

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

МЕТОДИЧНІ ВКАЗІВКИ

для проведення лабораторних занять з курсу

«Комп'ютерні мережі»

(у двох частинах)

для студентів напрямку підготовки 050202
«Автоматизація та комп'ютерно-інтегровані технології»
денної та заочної форм навчання

Частина 1

Затверджено
редакційно-видавничою
радою університету,
протокол № 1 від 20.03.2015 р.

Харків
НТУ «ХПІ»
2015

Методичні вказівки для проведення лабораторних занять з курсу «Комп'ютерні мережі» (у двох частинах) для студентів напряму підготовки 050202 «Автоматизація та комп'ютерно-інтегровані технології» денної та заочної форм навчання. Частина 1 / Уклад. Подустов М.О., Лисаченко І.Г., Лобойко В.О., Шутинський О. Г. – Х. : НТУ «ХП», 2015. – 48 с.

Укладачі: М.О. Подустов
І. Г. Лисаченко
В.О. Лобойко
О. Г. Шутинський

Рецензент: доц. І. Л. Красніков

Кафедра автоматизації хіміко-технологічних систем
та екологічного моніторингу

ВСТУП

Ці методичні вказівки розроблені для проведення лабораторного практикуму зі студентами денної та заочної форм навчання за напрямом «Автоматизація та комп'ютерно-інтегровані технології» з дисципліни «Комп'ютерні мережі».

Методичні вказівки вміщують необхідні теоретичні відомості для ознайомлення зі стандартами на послідовні інтерфейси *RS-232* та *RS-485* [1], які є апаратною основою побудови комп'ютеризованих систем управління технологічними процесами. Також у лабораторному практикумі розглянуто основні принципи використання *AT*-команд [2] для управління модемами (*GSM*-модемами), які все частіше використовують для дистанційного управління автономними технологічними об'єктами. Виконання лабораторного практикуму дозволяє студентам більш ефективно засвоїти в подальшому теорію та практику застосування промислових мереж.

Як апаратне забезпечення в лабораторному практикумі використані звичайні персональні комп'ютери (ПК), програмовані логічні контролери (ПЛК) виробництва компанії «*ВО ОВЕН*» (Україна) 100-ї серії [3, 4], *GSM*-модеми *TC-35 UART/RS-232* [5] виробництва компанії *Siemens AG* (Німеччина) та обладнання для з'єднання пристроїв: кабелі та конвертери інтерфейсів.

Програмне забезпечення для виконання лабораторного практикуму потрібно для прослуховування *SOM*-портів ПК та для створення проєктів з управляючими програмами. Це відповідно утиліта *HyperTerminal*, яка входить до складу операційних систем *Windows* від *Microsoft Corp.*, прикладна програма *COM Port Toolkit* для прослуховування інтерфейсів ПК та середовище *CoDeSys V2.3* [6, 7] для програмування ПЛК.

Лабораторна робота 1
ЗАСТОСУВАННЯ ІНТЕРФЕЙСУ RS-232
ДЛЯ ОРГАНІЗАЦІЇ ЗВ'ЯЗКУ МІЖ ДВОМА DTE-ПРИСТРОЯМИ

1.1. Мета та час проведення лабораторної роботи

На цю лабораторну роботу відповідно до робочої програми вивчення дисципліни «Комп'ютерні мережі» відводиться 2 акад. год. аудиторного та 2 акад. год. позааудиторного часу (СПС).

Цілі лабораторної роботи:

- закріпити на практиці отримані теоретичні знання про стандарт на асинхронний послідовний інтерфейс RS-232C;
- навчитися працювати з COM-портом ПК безпосередньо, за допомогою утиліти *HyperTerminal*, яка входить до складу операційних систем *Windows* від *Microsoft Corp.*;
- ознайомитися з прикладною програмою *COM Port Toolkit* для обміну даними через COM-порт ПК;
- ознайомитися з таблицею ASCII-кодів і принципами формування символічних повідомлень в ASCII- та HEX-кодах.

1.2. Апаратне забезпечення лабораторної роботи

Лабораторна робота виконується на двох ПК, які з'єднані «нуль-модемним» кабелем. Можливе застосування як повного, так і мінімального «нуль-модемного» кабелю.

УВАГА!

COM-порти ПК електрично не ізолювані.

**Тому для безпечного з'єднання ПК необхідно ОБОВ'ЯЗКОВО
підключати «нуль-модемний» кабель при вимкненому
живленні ПК, з дозволу і під спостереженням викладача**

Схему з'єднання комп'ютерів наведено на рис. 1.1, а схему «нуль-модемного» кабелю – на рис. 1.2.



Рисунок 1.1 – Схема лабораторного комплексу (схема з'єднання двох ПК без модема)

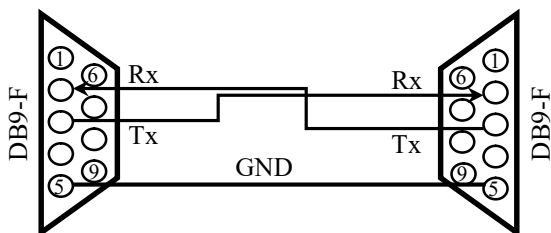


Рисунок 1.2 – Схема «нульмодемного» кабелю

У табл. 1.1 наведено варіанти налаштувань з'єднань, попарно з'єднаних сполучених ПК, та формат кадрів, що передаються в повідомленні. У вікні настроювання інтерфейсу *COM*-портів для всіх пар інтерфейсів *RS-232* встановіть однакову швидкість та формат кадру, що передається.

Таблиця 1.1 – Параметри настроювання інтерфейсу *COM*-портів ПК

Параметри інтерфейсу	Робочі місця (ПК)				
	<i>S1-S2</i>	<i>S3-S4</i>	<i>S5-S6</i>	<i>S7-S8</i>	<i>S9-S10</i>
Швидкість, біт/с	9600	9600	9600	9600	9600
Кількість біт у кадрі	8	8	8	7	7
Паритет	немає	пар.	непар.	пар.	непар.
Кількість стоп-бітів	1	1	1	2	2

У табл. 1.2 вказано зміст повідомлень в *ASCII*- та в *HEX*-кодi, якими будуть обмінюватись ПК.

Таблиця 1.2 – Зміст повідомлень, що надсилають ПК

Формат	Робочі місця (ПК)				
	$S1 \leftrightarrow S2$	$S3 \leftrightarrow S4$	$S5 \leftrightarrow S6$	$S7 \leftrightarrow S8$	$S9 \leftrightarrow S10$
<i>ASCII</i>	()	*	+	,
<i>HEX</i>	28	29	2A	2B	2C

1.3. Програмне забезпечення лабораторної роботи

Для реалізації обміну даними в лабораторній роботі застосовується програма *HyperTerminal*, яка вбудована у стандартний набір комунікаційних утиліт ОС *Windows XP/2000/ME*. Також в лабораторній роботі для сканування *COM*-портів і відправлення повідомлень використовується прикладна програма *COM Port Toolkit 4.0*, яка є безкоштовною і доступна для скачування з сайту [HTTP://www.compt.ru](http://www.compt.ru).

1.4. Короткі теоретичні відомості

Стандарт інтерфейсу *RS-232C* вперше опублікований асоціацією *EIA* [1] в 1969 році як варіант "C" рекомендованого стандарту (*RS – Recommended Standard*) за номером «232». Інтерфейс призначений для підключення апаратури, яка передає (або приймає) дані апаратури передавання даних – АПД (*DTE - Data Terminal Equipment*) до (від) апаратури каналів даних – АКД (*DCE - Data Communication Equipment*). Розділення апаратури на *DTE* та *DCE* дозволило стандартизувати відповідні пристрої різних виробників для організації обміну даними в інформаційних мережах.

Найпростіший приклад – це пряме з'єднання двох *DTE*, тобто двох ПК, яке зображене на рис.1.1. Приклад системи, коли функцію *DTE* виконіє комп'ютер, в якому *RS-232* реалізований як *COM*-порт, а функцію *DCE* – модем, що показаний на рис.1.3.

