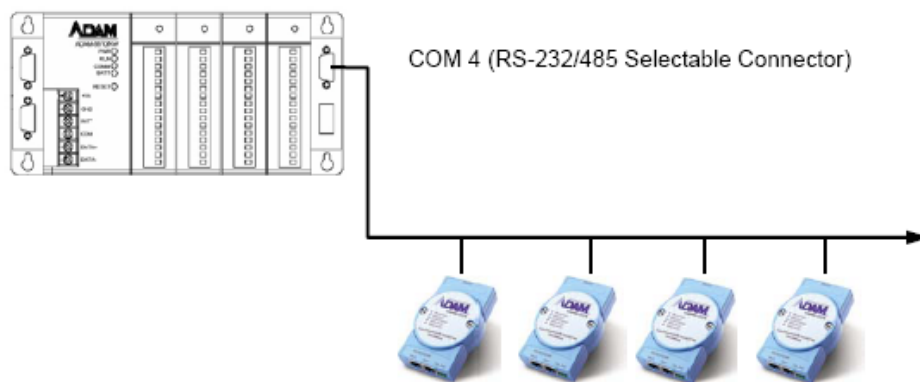


Рисунок 5.7 – Приєднання контролера до віддалених модулів вводу-виводу

Потрібний інтерфейс встановлюється за допомогою перемикача підключеного до цього порту. Порти ADAM 5510KW задовольняють всім вимогам комунікації та інтеграції і допомагають розширити системну структуру.

Інтерфейс RS-232 забезпечує швидкість передачі до 38400біт/с на відстані до 15.2 м. Режим роботи – асинхронний повний дуплекс, з'єднання точка - точка. Інтерфейс RS-485 має швидкість передачі до 38400 біт/с на відстані до 1220м. Режим – напівдуплексний (багатопрхідний).

За допомогою DIP – перемикачів і відповідно до табл.5.1-5.2 встановлюється ідентифікатор пристрою (перемикачі 1-5), протоколи портів COM1, COM2 (перемикач 6) та швидкість обміну даними (перемикачам 7і 8).

Таблиця 5.1 Установлення швидкості обміну даними у бодах

DIP 7	DIP 8	Baud Rate
ON	ON	9600 bps
OFF	OFF	9600 bps
ON	OFF	19200 bps
OFF	ON	38400 bps

ADAM 5510KW має чотири слоти для встановлення модулів вводу – виводу, які нумеруються від 0 до 3.

Таблиця 5.2 Установлення протоколів

DIP 6	COM 1	COM 2
OFF	MULTIPRO Protocol	MODBUS Protocol
ON	MODBUS Protocol	MULTIPRO Protocol

Канали модулів не залежно від місця їх розташування починають нумеруватися від 0.

ADAM 5510KW, як і ADAM 5510 має годинники реального часу та сторожовий таймер. Додатково у нього є три сторожові схеми – сторож операційної системи, сторож KW-додатку та сторож комунікаційного обслуговуючого пристрою Modbus. Це збільшує надійність пристрою і дає йому переваги при розв'язанні складних задач автоматизації.

ADAM 5510KW можна використовувати в якості дублюючого пристрою. Для цього необхідно іншій резервний контролер підключити паралельно основному і завантажити в нього таку саму програму, а також присвоїти йому відповідний ідентифікатор. Тоді у випадку помилки в комунікації або передачі даних в основному контролері, активізується резервний і продовжить функціонування системи управління.

ADAM 5510KW розроблений під напругу 24В нестабілізованого постійного струму, але підключати його можна до будь-якого джерела з напругою в межах +10 - +30В.

На передній панелі контролер має чотири індикатори:

PWR - індикатор напруги, коли контролер несправний індикатор блимає.

RUN – індикатор виконання програми, блимає безперервно, коли програма виконується.

COMM – індикатор комунікації, блимає кожного разу, коли головний комп'ютер обмінюється з контролером.

BAT – індикатор стану акумулятора, горить коли акумулятор резервування SRAM має низьку напругу.

ADAM 5510KW забезпечує багатозадачний та багатопрограмний режим роботи. Його комунікація, обробка даних і доступ до модулів вводу-виводу здійснюється незалежно одно від одного, тому якість функціонування та ефективність цього контролера краще ніж у традиційних PLC.

Монтаж контролерів здійснюється на вертикальну стінку за допомогою гвинтів або на DIN-рейку. Модулі вводу-виводу встановлюються у різні вільних слотів базового блоку легким натисненням. Приєднання проводів до клемників модулів – гвинтове. Клемники на модулях мають фронтальне розташування. Для зручності монтажу вони приєднуються за допомогою різнімів[27].

5.2.1. Програмування контролерів групи ADAM 5510KW

Контролери групи ADAM 5510KW програмуються технологічними мовами стандарту IEC- 61131-3. Це сталося завдяки купівлі фірмою Advantech

прав на використання програмного продукту KW MULTIPROG у німецького розробника KW-Software GmbH.

KW MULTIPROG – це вбудована основа Softlogic – контролера, яка використовується при автоматизації об'єктів з 1991 року. Вона підтримує усі стандартні мови програмування, причому в одному проекті можуть використовуватися усі мови.

Завдяки системі програмування MULTIPROG, контролер має напередвизначену бібліотеку великої кількості функцій, за допомогою якої легко створити у проекті таймер, перетворити тип змінної або перетворити послідовність строкової змінної і т. ін. Крім того, користувач може створити власний функціональний блок, використовуючи функції бібліотеки.

Графічний редактор дозволяє вільне розміщення об'єктів на робочому полі проекту, вибирати мову програмування та використовувати бібліотеку функцій. Він підтримує змішування графічних мов в одному робочому листі, розміщення об'єктів в розмірній сітці, вставлення нових елементів в існуючі мережі, переміщення об'єктів або мереж, показ графіків функцій та функціональних блоків для уявлення їх поведінки, різні кольори бібліотечних функцій та функціональних блоків.