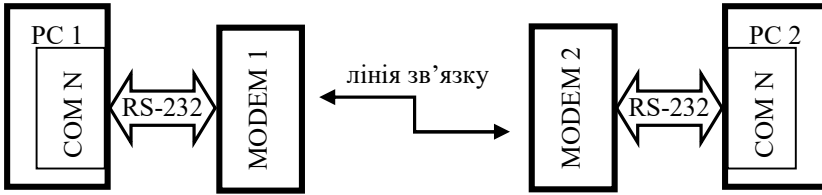


Рисунок 1.3 – Схема з'єднання двох ПК з використанням модемів

Інтерфейс *RS-232* належить до послідовних каналів зв'язку. Це означає, що дані (інформація) передаються послідовно, біт за бітом по одному дроту (на відміну від паралельного інтерфейсу, в якому, наприклад, кожен біт байта передається по окремому дроту, тобто байт передається по восьми дротах). Формат послілки – один байт даних і декілька службових бітів, деякі з яких можуть бути відсутніми. В цілому стандарт визначає електричні та механічні параметри інтерфейсу. Для передавання одного біта використовується несиметричний метод, тобто інформаційний сигнал передається за допомогою потенціалу в інформаційній лінії відносно загального дроту. При цьому обмін інформацією між ПК та периферійним пристроєм (модем або інший ПК) по інтерфейсу *RS-232* є двостороннім, тобто дані можуть передаватися від ПК в модем і прийматися зворотно одночасно по різних лініях. Такий режим обміну називається *дуплексним*.

Для передавання окремих бітів у стандарті *RS-232* передбачені електричні параметри передавачів та приймачів, функцію яких виконує стандартна мікросхема. Це, власне, приймач-передавач інтерфейсу, який має скорочену назву *UART* (*Universal Asynchronous Receiver-Transmitter* – універсальний асинхронний приймач та передавач). Мікросхема *UART* є частиною інтерфейсу між шиною мікропроцесора (*TTL*-логіка, 3.3 або 5 В) та трансивером каналу зв'язку. Тобто вихід *UART* безпосередньо не підключається до каналу зв'язку, а видає біти на рівні *TTL* на вхід трансивера, який, в свою чергу формує сигнали для передавання у лінію зв'язку.

У табл. 1.3 наведено електричні параметри сигналів, що формуються в лінії зв'язку передавачем та отримуються на вході приймачем. Значення параметрів управляючих ліній співпадають з параметрами інформаційних ліній, але в інверсному розумінні. Це означає, що управляючий сигнал, який має статус «ТАК», передається позитивним потенціалом. Можливе пряме використання *TTL*-логіки в інтерфейсі *RS-232*, але це не завжди зручно.

Таблиця 1.3 – Електричні параметри стандарту *RS-232*

Параметр інтерфейсу	Значення	Одиниця виміру
Напруга на виході передавача: логічна одиниця логічний нуль	-5...-25 +5...+25	В
Напруга на вході приймача	$\pm 3 \dots 25$	В
Невизначеність логічного рівня	± 3	В

У стаціонарному ПК на материнській платі розміщений спеціальний з'єднувач типу *DB-9(M)*, так званий комунікаційний порт (*COM*-порт). До з'єднувача за допомогою кабелю підключається інший ПК (ПЛК) або модем. Для першого випадку (див. рис. 1.1) використовують так званий «нульмодемний» кабель, що забезпечує зв'язок пристроїв типу *DTE-DTE*. Тому, для випадку прямого з'єднання двох ПК «нульмодемний» кабель з обох боків повинен мати з'єднувач типу *DB-9(F)*.

Для зв'язку ПК та модема, тобто *DTE-DCE*, застосовують так званий «прямий» кабель (рис.1.3). Так, пристрої *DTE* мають з'єднувачі типу *DB-9(M)*, а пристрої *DCE* – з'єднувачі типу *DB-9(F)*. Це означає, що кабель з'єднання ПК та модема має з боку ПК з'єднувачі типу *DB-9(F)*, а з боку модема – з'єднувачі типу *DB-9(M)*.

Далі модем підключається до лінії фіксованого телефонного зв'язку. На цей час для диспетчеризації та моніторингу автономних пунктів все більше застосовують безпроводні модеми, тобто *GSM*-модеми або радіомодеми. В останньому випадку використовується звичайний так званий «прозорий» радіоканал.

До складу кабелю входять інформаційні та управляючі лінії (дроти), які називають лініями інтерфейсу. Управляючі лінії потрібні для управління потоком даних по інформаційних лініях. Склад та призначення цих ліній залежить від умов з'єднання. Також є певні вимоги до обладнання та типів з'єднувачів. Взагалі, *DTE* відправляє дані на контакт «2» і приймає на контакт «3», а *DCE* відправляє дані на контакт «3» і приймає на контакт «2». У табл. 1.4 наведено склад і призначення ліній в інтерфейсі *RS-232* для з'єднувача типу *DB-9*. На рис. 1.4 показано з'єднувач *DB-9* з позначенням сигнальних та управляючих ліній.

Таблиця 1.4 – Позначення входів та виходів з'єднувача *DB-9* (з боку термінального пристрою)

Назва сигналу	Позначення сигналу	Призначення сигналу	Напрямок сигналу	Номер лінії
<i>Signal Common</i>	<i>GND</i>	Загальний сигнальний дріт	–	5
<i>Transmitted Data</i>	<i>TxD</i>	Передача даних. Вихід передавача	<i>DTE</i> → <i>DCE</i>	3
<i>Received Data</i>	<i>RxD</i>	Прийом даних. Вхід приймача	<i>DTE</i> ← <i>DCE</i>	2
<i>Request to send</i>	<i>RTS</i>	Запит терміналу дозволу на передачу даних	<i>DTE</i> → <i>DCE</i>	7
<i>Clear to send</i>	<i>CTS</i>	Модем вільний для передачі даних	<i>DTE</i> ← <i>DCE</i>	8
<i>DCE Ready</i>	<i>DSR</i>	Вказує на готовність модема до роботи	<i>DTE</i> ← <i>DCE</i>	6
<i>DTE Ready</i>	<i>DTR</i>	Готовність терміналу до обміну даними	<i>DTE</i> → <i>DCE</i>	4
<i>Received Line Signal Detector</i>	<i>DCD</i>	Показує наявність сигналу несучої на вході модема	<i>DTE</i> ← <i>DCE</i>	1
<i>Ring indicator</i>	<i>RI</i>	Запит на установку з'єднання від модема (прийом модемом ви-клик)	<i>DTE</i> ← <i>DCE</i>	9

Для управління потоком даних в *RS-232* передбачено два способи: апаратний та програмний. У цій роботі використаний мінімальний «нульмодемний» кабель, тобто немає управляючих ліній, тому обмін даними здійснюється без управління потоком. Для реалізації управління потоком даних апаратним способом можливі два варіанти залежно від обраних управляючих ліній. Частіше використовують пару сигналів *RTS–CTS*, що означає управління обміном з боку терміналу. Рідше пару *DSR–DTR*, де обмін ведеться за ініціативою модема. В обох випадках ведений пристрій відповідає готовністю шляхом включення відповідної лінії. Програмне управління потоком даних здійснюється за допомогою використання додаткових спеціальних *ASCII*-символів, які передаються по інформаційних лініях та мають назву «*XON*» та «*XOFF*».

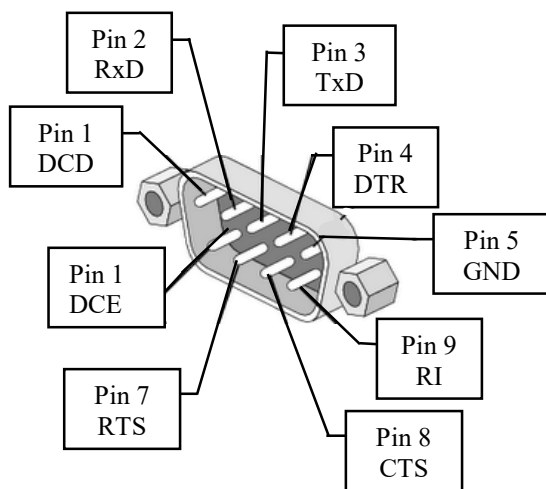
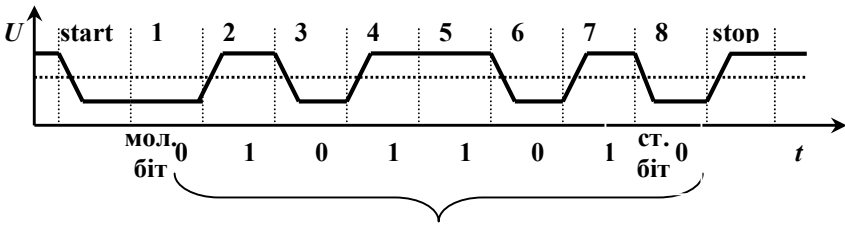


Рисунок 1.4 – Позначення сигнальних та управляючих ліній з'єднувача DB-9(M) (з боку термінального пристрою)

На рис. 1.5 показана діаграма передавання одного символу *ASCII*-коду (наприклад, символ «*Z*»= $5A_{\text{hex}}=0101_1010_2$), який передається по інформаційній лінії даних *TXD* інтерфейсу *RS-232*. Параметри кадру

такі: 8 біт даних, без перевірки на парність, один стоп-біт. Як впливає з рисунка, передача починається з так званого «старт-біта», потім йдуть біти даних, починаючи з молодшого (їх може бути від п'яти до восьми), далі слідує біт паритету або парності (який може бути відсутнім) і потім слідує стоп-біти (їх може бути або один, або два залежно від наявності біта парності). На практиці використовують вісім бітів даних, при цьому біт паритету (на рисунку не показаний), як правило, не використовується, а при максимально високій швидкості передачі бажано передавати два стоп-біти.



ASCII-символ «Z» ($5A_{hex}$)

Рисунок 1.5 – Діаграма передавання одного символу

При формуванні кадру можуть додаватися різні комбінації кількості бітів паритету та стопових бітів. При прийомі кадрів службові біти автоматично видаляються. Зазвичай стартовий та стоповий біти обрамляють один байт інформації (8 біт), проте в полі даних може бути 5, 6, 7 або 9 бітів. Для відокремлення кадру від наступного використовують 1, 1.5 або 2 стоп-біти, що означає інтервал часу, який дорівнює величині, яка зворотна швидкості передавання даних. Два стопових біти використовують для зменшення вірогідності розузгодження приймача та передавача при щільному трафіку, тобто швидкості передавання даних. Приймач ігнорує другий стоповий біт, сприймаючи його як коротку паузу в лінії.

Біт парності призначений для контролю за правильністю передачі даних і вибирається з ряду: парний (*even*), непарний (*odd*), відсутній

(*none*). При виборі останнього на фізичному рівні не проводиться контроль помилок. Якщо використовується *парний* (*even*) біт паритету, то при передачі підраховується кількість одиничних бітів даних, і якщо їхня кількість непарна, то добавляється біт паритету рівним логічній "1", в протилежному випадку добавляється логічний "0". При *непарному* (*odd*) паритеті – навпаки, передавач буде добавляти до бітів даних логічну "1" або "0" так, щоб сума одиничних бітів у бітах даних разом з паритетним була непарною. Приймач перевіряє суму прийнятих одиничних бітів даних та паритету і якщо вона не співпадає з типом наперед визначеного біта паритету, сигналізує про це верхнім рівням, які вирішують можливість повторного запиту.

Оскільки службові біти займають частину бітового потоку, то результуюча пропускна здатність не дорівнює швидкості з'єднання. Наприклад, для 8-бітових посилок формату «8-N-1» синхронізуючі біти займають 20 % потоку, що для фізичної швидкості 115 200 бод дає бітову швидкість даних 92 160 бим/с або 11 520 байм/с.

Насамкінець відзначимо, що інтерфейс RS-232 належить до з'єднань типу «точка-точка» і забезпечує зв'язок на відстанях не більш ніж 15 м на невеликих швидкостях і 1,5 м на швидкості 115 200 бод. Такі параметри зумовлені електричними параметрами стандарту та якістю провідників у кабелі.

1.5. Послідовність виконання лабораторної роботи

1.5.1. Обмін даними через утиліту HyperTerminal

Після підключення кабелю та вмикання ПК необхідно запустити за допомогою кнопки «Пуск» на робочому столі і послідовного вибору таких пунктів меню – «Все программы», «Стандартные» і «Связь» – програму *HyperTerminal*. Далі, за допомогою меню програми, потрібно створити нове підключення (на рис.1.6 – це «COM10») по вказанню у таблиці COM-портом (наприклад, це порт №10) з указаними параметрами інтерфейсу. Далі, після здійснення підключення позмінно передавайте повідомлення від одного ПК іншому. При цьому, за умовчанням, на ПК-

відправнику в рядку введення повідомлення не відобразатиметься. Щоб бачити результат введення символів, необхідно виставити позначку в основних властивостях з'єднання, через вкладку *Setting* в налаштуваннях символічного обміну, як це показано на рис.1.6.

Окрім символічних повідомлень, через *HyperTerminal* можна відправляти та отримувати файли. Для цього в меню швидкого доступу *Transfer* в *Hyperterminal* є команди *Send File* і *Receive File*.

Створіть в текстовому редакторі «Блокнот» файл з коротким повідомленням власного змісту. Далі відправте свій файл на ПК-абонент, тобто спочатку від одного ПК на інший, потім інший файл – назад.

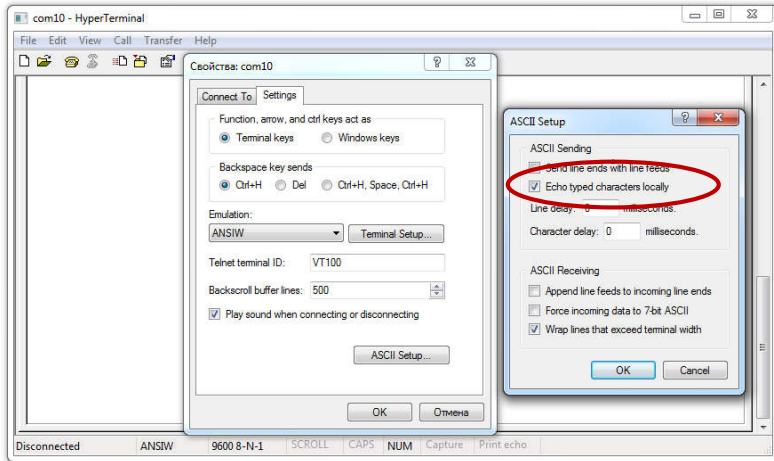


Рисунок1.6 – Вікно налаштування програми *HyperTerminal*

УВАГА!

З одним і тим же COM-портом не можуть одночасно працювати дві і більш програми (за рідкісним виключенням – ComRead). Тому перед запуском програми COM Port Toolkit 4.0 обов'язково закрийте всі програми, які мають доступ до цього COM-порту.

Отже, після виконання завдання, відключіться від COM-порту та закрийте програму *HyperTerminal*.

1.5.2. Обмін даними за допомогою програми COM Port Toolkit

Використовуючи меню швидкого запуску, знайдіть та запустіть програму на обох ПК, які з'єднані «нульмодемним» кабелем. Далі створіть та побудуйте з'єднання (профіль) з COM-портом. Для свого профілю відповідно до таблиці створіть повідомлення у вбудованому в програму блокуванні. По черзі пересилайте повідомлення від одного ПК до другого та спостерігайте результати пересилання в інформаційному полі. Це будуть повідомлення в HEX- або ASCII- форматі та час приходу повідомлення.