Текстовий редактор дозволяє легко і швидко створювати програми. Для цього існують різні функціональні можливості:

- підсвічування синтаксису вказує на ключові слова програми;
- майстер редагування полегшує редагування, вказуючи на доступні елементи і блоки.

MULTIPROG має могутній засіб налагодження, діагностування, моделювання та функціонального втручання із зручним інтерфейсом користувача. За допомогою MULTIPROG перевірка програми контролера здійснюється на емуляторі без участі самого контролера.

MULTIPROG підтримує оперативне редагування програми, завдяки чому можна вносити зміни і завантажувати їх в контролер не зупиняючи процесор.

MULTIPROG запускається з головного комп'ютера. Коли програму створено вона завантажується в контролер через порт COM1 і контролер починає працювати.

MULTIPROG працює разом з програмованою операційною системою реального часу ProConOs (Programmable Controller Operating System), яка вбудована у контролер[27].

5.2.1.1 Методика створення проекту у системі програмування KW MULTIPROG

Створити SFC-мовою проект керування роботою світлофора, який встановлений на перехресті доріг. Світлофор має перебувати в чотирьох станах: зеленому, червоному, жовтому і жовто-червоному. Тривалість зеленого і червоного станів 10с, а жовтого і жовто-червоного – 3с. Світлофор вмикається в роботу дворазовим натисненням кнопки «Пуск». Цикли роботи світлофора підраховує лічильник. Після 10 циклів роботи світлофор вимикається а лічильник скидається. На першому етапі загоряється зелене світло правого світлофора і червоне лівого, на другому етапі загоряється жовте на правому світлофорі і жовто-червоне на лівому, на третьому загоряється на правому

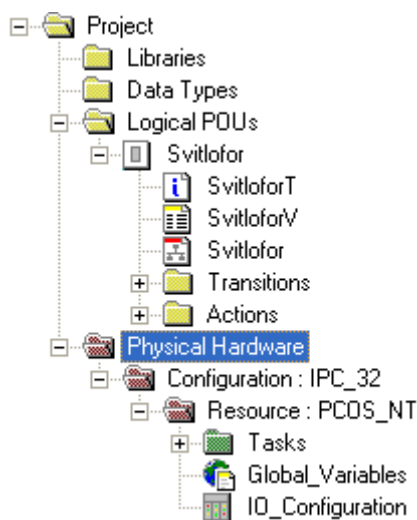
світлофорі червоне і зелене на лівому, на четвертому етапі загоряється жовто-червоне на правому світлофорі і жовте на лівому.

Створення проекту

Створення проекту здійснюється після запуску програми MULTIPROG V3.3. Використовуючи в меню шлях File→ New Project, двічі клацнути по піктограмі Project Wizard у вікні New Project.

За замовчуванням шлях створення проекту вводиться автоматично. Для вибору іншого шляху у вікні Project Wizard натиснути кнопку з трьома крапками і у діалозі Select Directory вибрати теку для нового проекту. Ввести ім'я першого організатора програми (POU), наприклад Svitlofor, вибрати SFC-мову. Ввести ім'я конфігурації, наприклад Configuration, і вибрати тип конфігурації проекту IPC_32, тому що компілятор генерує intel- код для системи виконання ProConOS 3,2. Ввести ім'я ресурсу – Resource і вибрати його тип - PCOS_NT. Ввести ім'я задачі –Task, і вибрати її тип – Cyclic.

Після натиснення кнопки Next на екрані з'явиться зміст зроблених установок. Якщо помилок немає натиснути Finish, в іншому разі, користуючись кнопкою Back, скоригувати кроки проекту де є помилки. Створений проект Svitlofor з'явиться у вікні проектного дерева у піддеревах Logical POU's і Physical Hardware. В меню дерева проекту розкрити теку Physical Hardware:



Для модифікації імені конфігурації правою клавішею миші клацнути по Configuration:IPC_32 і у меню, що з'явиться вибрати Properties. Надрукувати ім'я конфігурації, наприклад ADAM 5510KW_Project, у вікні, що відкриється, і натиснути ОК.

Після цього ім'я конфігурації в дереві проекту зміниться на ADAM 5510KW_Project: IPC_32, при цьому IPC_32 додається автоматично.

В меню дерева проекту клацнути правою клавішею по Ressours: PCOS_NT, а у підменю, що з'явиться, клацнути по “Properties...”. Змінити ім'я на ADAM 5510KW і натиснути ОК.

Для конкретизації цільового об'єкту завантаження - симулятор реального часу чи комунікаційний порт реального ADAM 5510KW, необхідно встановити ресурс.

Клацнути правою клавішею миші по ADAM 5510KW: PCOS_NT. У підменю, що відкриється, активізувати “Settings...”. На екрані з’явиться діалогове вікно: PCOS_NT, в якому:

COM- вказує у який серійний порт відправлено програму;

Simulation- вказує в який симулятор відправлено програму;

Baud- конкретизує швидкість компіляції у Бодах через порт;

Stopbits- конкретизує кількість стопових біт при комунікації через порт;

Databits- конкретизує кількість бітів, що передаються через порт;