Для отримання порівняльних оцінок часу передачі повідомлень в різних форматах (HEX та ASCII) необхідно спочатку відправити п'ять повідомлень від одного ПК до другого ПК. При цьому потрібно зафіксувати час передачі повідомлення в *мілісекундах*. Потім необхідно відправити інше повідомлення назад, тобто від другого ПК до першого і теж зафіксувати час передачі. При однакових налаштуваннях COM-портів та однаковій кількості символів середній час прямої і зворотної передачі повинен збігатися.

Розглянемо передачу наступного повідомлення, наприклад передачу ASCII-символу «=». Якщо цей символ передавати в режимі HEX, то програма вкаже на неможливість передачі подібного символу, оскільки у HEX-форматі допустимими є лише 16 символів: цифри від 0 до 9 та букви латинського алфавіту A, B, C, D, E і F. Проте символ «=» можна передати у HEX-форматі, використовуючи його ASCII-код. У цьому випадку це буде посилка двох HEX-символів «3 D». Проте, якщо передати символи «3 D» в рядковому форматі, то на приймальній стороні буде отримана комбінація «33 44». Крім того, відзначимо, що передача HEX-символів у HEX-форматі можлива тільки, якщо передається два символи. Розглянутий вище приклад передачі символу «=» вказує на необхідність визначення умов передачі даних, тобто режиму та формату передачі даних. Також необхідно враховувати, що в ASCII-режимі використовуються службові символи «CR» (повернення каретки) та «LF» (переклад рядка). У свою чергу, цим символам відповідають такі HEX-

коди: «0D» і «0A», що відповідає двійковому коду: «0001101» і «0001010» або десятковому коду – «13» і «10».

Отже, після отримання середніх оцінок часу передачі побудуйте діаграму стану лінії зв'язку при відправці повідомлення в масштабі 10 мм = 1/9600.

Далі, на приймальній стороні, змініть швидкість передачі даних на 19 200 біт/с та повторіть дії. Спостерігайте час приходу повідомлення і прийняті символи. Порівняйте отримані результати.

Після проведення роботи зробіть висновки про важливість налаштування швидкості передавання даних з обох боків лінії зв'язку.

1.6. Оформлення результатів проведення заняття та контрольні питання

Складіть звіт про виконану лабораторну роботу, де наведіть схему з'єднань, умови обміну даними та діаграму повідомлень, що відправляється. Також вкажіть, яке повідомлення прийняте на однаковій швидкості та подвійній швидкості, для текстового та символьного режимів.

Після проведення заняття дайте логічне пояснення отриманим результатам та інтерпретуйте побудовані діаграми.

Крім того, студент повинен відповісти на контрольні питання, приклади яких наведено нижче:

- 1) Яке призначення всіх контактів 9-пінового з'єднувача для реалізації інтерфейсу *RS-232C*?
- 2) Які лінії та для чого використовують в «мінімальному» «нульмодемном» кабелі?
- 3) Яким чином та за допомогою яких ліній реалізується апаратне управління передачею даних по інтерфейсу *RS-232C*?
- 4) Поясніть, чому при збільшенні швидкості приймання даних спотворюється посилання при однаковому форматі кадрів?
- 5) У чому полягає принцип програмного управління потоком даних по інтерфейсу *RS-232C*?
- 6) Які функції виконує мікросхема *UART* в інтерфейсі *RS-232C*?

Лабораторна робота 2
ЗАСТОСУВАННЯ ПЕРЕТВОРЮВАЧА ІНТЕРФЕЙСІВ
RS-232/RS-485 ДЛЯ ЗВ'ЯЗКУ ДВОХ DTE-ПРИСТРОІВ

2.1. Мета та час проведення лабораторної роботи

На цю лабораторну роботу відповідно до робочої програми вивчення дисципліни «Комп'ютерні мережі» відводиться 2 акад. год. аудиторного та 2 акад. год. позааудиторного часу (СРС).

Цілі лабораторної роботи:

- закріпити на практиці отримані теоретичні знання про стандарти на послідовні інтерфейси *RS-232* та *RS-485*;
- ознайомитися з реалізацією прямого управління інтерфейсом *RS-485* з боку ПЛК ОВЕН150 для передачі повідомлень на прикладі використання бібліотеки *SysLibCom.lib*, яка входить до комплексу *CoDeSys*;
- закріпити практичні навички роботи в утиліті *HyperTerminal* та прикладній програмі *COM Port Toolkit* для обміну даними з ПЛК через *COM*-порт ПК.

2.2. Апаратне забезпечення лабораторної роботи

Лабораторна робота виконується на робочих місцях з ПК та підключеними до них ПЛК, які встановлені на стендах. Оскільки ПК не мають вбудованого інтерфейсу *RS-485*, то для зв'язку з ПЛК застосовуються перетворювачі інтерфейсів *RS-232/RS-485* або *USB/RS-485*. В останньому випадку в ОС *Windows* за допомогою драйвера емулюється віртуальний *COM*-порт, з яким можуть без обмежень працювати утиліта *HyperTerminal* або програма *COM Port Toolkit*.

УВАГА!

Номер та параметри віртуального *COM*-порту визначаються за допомогою команди «Управління», що викликається через контекстне меню із списку з ім'ям «Комп'ютер»

Схеми з'єднання ПЛК та ПК за допомогою автоматичних перетворювачів інтерфейсів *USB/RS-485* та *RS-232/RS-485* наведено відповідно на рис. 2.1 та 2.2. Ці перетворювачі насамперед здійснюють функцію узгодження електричних параметрів вказаних інтерфейсів. Крім того, вони забезпечують узгодження роботи передавачів та приймачів під час обміну даними.

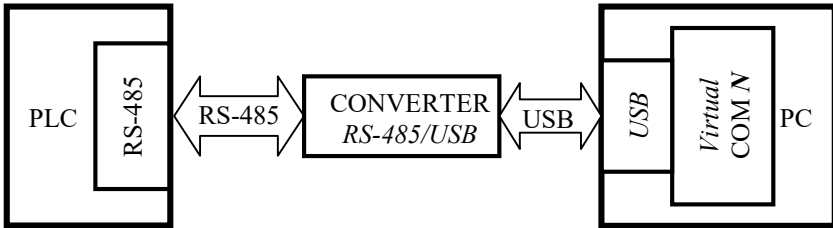


Рисунок 2.1 – Схема з'єднання ПЛК та ПК за допомогою перетворювача інтерфейсів *USB/RS-485*

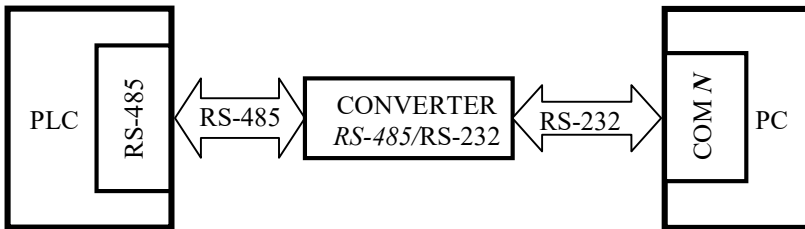


Рисунок 2.2 – Схема з'єднання ПЛК та ПК за допомогою перетворювача інтерфейсів *RS-232/RS-485*

У табл. 2.1 наведено варіанти налаштувань з'єднань ПЛК і ПК та формат кадрів, що передаються. У вікні налаштування інтерфейсу *COM*-портів для всіх пар інтерфейсів *RS-232* (ПК) та *RS-485* (ПЛК) встановить однакову швидкість та формат кадру.

У табл. 2.2 запропоновано зміст повідомлень для відправлення від ПЛК до ПК в *ASCII*-кодi.

Таблиця 2.1 – Параметри налаштування інтерфейсу COM-портів у ПК та ПЛК

Параметри інтерфейсу	Робочі місця (ПК)				
	<i>S1, S2</i>	<i>S3,S4</i>	<i>S5,S6</i>	<i>S7,S8</i>	<i>S9,S10</i>
Швидкість, біт/с	9600	9600	9600	9600	9600
Кількість біт у кадрі	8	8	8	7	7
Паритет	немає	пар.	непар.	пар.	непар.
Кількість стоп-бітів	1	1	1	2	2

Таблиця 2.2 – Зміст повідомлення для відправлення

<i>ПК</i>	<i>Текст сообщения</i>	<i>ПК</i>	<i>Текст сообщения</i>
<i>S1</i>	The receive buffer is full	<i>S6</i>	What is pressure
<i>S2</i>	Start the standby pump	<i>S7</i>	What is the flow
<i>S3</i>	Communication parameters	<i>S8</i>	What is the temperature
<i>S4</i>	Set output bit	<i>S9</i>	What type of connection
<i>S5</i>	What is the time	<i>S10</i>	What type of connection

2.3. Програмне забезпечення лабораторної роботи

Для управління послідовним портом *RS-485* і відправлення повідомлень з боку ПЛК150 необхідно використовувати проект з призначеною для користувача програмою, яка створена в середовищі *CoDeSys* [3, 4, 6, 7] на мові *ST*. У програмі використовується системна бібліотека *SysLibCom.lib*, яка входить в *CoDeSys*. У лабораторній роботі пропонується використати готовий проект, який необхідно скоректувати відповідно до завдання. Проект є навчальним і реалізує лише раз в одну секунду відправлення повідомлень в режимі передачі *ASCII*-символів.

Для прийому та аналізу повідомлень від ПЛК в ПК застосовуються утиліта *HyperTerminal* та програма *COM Port Toolkit*, які вже відомі та розглянуті раніше на лабораторному занятті № 1.

2.4. Короткі теоретичні відомості

Іноді використання інтерфейсу *RS-232* не відповідає поставленим вимогам щодо обміну даними. Це стосується в першу чергу обмеження на дальність дії лінії зв'язку (до 15 м) та на модель обміну даними («точка-точка»). Такі показники не завжди відповідають використанню в умовах промислового виробництва. Ці недоліки інтерфейсу можуть бути усунені, якщо використати інший інтерфейс зв'язку, а саме *RS-485*.

Так, інтерфейс *RS-485*, описаний у стандарті *EIA/TIA-485* [8], є найбільш поширеним в промисловій автоматизації. Пов'язано це з тим, що за всіма основними параметрами цей інтерфейс є якнайкращим зі всіх можливих. На цьому інтерфейсі побудовано більшість відомих промислових протоколів (*ModBus*) та мереж (*Profibus*). Основними його перевагами є:

- обмін даними здійснюється лише по одній крученій парі;
- можливість організації мережі (топологія *шина*);
- багатоточковий режим зв'язку;
- велика довжина лінії зв'язку;
- можливість живлення пристроїв;
- достатньо висока швидкість передачі даних без зниження їх якості та надійності.

В основі побудови інтерфейсу *RS-485* лежить диференціальний спосіб передачі сигналу, коли напруга, відповідна рівню логічної одиниці або нуля, відлічується не від «землі», а вимірюється як різниця потенціалів між двома передавальними лініями: *Data+* та *Data-*. При цьому напруга кожної лінії відносно «землі» може бути довільною, але не повинна виходити за діапазон – 7...12 В.

Приймачі сигналу є диференціальними, тобто сприймають лише різницю між напругою на лінії *Data+* та *Data-*. При різниці напруги більше 200 мВ, до +12 В вважається, що на лінії встановлено значення логічної одиниці, при напрузі менше –200 мВ, до – 7 В – логічного нуля. Для реалізації багатоточкового режиму обміну в інтерфейсі *RS-485* використаний варіант побудови мережі, коли приймачі по відношенню до передавача є так званим вхідним імпедансом (опором), причому опір приймачів не обмежується, але це збільшує навантаження на передавач та робить систему чутливою до перешкод.

Інтерфейс *RS-485* працює таким чином. Перш за все на всіх трансиверах, крім інформаційних сигналів *RD* та *TD*, з'явився додатковий сигнал управління передачі/прийому. Тепер, при закінченні передачі пристрій може відключити свій передавач (перевести його високоімпедансний стан), і надати можливість іншим трансмітерам підключатися для передачі. Іншими словами, сигнал дозволу передачі переводить трансмітер з активного стану в пасивний, тобто передавач може генерувати логічну „1”, логічний „0” або знаходитись в пасивному стані. В який час та в якому порядку передавачі вузлів будуть отримувати доступ до шини у стандарті не зазначено, це питання повинне вирішуватись на канальному рівні (організація доступу до шини). Реалізація інтерфейсу повинна забезпечити працездатність обладнання при короткочасних колізіях (коли два передавачі займають одночасно шину), що, до речі, теж не описано в стандарті.

У стандарті на інтерфейс *RS-485* не описано вимог до середовища передавання даних. Але найчастіше використовується двопровідна схема передачі даних на фізичному рівні. Це, наприклад, може бути екранована кручена пара. Передавач та приймач кожного вузла підключається до єдиної шини, тобто контакт «А» кожного передавача та контакт «А'» кожного приймача підключаються до загальної лінії «А», так само і контакти «В» та «В'» підключаються до лінії «В». Таким чином, коли передавач одного вузла передає послідовність бітів, всі приймачі інших

вузлів їх приймають. Так в інтерфейсі *RS-485* реалізовано півдуплексний режим обміну, тобто система дозволяє в один момент часу вузлу передавати інформацію, а в інший приймати.

Бітова швидкість вибирається залежно від сумарної довжини лінії, характеристик кабелю, і, як правило, наводиться в документації до обладнання.

При великих швидкостях і при значних відстанях необхідно вирішити проблему відбиття сигналу на кінцях лінії. З курсу електротехніки відомо, що основою для мінімізації відбиття є використання узгоджувальних резисторів з номіналами, які відповідають хвильовому опору кабелю. Як правило, у промислових мережах використовують кабель з характеристичним імпедансом *120 Ом*, тому на обох кінцях шини між лініями «А» і «В» підключений узгоджувальний резистор (термінатор) з відповідним опором. Зауважимо, що це потрібно робити лише на кінцевих вузлах шини.

Відзначимо, що стандарт на інтерфейс *RS-485* також не описує типи з'єднувачів. Історично склалось, що підключення вузлів до ліній зв'язку здійснюється за допомогою гвинтового з'єднання, іноді використовують пружинні з'єднувачі, які впровадила компанія *WAGO* (Німеччина). Також використовують з'єднувачі типу *DB-9*. В останньому випадку лінії «А» і «В» підключають відповідно до контактів № 8 та № 3. На клемних колодках деяких пристроїв лінії «А» і «В» позначають відповідно «*RxD/TxD-N*» та «*RxD/TxD-P*».

Далі наведені основні моменти, на які необхідно звернути увагу при роботі з портами інтерфейсу *RS-485* в *ПЛК150 OBEH*.

Для написання управляючої програми контролера необхідно мати як мінімум загальне уявлення про принципи роботи з *COM*-портом. Перш ніж ПЛК почне приймати або посылати інформацію необхідно відкрити та налаштувати потрібний порт. Вибір та налаштування параметрів порту визначається апаратною структурою ПЛК. Виходячи з того, що ПЛК *OBEH* не мають операційної системи, а діють під управлін-

ням системи виконання, адрес портів в контролері позначається через його вказівник. У документації на ПЛК можна отримати посилання на порти його інтерфейсів. Так, в *ПЛК150* є два послідовних порти: *RS-232Debug* та *RS-485*. Ці порти мають відповідні адреси: «4» та «0».