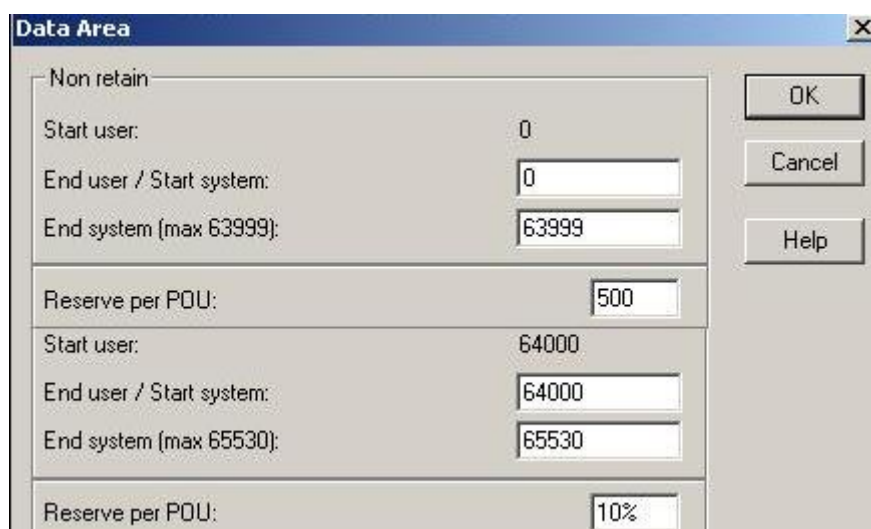
Parity- вказує біт парності при передачі через вибраний порт;

Timeout- конкретизує інтервал часу (мс), протягом якого система програмування з’єднується з технічними засобами.

Для установлення порту контролера необхідно у каталозі «Port:» активізувати альтернативу COM1.

Для установлення швидкості компіляції необхідно у таблиці «Baud:» зберегти вказані уставки.

Натиснути кнопку “Data Area” і визначити області пам’яті для користувача, системних прапорців і резерву POUs, відповідно до даних наведених у діалоговому вікні “Data Area”:



Для редагування робочого листка конфігурації модулів вводу-виводу (В/В) використовується діалог “ I/O_Configuration”, який звичайно містить початок і кінець адреси модулів вводу-виводу, декларацію пристрою (ім’я драйвера та адреса пам’яті) і т.ін.

Щоб відкрити діалогове вікно, у проектному дереві двічі клацніть по “I/O_Configuration”.

Якщо у вікні “ I/O_Configuration” будуть установлені непотрібні модулі вводу “IN” і виводу “OUT”, то для видалення необхідно їх по черзі активізувати лівою клавішею миші і натиснути кнопку “Delete”(Видалити).

Припустимо, що комплектація контролера складається з модуля аналогового вводу ADAM 5018 у слоті 0 і модуля дискретного вводу-виводу ADAM-5050 у слоті 1. Тоді у діалоговому вікні “ I/O_Configuration” необхідно вибрати вкладку “INPUT” (Вхід) і натиснути кнопку “Add”(Додати);

На екрані з'явиться діалогове вікно Add I/O Group. У полі Name присвоїти модулю ADAM 5018 ім'я AI_5018 (Тут AI – скорочено аналоговий ввід), а у полі Task: вибрати Task.

Для установлення логічної адреси модуля вводу-виводу існує три параметри:

- стартова адреса;
- місткість каналів модуля;
- кінцева адреса.

Для стартової адреси система сама знайде вільну адресу.

Місткість каналів модулів визначається їх призначенням, табл.5.3.

Кінцеву адресу система розраховує автоматично.

Таблиця 5.3 Місткість каналів модулів вводу-виводу

Тип модуля	Кількість каналів	Місткість
Аналоговий ввід	8	16 бітів
Аналоговий вивід	4	8 бітів
Дискретний ввід	16	2 біти
Дискретний вивід	16	2 біти
Релейний вивід	8	1 біт

Оскільки у аналогового модуля вводу ADAM-5018 місткість каналів складає 16 бітів, у полі початкової адреси має бути 0, а у полі місткість – 16. При цьому кінцева адреса (End address) – 15 визначається автоматично після установлення Board I/O Module.

При установленні Board I/O Module, вибрати ADAM-5510KW, щоб забезпечити підтримку модулів ADAM-5000. Якщо модулі вводу-виводу серії ADAM- 4000 необхідно вибрати Read Modbus slave.

Після вибору Board I/O Module натиснути кнопку Driver parameter, щоб забезпечити можливість установлення модулів серії ADAM-5000.

Оскільки вибрано ADAM-5510KW, то ім'я драйвера за замовчуванням – ADAMDrv.

Місце розташування модуля ADAM-5018 вказує на те, що адреса слота установлюється як 0. У полі Module name вибрати ADAM-5018 [T/C]. Тут 7-кількість каналів, "T/C"- скорочено визначає аналоговий ввід сигналів термопар.

Після вибору імені модуля система автоматично вибере тип даних – WORD.

Щоб установити серійний порт натиснути кнопку "ADAM5510KW system setup" і вибрати для модуля V/B Data Format : NOPARITY,8.1.

Натиснути ОК і конфігурація модуля V/B буде показана у відповідному вікні.

Щоб додати модуль дискретного вводу-виводу треба вибрати вкладку OUTPUT у діалоговому вікні "Configuration I/O" і натиснути кнопку "Add...". З'явиться діалогове вікно Add I/O Group. Модифікувати в ньому дискретний модуль ADAM-5050 іменем: DIO_5050, а в полі Task: вибрати Task і визначити логічну адресу (Logical addressed), як - Start address: 0, Length:2, а панель

модуля В/В (Board I/O Module) і параметр драйвера (Driver parameter) за аналогією з модулем вводу.

При цьому у полі Name за замовчуванням з'явиться ADAMDrv.

Ідентифікатор пристрою (Device ID): 1.


Оскільки ADAM5050 розташований у слоті 1, то у полі Slot: записана 1. Ім'я модуля В/В показується у полі Module name: ADAM 5050 [16UDIO]. Тут 16 – кількість каналів, DIO – дискретний ввід-вивід.