Безпосередня робота з портом – відкриття, закриття, налаштування, читання та запис реалізуються через бібліотеку *SysLibCom.lib*. Після відкриття та налаштування порту контролер автоматично починає слухати інтерфейс і збирати інформацію у вхідний буфер. Розмір буфера становить ~1.5 Кб, тому якщо контролер довгий час не прочитував інформацію і при цьому порт був відкритий, необхідно періодично обнуляти вхідний буфер. Зробити це можна, викликаючи функцію читання порту *SysComRead*, до тих пір, поки ця функція не повертатиме нульове значення. Після цього можна починати роботу з пристроями.

Насамперед контролер повинен сформувати необхідну команду, яку потрібно послати в порт. Після формування команди її можна послати в порт. Зробити це можна, використовуючи функцію *SysComWrite*. На входи цієї функції подається масив байт та кількість байтів необхідних для передачі. При успішній відправці функція поверне кількість відправлених байтів. Після відправки команди контролер необхідно перевести в режим читання порту. Відповідь від пристрою може прийти не відразу, а з деякою затримкою. Пов'язано це з двома речами: поперше, пристрою необхідно отримати команду, обробити та підготувати відповідь, на це, як правило, витрачається незначний час, по-друге, в більшості пристроїв організовано затримки відповіді, для реалізації таймаутів. Якщо в пристрої правильно введені мережні налаштування і лінія зв'язку захищена від перешкод, то відповідь від пристрою прийде з невеликою затримкою, як правило, це не більше 50 мс. Якщо лінія зв'язку погана або періодично бувають великі перешкоди, які порушують обмін, то необхідно ввести таймер очікування відповіді, щоб контролер не чекав відповідь до безкінечності, а послав новий запит. Якщо перешкоди проходять часто, то можна організувати декілька повторних запитів однієї і тієї ж команди.

Після посилання команди ПЛК необхідно перевести в режим читання порту, для прийому відповіді. Робиться це функцією *SysComRead*. Під час надходження даних у вхідний буфер, ця функція їх звідти бере і передає у вхідний масив. При цьому функція повертає кількість прийнятих байтів. Інформацію, що приймається, можна аналізувати відразу або збирати в більший буфер (тобто склеювати кадр), і далі обробляти повністю прийняту відповідь.

2.5. Послідовність проведення лабораторної роботи

2.5.1. Підготовка ПЛК для відправлення повідомлень

Отже, скопіюйте файл з проектом з сервера та завантажте його в середовище *CoDeSys* для подальшого завантаження в ПЛК. Нижче наведено лістинг проекту з блоком об'явлених змінних та програмою, яка написана на мові *ST*.

Блок об'явлених змінних для *POU PLC_PRG*.

```
PROGRAM PLC_PRG
VAR
    port_opened: BOOL:=FALSE;
    com_handle: DWORD;
    com_num: PORTS:=0;
    com_set: COMSETTINGS;
    com_setEx: COMSETTINGSEX;
    res: BOOL;
    tOn1: TON;
    start_tmr: BOOL:=FALSE;
    tr1: R_TRIG;
    snd_str: STRING:='TEST1$N';
    port_init: BOOL;
    rcvbuf: ARRAY [0..1023] OF BYTE;
    sz: DWORD;
END_VAR
```

Текст програми *POU PLC_PRG*.

(*Приклад видачі в інтерфейс RS-485 кожну секунду рядка "TEST1" *)

(*Налаштування порту 9600, 7 біт, немає парності, один стоп біт *)

(*Відкриття порту*)

IF NOT port_opened THEN

 com_handle:=SysComOpen(com_num);

 IF com_handle<>16#FFFFFFFF THEN

 port_init:=TRUE;

 ELSE

 SysComClose(com_num);

 com_handle:=SysComOpen(com_num);

 port_init:=TRUE;

 END_IF

com_setEx.Size:=SIZEOF(com_setEx);

com_setEx.Port:=com_num;

com_setEx.dwBaudRate:=9600;

com_setEx.byStopBits:=0;

(*byStopBits:BYTE; Кільк. стоп. біт 0=один, 1=полтора, 2=два *)

com_setEx.byParity:=0;

(* byParity:BYTE; Режим перев. на парність 0=нет, 1=непар, 2=пар*)

com_setEx.dwTimeout:=0;

(* DWORD; Не використовується, повинно бути =0 *)

com_setEx.dwBufferSize:=0;

(* DWORD; Не використовується, повинно бути =0 *)

com_setEx.dwScan:=0;

(*DWORD; Не використовується, повинно бути =0 *)

com_setEx.cByteSize :=7;

(*BYTE; Довжина символу в бітах 5-8 біт. *)

res:=SysComSetSettingsEx(com_num,ADR(com_setEx));

(*Успішно відкрили*)

```
IF NOT res THEN
    port_opened:=TRUE;
END_IF
END_IF
```

```
IF NOT port_opened THEN
    RETURN;
END_IF
```

```
ton1(In:=start_tmr, pt:=t#1s);
start_tmr:=TRUE;
tr1(clk:=ton1.Q);
IF tr1.Q THEN
    SysComWrite(com_num,ADR(snd_str),LEN(snd_str),0);
    start_tmr:=FALSE;
END_IF
```

```
sz:=SysComRead(com_num,ADR(rcvBUF),1024,0);
```

Запропонований проект є навчальним та має на меті лише демонстрацію можливостей ПЛК щодо управління послідовним портом для відправлення повідомлень. Тому буде розглянуто лише основні елементи проекту.

Отже, в блоці *PLC_PRG* реалізовано такий алгоритм:

- перевірка на готовність до роботи та відкриття порту;
- встановлення параметрів порту (швидкість та формат кадру);
- якщо порт відкритий без похибок, то формування кожному секунду повідомлень;
- читання буфера даних у кожному циклі ПЛК після його очищення.

Отже, скопіюйте проект та завантажте його код в ПЛК. Для завантаження коду до ПЛК використовуйте інтерфейс *Ethernet*. За необхідністю налаштуйте проект. Далі запустіть проект на виконання та спостерігайте за змінними проекту.

2.5.2. *Приймання повідомлень через HyperTerminal*

Після завантаження програми до ПЛК перевірте наявність повідомлень в буфері *COM*-порту. Для цього запустіть програму *HyperTerminal* та налаштуйте з'єднання. У разі успішного отримання повідомлень відключіться від порту та закрийте програму.

2.5.3. *Обмін даними за допомогою програми COM Port Toolkit*

Запустіть на ПК програму моніторингу *COM*-порту та створіть і налаштуйте з'єднання (профіль) роботи з *COM*-портом. Спостерігайте за результатами приймання повідомлень в інформаційному полі програми. Далі змініть умови приймання повідомлень, збільшивши та зменшивши швидкість передавання даних у два рази (наприклад, для швидкості 9600 біт/с подвійна швидкість буде 19 200 біт/с, а половинна – 4800 біт/с). Зафіксуйте отримані результати в *HEX*-форматі та *ASCII*-коді.

Побудуйте діаграму, яка демонструє порядок слідування біт для перших двох символів повідомлення, які відправляє ПЛК (це символи «Т» та «Е»). Нижче побудуйте діаграму для подвійної та половинної швидкості. Порівняйте отримані результати.

2.6. Оформлення результатів проведення заняття та контрольні питання

Складіть звіт про виконану лабораторну роботу, де наведіть схему з'єднань, умови обміну даними та діаграму повідомлень, що відправляються. Також вкажіть, яке повідомлення прийняте на однаковій швидкості та подвійній швидкості, для текстового (*ASCII*) та символного режимів (*HEX*).

За підсумками проведення заняття дайте логічне пояснення отриманим результатам та інтерпретуйте побудовані діаграми.