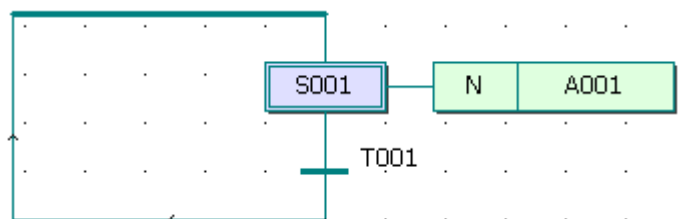
Щоб установити серійний порт натиснути кнопку “ADAM5510KW system setup” і в Data Format вибрати для модуля В/В - NOPARITY,8,1.

Натиснути кнопку OK і ADAM-5050 потрапить до списку модулів OUTPUT.


Після визначення модулів контролера здійснюється розробка програмного коду.

Розробка коду програми


В проектному дереві відкрити теку Logical POU's, вибрати курсором вкладку програмного коду Svitlofor і двічі натиснути ліву клавішу миші. Відкриється робоче поле для створення SFC-коду, на якому позначити курсором місце початку програми. В панелі інструментів натиснути ліву клавішу миші на кнопці  (Creat step transition sequence) і на робочому полі з'явиться SFC-схема, яка містить крок S001 з дією A001 і перехід T001, а також повернення назад:

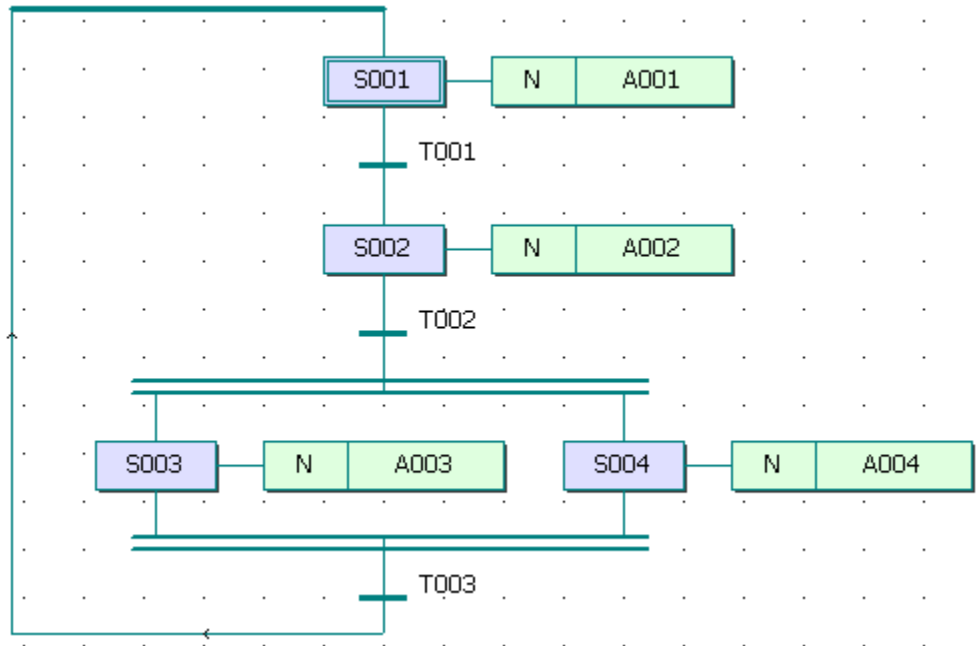



Якщо з'явився не початковий крок, а звичайний, тобто прямокутник кроку створений не подвійною лінією, а одинарною, то двічі натиснути ліву клавішу миші на прямокутнику і у діалоговому вікні Step, що з'явиться, активізувати Initial step. Закрити вікно Step, натиснути OK. На робочому полі звичайний крок перетвориться у початковий.


Щоб додати другу пару крок - перехід, призначену для програмування вихідної фази роботи світлофорів (в одному напрямі руху – зелене світло, а в іншому – червоне), натиснути ліву клавішу миші спочатку на переході, а потім в панелі інструментів на кнопці  (Creat step transition sequence). Існуюча SFC-схема збільшиться на один крок S002 і перехід T002.

Оскільки світлофор забезпечує регулювання руху на перехресті, послідовна мережа кроків і переходів повинна мати розгалуження на дві паралельні гілки, в яких будуть програмуватися фази роботи двох світлофорів.

Для створення паралельних гілок розгалуження позначити курсором другий перехід T002 SFC-схеми і в панелі інструментів клацнути по кнопці  (Insert Simultaneous/Alternative Divergence). У діалоговому вікні Divergence, що з'явиться, у полі Branches Count надрукувати кількість гілок - 2 і натиснути OK. Програмний код у робочому полі набуває вигляд:



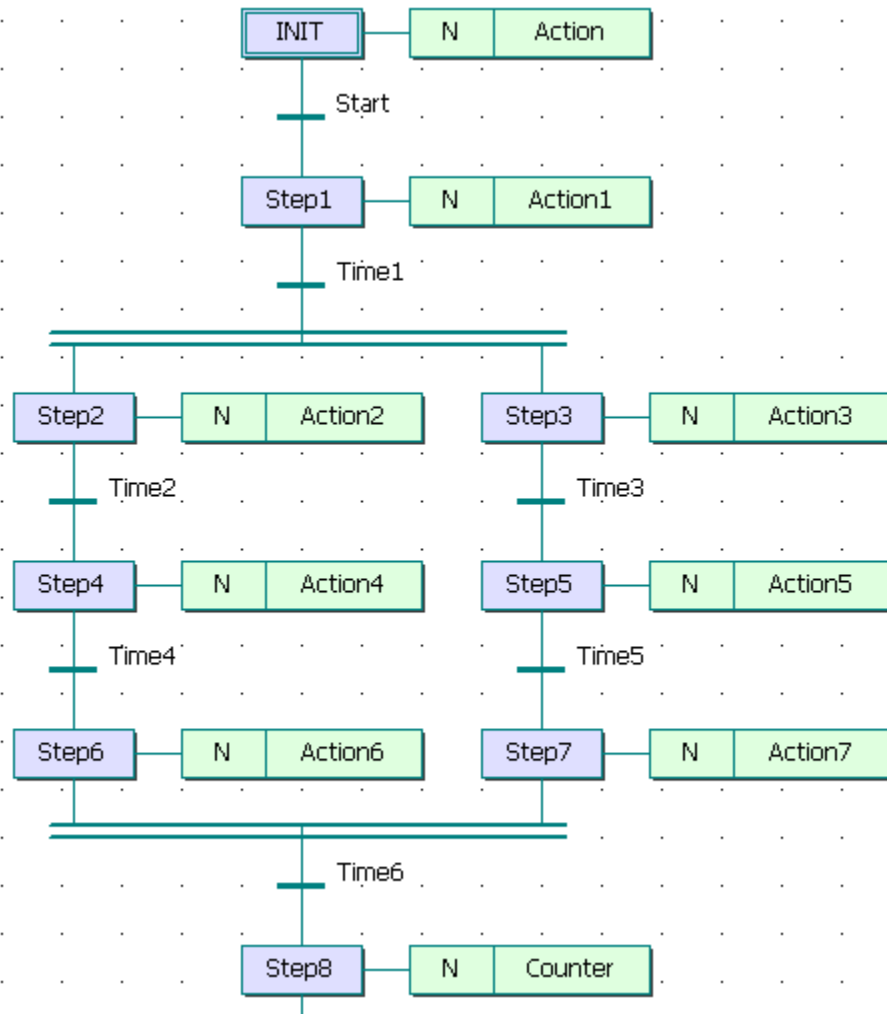
У кожній гілці SFC-схеми послідовно додати по дві пари кроків і переходів для програмування решти фаз роботи світлофорів. Здійснюється це за допомогою кнопки  (Creat step transition sequence) в панелі інструментів після позначення лівою клавiшею миші відповідного кроку.


Для створення лічильника циклів роботи світлофорів позначити лівою клавiшею миші перехід під подвійним сходженням паралельних гілок і додати нову пару крок - перехід, використовуючи кнопку  (Creat step transition sequence) в панелі інструментів.

Відповідно до умов роботи, світлофор повинен зробити 10 циклів і після цього вийти з циклу і вимкнутися, тобто в програмі має бути передбачено альтернативне розгалуження зі стрибками до різних кроків. Тому компонентам SFC-схеми потрібно призначити імена.

Клацнути правою клавiшею миші по кроку S001 і в меню, що з'явиться, лівою клавiшею миші активізувати Object Properties... У вікні Step надрукувати ім'я INIT і натиснути ОК. Таке саме зробити послідовно з дією A001 та переходом T001, привласнюючи їм у діалоговому вікні Action Properties імена, відповідно Action з класифікатором (Qualifier) N та Start у діалоговому вікні Transition, при цьому в обох вікнах альтернатива Detail має бути активізована.

За аналогічною методикою призначити імена решти крокам і переходам, а також діям біля кожного кроку:



Для створення альтернативних гілок розгалуження необхідно після кроку Step8 SFC-схеми створити ще одну пару крок – перехід і позначити лівою клавішею миші останній крок. В панелі інструментів клацнути по кнопці  (Insert Simultaneous/Alternative Divergence) і у діалоговому вікні Divergence, що з'явиться, у полі Branches Count надрукувати 2. Натиснути ОК і у гілках альтернативного розгалуження, що з'являться, створити стрибки на початковий крок INIT і крок Step1. Для цього треба позначити курсором перехід в лівій гілці розгалуження і додати пару крок – перехід. Правою клавішею миші клацнути по кроку і в меню, що з'явиться, лівою клавішею миші вибрати Object Properties... У полі Type активізувати альтернативу Jump і клацнути ОК. У діалоговому вікні Jump/End step insert control, що з'явиться, натиснути кнопку Change і в лівій гілці залишаться тільки перехід і безадресний стрибок у вигляді прямокутника зі стрілкою.

Таким самим чином зробити безадресний стрибок і у правій гілці альтернативного розгалуження.


Щоб призначити адресу стрибку лівою клавішею миші подвійно клацнути по ньому і у діалоговому вікні Step, що з'явиться, вказати ім'я адреси. Для лівого стрибка у якості адреси позначити початковий крок INIT, а для правого стрибка – крок Step1 у натиснути ОК.

Після створення SFC-схеми необхідно запрограмувати дії кроків і умови переходів відповідно до алгоритму роботи світлофора.

Відповідно до синтаксису SFC – мови, перший крок є ініціалізаційним. Тому змінним світлофора red1, yellow1, green1, red2, yellow2, green2,


необхідно присвоїти значення FALSE. Для цього лівою клавшею миші двічі клацнути по дії початкового кроку Action, у діалоговому вікні Insert, що з'явиться, вибрати мову програмування, наприклад ST, і натиснути ОК. Відкриється робоче поле дії Action, у якому всім змінним присвоїти значення FALSE:

```
red1:=False;
yellow1:=False;
green1:=False;
red2:=False;
yellow2:=False;
green2:=False;
```

Щоб забезпечити обнуління лічильника після 10 циклів роботи світлофора перейти на новий рядок і в панелі інструментів активізувати кнопку  (Edit Wizard). У таблиці Group вікна Edit Wizard, що з'явиться на екрані, вибрати меню Keywords. Подвійно клацнути по оператору IF і на робочому полі проекту з'явиться його синтаксис з відповідним коментарем:

```
IF (*EXPRESSION (must return a boolean value)*)
  THEN (*If returned value of EXPRESSION = TRUE*)
  (*STATEMENT*);
END_IF;
```




Щоб доповнити цю заготовку необхідним змістом треба курсор спочатку установити після оператора IF і надрукувати Cycle_Count>10, а потім установити після оператора THEN і надрукувати Cycle_Count:=0.