Крім того, студент повинен відповісти на контрольні питання, приклади яких наведено нижче:

- 1) Які лінії використовують для реалізації *RS-485*?
- 2) Який метод кодування інформації застосовано в інтерфейсі *RS-485*?
- 3) Яким чином стало можливим багатоточкове з'єднання в інтерфейсі *RS-485*?
- 4) Які переваги інтерфейсу *RS-485* в порівняно з інтерфейсом *RS-232* при його використанні в умовах промисловості?
- 5) Яким чином працюють автоматичні перетворювачі інтерфейсів *RS232/RS485* та *USB/RS485*?
- 6) Який режим обміну реалізований в інтерфейсі *RS-485*?
- 7) Для чого використовують стартовий та стоповий біти при символному обміні даними?
- 8) Яким чином працює алгоритм перевірки на парність?
- 9) Чим відрізняється принцип кодування бітів в інтерфейсах *RS-232* та *RS-485*?

Лабораторна робота 3

ПРИНЦИПИ ВЗАЄМОДІЇ DTE-ПРИСТРОЇВ З GSM-МОДЕМОМ

3.1. Мета та час проведення лабораторної роботи

На цю лабораторну роботу відповідно до робочої програми вивчення дисципліни «Комп'ютерні мережі» відводиться 2 акад. год. аудиторного та 2 акад. год. позааудиторного часу (СРС).

Цілі лабораторної роботи:

- ознайомитися з правилами роботи з *GSM*-модемом та *AT*- командами для управління модемами;
- закріпити на практиці отримані теоретичні знання про стандарти на послідовні інтерфейси *RS-232* та *RS-485*;

- отримати практичні навички роботи з *GSM*-модемами для відправлення *SMS*-повідомлень з ПК за допомогою утиліти *HiperTerminal*;
- ознайомитися з принципами формування *AT*-команд у *ПЛК150 ОВЕН*.

3.2. Апаратне забезпечення лабораторної роботи

Лабораторна робота виконується на робочих місцях з ПК та підключеними до них ПЛК, які встановлені на стендах. Для організації зв'язку з телефоном абонента використовується *GSM*-модем типу *TC35 UART/232* [4] виробництва компанії *Siemens AG* (Німеччина).

На рис. 3.1 наведено схему з'єднання ПК та *GSM*-модема за допомогою модемного кабелю, схему якого показано на рис. 3.2.

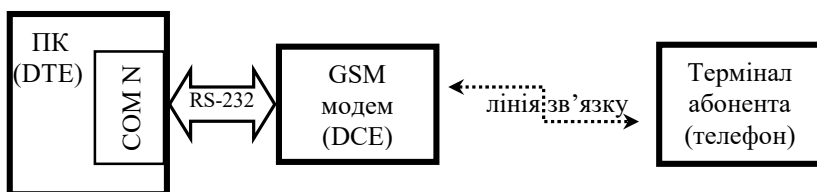


Рисунок 3.1 – Схема з'єднання ПК та *GSM*-модема через інтерфейс *RS-232*

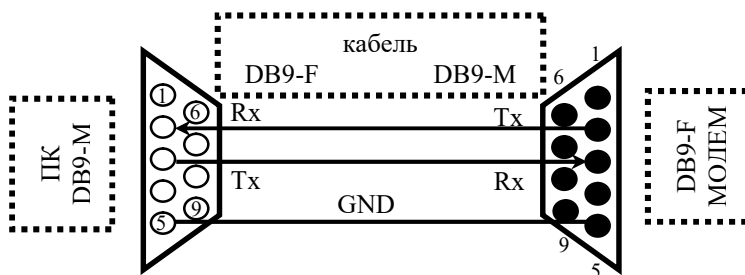


Рисунок 3.2 – Схема кабелю для з'єднання ПК та *GSM*-модема через інтерфейс *RS-232*

На рис. 3.3 наведено схему з'єднання ПЛК та GSM-модема за допомогою модемного кабелю, схему якого показано на рис.3.4. На рис. 3.4 контакти управляючих ліній «4» (*DTR*), «6» (*DSR*) та «7» (*RTS*) з'єднані. З шостого контакту модем вмикає готовність, на 4-й та 7-й контакти. Це потрібно для реалізації апаратного управління потоком даних всередині модема шляхом імітації сигналів готовності та запиту від контролера.

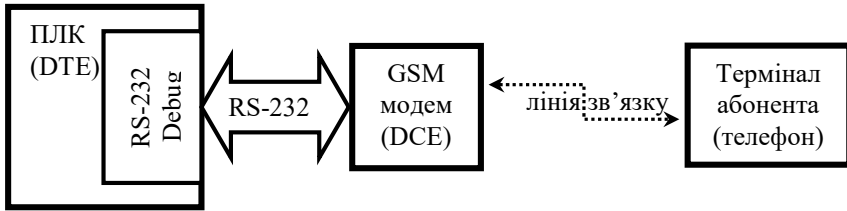


Рисунок 3.3 – Схема з'єднання ПЛК та GSM-модема

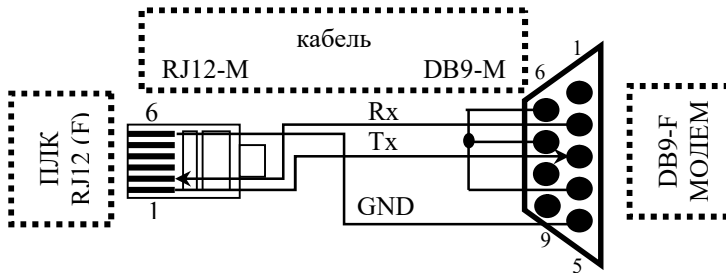


Рисунок 3.4 – Схема кабелю для з'єднання ПК та GSM-модема

3.3. Програмне забезпечення лабораторної роботи

Організація обміну даними між ПК та підключеним до нього GSM-модемом в лабораторній роботі здійснюється за допомогою програми *HyperTerminal*.

Для реалізації обміну даними між ПЛК з підключеним GSM-модемом в лабораторній роботі застосовується середовище *CoDeSys*, в

якому створений проект для ПЛК150 ОВЕН. Перевірка та моніторинг АТ-команд, які надсилає ПЛК, здійснюється за допомогою програми *COM Port Toolkit*.

3.4. Короткі теоретичні відомості

Для організації зв'язку між пристроями за допомогою телекомунікаційного обладнання міжнародними стандартами [1] запроваджено низку термінів, які визначають функціональну належність кожного елемента системи зв'язку, а саме:

- 1) *ME (Mobile Equipment)* – мобільне обладнання;
- 2) *MS (Mobile Station)* – мобільна станція;
- 3) *TA (Terminal Adapter)* – кінцевий пристрій-перетворювач;
- 4) *DCE (Data Communication Equipment)* – кінцеве обладнання каналу зв'язку (звичайний або *GSM*-модем, комутатори, хаби та конвертери) для передавання даних.

Стосовно організації зв'язку по дротових або *GSM*-каналах за допомогою модемів, підключених до керуючого пристрою по послідовному інтерфейсу, додатково використовують терміни для позначення кінцевих керуючих пристроїв:

- 1) *TE (Terminal Equipment)* – кінцеве обладнання;
- 2) *DTE (Data Terminal Equipment)* – кінцеве обладнання для перетворення інформації користувача (ПК, роутер, касовий апарат, термінал *POS* тощо) в дані для передавання по лінії зв'язку.

Існує також пасивне комунікаційне обладнання. Це *DSU/CSU (Data Service Unit / Channel Service Unit)* – пристрої, які застосовують для з'єднання ліній зв'язку, розгалуження сигналів, а також з'єднувачі для підключення до пристроїв.

Під час підключення роутера користувача до провайдера за допомогою телефонних ліній зв'язку він налаштовується в якості пристрою *DTE*, проте розтер провайдера буде пристроєм *DCE*. Різниця у тому, що при-

стрій *DTE* має право ініціювати сеанс зв'язку, а пристрій *DCE* лише слухає лінію.

Отже, для роботи з *GSM*-модемами застосовують *AT*-команди (перелік команд *Hayes*). Цей набір команд був розроблений компанією у 1997 році для модема власної розробки «*Smartmodem 300 baud*». Цей перелік команд вміщує низку коротких текстових повідомлень, які об'єднуються для формування повних операцій. Наприклад, це набирання телефонного номера, початок з'єднання або параметри підключення.