Для оголошення змінних, у цьому ж полі, курсор по черзі установити на кожне з імен і в панелі інструментів лівою клавшею миші клацнути по кнопці  (Variable). У діалоговому вікні Variables, що з'явиться, у рядку Name відобразиться ім'я вибраної змінної. Натиснути кнопку Apply, відкрити сторінку діалогу Common і у рядку Usage вибрати статус змінної VAR, а у рядку Data Type її тип BOOL. У рядку I/O address надрукувати фізичну адресу вихідного каналу модуля виводу, наприклад %QX0.0 для red1, %QX0.1 для yellow1, %QX0.2 для green1, %QX1.0 для red2, %QX1.1 для yellow2 і %QX1.2 для green2. Натиснути ОК і діалогове вікно закриється, а оголошена змінна автоматично буде занесена у список змінних проекту.




Після оголошення змінних закрити робоче поле дії Action і зберегти її код, натиснувши у діалоговому вікні MULTIPROG, що з'явиться, кнопку ОК.

Відповідно до заданого алгоритму, умовою початку роботи світлофора є дворазове натиснення стартової кнопки. Програмування цієї умови здійснюється LD-мовою, використовуючи прямий зв'язок, при якому перехід не має тіла коду, а безпосередньо сполучається з LD-мережею.

Клацнути правою клавшею миші по переходу Start, у діалоговому вікні Transition, що з'явиться, активізувати Direct Connection, зробити порожнім рядок Name і натиснути ОК. Діалогове вікно закриється, а у робочому полі SFC - схеми після початкового кроку INIT залишиться безіменна смужка переходу з блакитною крапкою. Для створення умови переходу LD-мовою клацнути лівою клавшею миші у зручному місці робочого поля зліва від безіменного переходу.

В панелі інструментів натиснути послідовно кнопки  (Add contact right) і  (Left powerail). В установленому місці робочого поля з'явиться LD- мережа 001 з лівою шиною живлення і контактом. В панелі інструментів натиснути кнопку  (Connect objects) і за допомогою курсору і лівої клавіші миші з'єднати прямий контакт LD- мережі з блакитною крапкою смужки безіменного переходу.

Лівою клавішею миші позначити лінію, що сполучає прямий контакт і перехід. В панелі інструментів клацнути по кнопці  (Edit Wizard) і з таблиці Group (Група), що з'явиться на екрані, вибрати функціональний блок лічильника STU. Двічі клацнути по блоку STU і у діалоговому вікні Variable properties, що з'явиться, змінити типове ім'я STU-1 на Svit_Count. Натиснути послідовно кнопки Apply і OK і на позначеному місці LD- мережі з'явиться лічильник.

Щоб задати параметри лічильнику, клацнути на функціональному блоці STU по синій крапці Reset-входу. В панелі інструментів натиснути кнопку  (Add contact left) і на Reset - вході лічильника з'явиться контакт C002. В панелі інструментів активізувати кнопку  (Connect objects) і за допомогою курсору, а також лівої клавіші миші з'єднати прямий контакт з шиною живлення LD- мережі. В панелі інструментів натиснути кнопку  Mark, щоб деактивізувати кнопку Connect objects.

Для оголошення змінної, яка має увімкнути світлофор, двічі клацнути лівою клавішею миші по контакту C000. У діалоговому вікні Contact/Coil Properties, що з'явиться, замінити типове ім'я C000 на Start. Натиснути кнопку Apply (Застосувати) і на сторінці Common зі списку Usage вибрати статус змінної VAR, зі списку Data Type - тип BOOL, а у рядку I/O address надрукувати фізичну адресу змінної Start - %IX0.0. Після натиснення OK оголошена змінна потрапить у відповідний список, а над контактом C000 з'явиться ім'я Start.

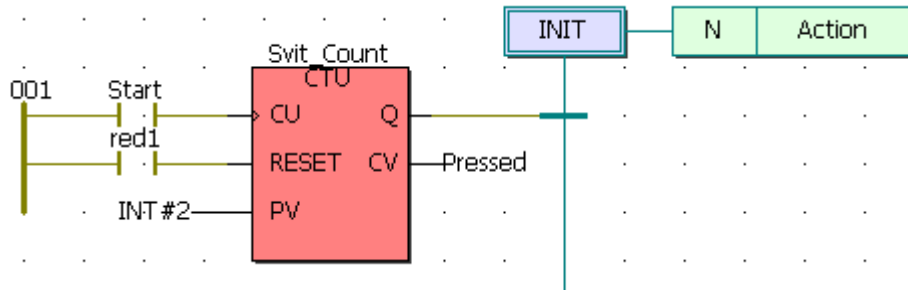
Щоб надати контакту C002 ім'я red1, який має скидати лічильник, необхідно двічі клацнути по контакту. У діалоговому вікні Contact/Coil Properties, що з'явиться, зі списку вже оголошених змінних вибрати red1, клацнути лівою клавішею миші і натиснути OK.

Двічі клацнути по блакитній крапці PV-входу і у діалоговому вікні Variable properties, що з'явиться, у рядку Name надрукувати константу INT#2, щоб світлофор увімкнувся після дворазового натиснення стартової кнопки. Після натиснення OK на PV-вході лічильника з'явиться INT#2.


Двічі клацнути по зеленій крапці CV-виходу лічильника і у діалоговому вікні Variable properties у рядку Name надрукувати Pressed - це внутрішня змінна, яка запам'ятовує поточне значення лічильника.

Натиснути кнопку Apply і на сторінці діалогу Common у рядку Usage вибрати VAR, а в рядку Date type (Тип даних) – INT. Після натиснення OK у робочому полі поряд з CV-виходом з'явиться Pressed.

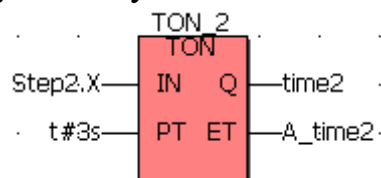
Програмний код умови першого переходу отримує наступний вигляд:



Умовами решти переходів є час роботи той чи іншої фази світлофорів. А дії кроків визначаються станами змінних світлофорів.

Щоб реалізувати програмування таймерів для кожного з переходів необхідно двічі клацнути лівою клавішею по відповідному переходу і у вікні Insert вибрати мову програмування. Після закриття діалогового вікна Insert відкриється робоче поле для створення програмного коду умови переходу. В панелі інструментів натиснути кнопку  і у вікні Edit Wizard знайти функціональний блок таймера TON. Для переносу екземпляра цього блока у робоче поле необхідно двічі клацнути по ньому лівою клавішею миші і у вікні Variables, що відкриється, призначити йому ім'я, або залишити запропоноване по замовчуванню. Після підтвердження імені, екземпляр блока зобразиться у робочому полі і йому потрібно визначити параметри, які забезпечать необхідний час затримки, відповідно до синтаксису вибраної мови.

Наприклад, для програмування умови переходу Time2 FBD-мовою, необхідно перенести екземпляр функціонального блока TON на її робоче поле і двічі клацнути по блакитній крапці входу IN. Відкриється діалогове вікно Variable Properties, в якому необхідно записати ім'я джерела вхідного сигналу (крок Step2) у вигляді Step2.X і натиснути ОК. Після цього двічі клацнути лівою клавішею миші по блакитній крапці входу PT. Відкриється вікно Variable Properties, в якому записати параметр таймера (3с) у вигляді T#3s і натиснути ОК. Двічі клацнути по зеленій крапці виходу Q і у вікні Variable Properties записати ім'я переходу time2, що програмується. Двічі клацнути по зеленій крапці виходу ET і у вікні Variable Properties записати ім'я змінної, яка фіксує поточний час відліку таймера, і натиснути кнопку Apply (Застосувати). Відкриється сторінка діалогового вікна Common, в якій вибрати статус змінної VAR і її тип TIME. Після натиснення кнопки ОК на робочому полі залишиться запрограмована умова переходу Time2 у вигляді:



Умова переходу Time1 ST-мовою має вигляд:

```
TON_1(IN:=Step1.X(* BOOL *),PT:=t#10s(* TIME *));
time1(* BOOL *):=TON_1.Q;
A_time1(* TIME *):=TON_1.ET;
```

Умова переходу Time3 IL-мовою має вигляд :

```
LD Step3.X (* BOOL *)
ST TON_3.IN
```

```

LD t#3s      (* TIME *)
ST  TON_3.PT
CAL  TON_3
LD  TON_3.Q
ST time3     (* BOOL *)
LD  TON_3.ET
ST  A_time3(* TIME *)

```

Умова переходу Time4 ST-мовою має вигляд:

```

TON_4(IN:=Step4.X(* BOOL *),PT:=t#10s(* TIME *));
time4(* BOOL *):=TON_4.Q;
A_time4(* TIME *):=TON_4.ET;

```

Умова переходу Time5 ST-мовою має вигляд:

```

TON_5(IN:=Step5.X(* BOOL *),PT:=t#10s(* TIME *));
time5(* BOOL *):=TON_5.Q;
A_time5(* TIME *):=TON_5.ET;

```

Умова переходу Time6 ST-мовою має вигляд:

```

TON_6(IN:=Step6.X (* BOOL *),PT:=t#3s(* TIME *));
time6(* BOOL *):=TON_6.Q;
A_time6(* TIME *):=TON_6.ET.

```


Останній крок програми Step8 призначений для лічильника циклів роботи світлофора. Дія цього кроку запрограмована ST-мовою з використанням умовного оператора IF, як це здійснювалося при програмуванні початкового кроку INIT:

```

IF time6      (*EXPRESSION (must return a boolean value)*)
THEN Cycle_Count:=Cycle_Count+1    (*If returned value of EXPRESSION = TRUE*)

(*STATEMENT*);
END_IF;


```

В альтернативному розгалуженні умова переходу на крок INIT (при кількості циклів роботи світлофора>10) виконана IL-мовою з використанням оператора GT з бібліотеки Operators вікна Edit Vizard, яке відкривається кнопкою :

```

LD Cycle_Count  (* IN1 as ELEMENTARY *)
GT 10  (* IN2 as ELEMENTARY *)
ST Reset      (* Result as BOOL *).

```

Умова переходу на крок Step1 (при кількості циклів роботи світлофора<10) запрограмована IL-мовою з використанням оператора LE з бібліотеки Operators вікна Edit Vizard, яке відкривається кнопкою :


```

LD Cycle_Count  (* IN1 as ELEMENTARY *)
LE 10  (* IN2 as ELEMENTARY *)
ST Trans      (* Result as BOOL *).


```


Для перевірки програми використовується симулятор. Щоб його активізувати у дереві проекту клацнути правою клавішею миші по теці

Resource і увійти в пункт меню Settings. З'явиться вікно Resource settings for IRC_32, у якому позначити Simulation 1 і закрити діалог, клацнувши ОК.

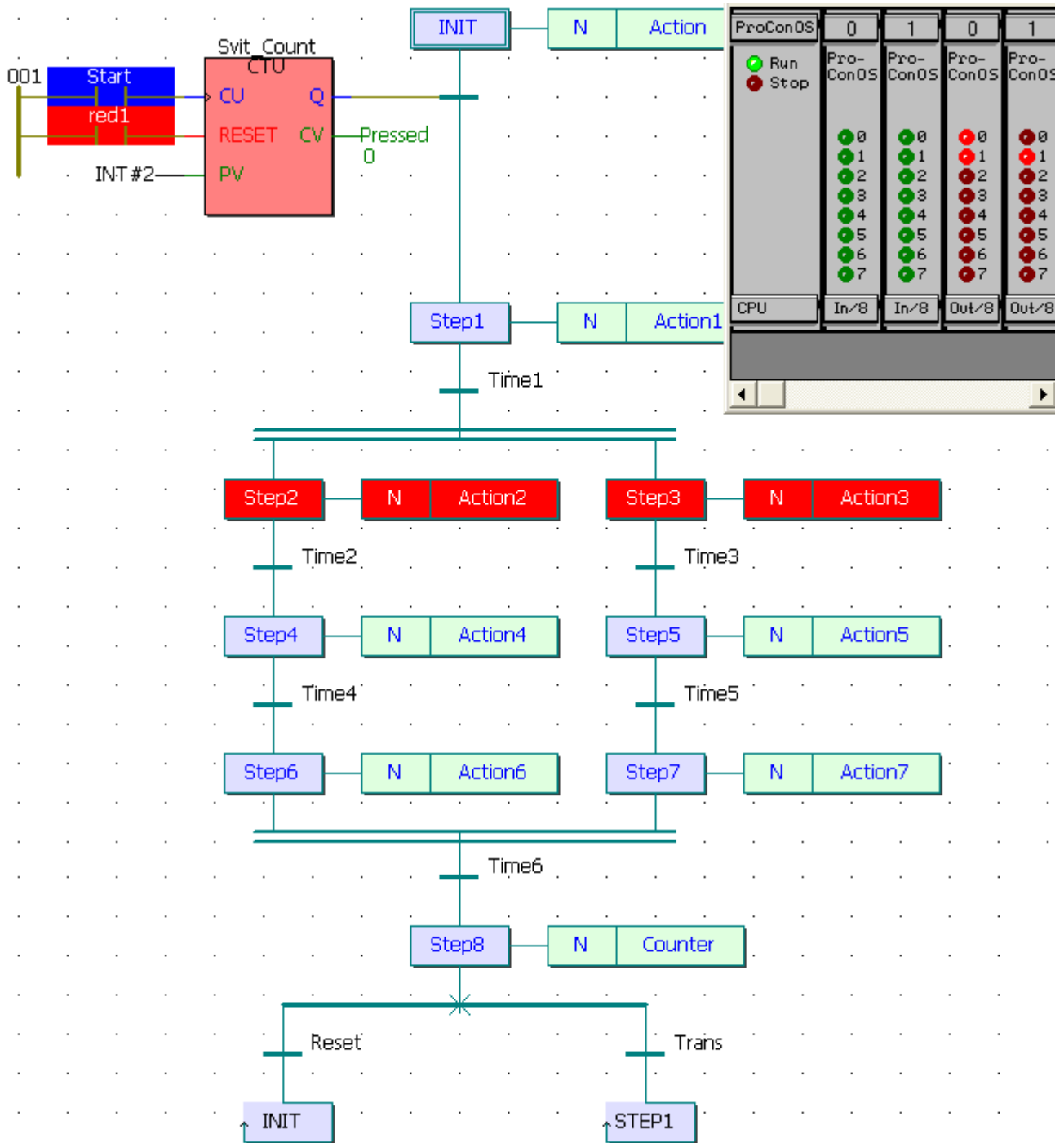
Компіляція проекту відбувається після натиснення кнопки  (Make) в панелі інструментів. У вікні Build унизу екрана відобразяться помилки (Error) і попередження (Warning), якщо будуть знайдені протягом компіляції.



Щоб вивести на екран список помилок або попереджень необхідно клацнути по Error чи Warning. Після подвійного натиснення лівої клавiші миші на вказаній у списку помилці або попередженні, відкриється відповідна сторінка, яку необхідно скоригувати, а потім повторно компілювати проект.

Створений без помилок проект завантажується до симулятора за допомогою кнопки  (Project Control Dialog). У діалоговому вікні Resource, що відкриється, натиснути кнопку Download, а потім продублювати завантаження у частині Project вікна Download. Успішний процес завантаження проекту супроводжується бігом синьої стрічки унизу екрана.

У діалоговому вікні Resource натиснути кнопку Cold для холодного пуску симулятора, а в панелі інструментів клацнути по кнопці  (Debug on/off), щоб перейти в оперативний режим роботи. При цьому усі змінні позначаються відповідними кольорами. Істинний стан – червоним, а хибний – синім.

Лівою клавiшею миші клацнути по Demoio_Driver унизу екрана, для відкриття симулятора I/O. Двічі подвійно клацнути лівою клавiшею миші по нульовому світлодіоду нульового модуля вводу In, оскільки саме цю адресу має змінна Start. Програма почне виконуватися, про що свідчить перехід червоного кольору від початкового кроку INIT до кроку STEP1 і далі, а також робота світлодіодів на симуляторі модулів 0 і 1 виводу Out, відповідно до адреси вихідних змінних:



Щоб завантажити проект до ПЛК необхідно зупинити роботу емулятора кнопкою Stop у вікні Resource і вийти з оперативного режиму за допомогою кнопки  (Debug on/off) в панелі інструментів. Клацнути правою клав'яшею миші по теці Resource, а потім по пункту меню Settings і у вікні Resource settings for IRC_32 позначити COM1. Закрити діалог, клацнувши ОК. В панелі інструментів натиснути кнопку , щоб з'явилося діалогове вікно Resource: ADAM 5510KW, і натиснути послідовно кнопки Stop, Reset і Download. У діалоговому вікні Download, що з'явиться, натиснути кнопку Download у частині Bootproject і проект завантажиться у пам'ять ПЛК, про що свідчить біг синьої стрічки внизу екрана.

У вікні Resource:(ADAM5510KW) знову натиснути кнопку Download, а потім натиснути кнопку Activate у частині Bootproject вікна Download. Після

цього програму можна запустити, використовуючи Cold start у вікні Resource[1,27].