Для розпізнавання *AT*-команд вони мають бути занотованими в специфічному вигляді. Кожна команда починається з символів *AT* або *at* (від англ. *Attention*), які доповнені однією або декількома командами. Відправлення команд здійснюється натисканням на клавішу «*Enter*». Модем оброблює команди, якщо він знаходиться в «командному режимі» або *offline*. Інакше він не відповідає та не виконує ніякі команди.

Стандартом на перелік *AT*-команд є документ під назвою *Data Transmission Systems and Equipment – Serial Asynchronous Automatic Dialing and Control*, більш відомий як *TIA/EIA-602* [2]. В подальшому організація *ITU-T* (*International Telecommunication Union - Telecommunication sector* – сектор стандартизації електрозв'язку Міжнародного союзу електрозв'язку), випустила документ *V.250*, який вміщує весь перелік команд *TIA/EIA-602* та декілька додаткових команд. Пізніше був оприлюднений стандарт *TIA/EIA-602-A*, який мав лише посилання на *V.250* та невелику історичну довідку розвитку стандарту.

Опис *AT*-команд наведено у документі *SIM300_ATC_V1.06* від 2006-12-04, який поширюється компанією *SIMCom Wireless Solutions Co., Ltd* (Китай) [2, 4].

3.4.1 Синтаксис *AT*-команд

Існують три типи переліку команд:

- основний (*basic*);
- додатковий (*S-parameter*);
- розширений (*extended*).

Усі команди в кожному рядку починаються з префікса AT. Закінчення команд позначається введенням символів переведення рядка <CR> (0D_{hex}).

Команди з основного переліку мають формат «AT<x><n>» або «AT&<x><n>», де «<x>» – це команда та «<n>» – аргумент команди. Наприклад, для включення режиму «*ЭХО*» необхідно відправити команду «ATE1». В цьому режимі модем (*DCE*) повертає до ПК (*DTE*) отримані від нього символи.

Команди з додаткового переліку *S*-типу мають формат «ATS<n>=<m>», де «<n>» – це індекс *S*-реєстру для встановлення та «<m>» – значення для присвоєння.

Команди з розширеного переліку бувають чотирьох типів:

- команда типу «*test*» для перевірки модема на його функціональність та визначення діапазону можливих значень параметрів – «AT+<x>=?»;
- команда типу «*read*» для визначення поточних налаштувань модема та значень параметрів – «AT+<x>?»;
- команда типу «*write*» для встановлення в модемі налаштувань користувача – «AT+<x>=<...>»;
- команда типу «*execution*» слугує для зчитування та виконання незмінних параметрів у модемі – «AT+<x>».

На *AT*-команди модем відповідає залежно від типу та змісту команди, а також може генерувати повідомлення про різні події: виклик, отримання *SMS* тощо. У табл. 3.1 наведені відповіді та повідомлення модема в текстовому та кодовому форматі.

Далі наведено приклад *AT*-команд з відповідями модему та коментарями.

AT+CCUG?	<i>Запит поточних налаштувань модема</i>
+CCUG: 1, 10, 1	<i>Відповідь модема</i>
OK	

AT+CCUG= ,9 *Встановлення нового значення другого параметру*
 ОК
 AT+CCUG? *Запит поточних налаштувань модема*
 +CCUG: 1, 9, 1 *Відповідь модема*
 ОК

Таблиця 3.1 – Повідомлення та відповіді модема на AT-команди

Текстове повідомлення	Код повідомлення	Розшифрування повідомлення
ОК	0	Готовий до виконання команд
CONNECT	1	З'єднання встановлено. <i>DCE</i> перемикається з «командного» режиму в режим передавання даних <i>online</i>
RING	2	<i>DCE</i> визначило вхідний виклик з мережі
NO CARRIER	3	З'єднання розірвано або неможливо встановити з'єднання
ERROR	4	Команда не розпізнана, перебільшена довжина рядка, недопустиме значення параметра або інші помилки в командному рядку
NO DIALTONE	6	Не визначений режим тонового виклику
BUSY	7	Сигнал режиму чергування
NO ANSWER	8	«@» використовується модифікатор виклику. Немає відповіді від абонента, який викликається протягом 5 с
CONNECT <text>	визначена виробником	Специфічна інформація для <i>DTE</i> : швидкість, наявність помилок, статус тощо

Як відомо з попередніх робіт, для управління потоком даних використовують два способи: програмний та апаратний. Нагадаємо, що програмне управління потоком полягає у відправці в модем двох символів для старту «XON» та зупинки «XOFF» потоку даних. Такий спосіб зручний при використанні «мінімального» кабелю. Для включення програмного управління необхідно послати команду «AT + IFC = 1, 1». Однак для того, щоб цей режим залишився активним після перезавантаження модема, необхідно подати команду «AT & W» для запису параметрів налаштування в енергонезалежну пам'ять модема.

Апаратне управління потоком даних реалізовано за допомогою управляючих сигналів *RTS/CTS*. Якщо буфер пам'яті модема заповнений, то сигнал *CTS* скидається (*FALSE*) для тимчасового припинення передавання даних.

Нижче поданий перелік деяких *AT*-команд:

- 1) *ATE0* – вимкнути режим «XO»;
- 2) *ATE1* – увімкнути режим «XO»;
- 3) *AT+IPR=<швидкість>* – встановити швидкість обміну даними для послідовного порту;
- 4) *AT&W* – записати в енергонезалежну пам'ять раніше встановлені параметри;
- 5) *AT+CREG?* – запит інформації про реєстрування в мережі *GSM*.
Відповідь модема буде такого формату: «+CREG:<n>,<m>»,
де «n» може набувати таких значень:

- 0 – звіт про реєстрування в мережі вимкнений;
- 1 – звіт про реєстрування в мережі увімкнений;

«m» може набувати таких значень:

- 0 – не зареєстровано;
- 1 – зареєстровано;
- 2 – йде пошук мережі;
- 3 – в реєстрації відмовлено;
- 4 – зареєстровано в роумінгу;

6) ATD<номер> CONNECT<швидкість> – встановлення CSD-з'єднання;

7) AT+CMGF=1 – встановлення режиму передавання текстових повідомлень, тобто режиму для передавання SMS-повідомлень;

8) AT+CMGS=<номер> відповідь модема >
<текст> відповідь модема <текст>

Ctrl+Z (1A_{hex}) відповідь модема **OK** – для відправлення SMS-повідомлення.

Відповідь модема починається та закінчується послідовністю символів переведення рядка та повернення каретки: <0D_{hex}> та <0A_{hex}>.

Якщо модем отримав SMS-повідомлення, то він генерує звіт такого змісту:

+CMTI:<пам'ять>,<індекс>,

де <пам'ять> – тип використаної пам'яті модема, де записано повідомлення;

<індекс> – порядковий номер повідомлення в пам'яті.

Прочитати прийняте повідомлення можна за допомогою команди:
AT+CMGR=<індекс>,0.

3.5. Послідовність проведення лабораторної роботи

3.5.1. Взаємодія ПК та GSM-модема для відправлення SMS-повідомлення

Після підключення модема до COM-порту ПК та вмикання його живлення запустить утиліту *HyperTerminal* та налаштуйте інформаційне з'єднання ПК з модемом

AT (*перевірка зв'язку з модемом *)
OK

AT+CPIN=1111 (*введення PIN-коду, якщо необхідно*)
OK

ATE0 (*вимкнення луна в каналі RS485*)
OK

AT+CBST=7 (*вибір швидкості обміну 9600 біт/сек*)
OK

AT+CMEE=1 (*увімкнення виведення коду похибок*)
OK

AT+CMGF=1 (*текстовий формат повідомлення *) OK

AT+CSCS="GSM" (*встановлення алфавіту GSM для SMS-повідомлень*)
OK

AT&W (*запам'ятовування налаштувань*)
OK

Далі, використовуючи *AT*-команди, відправте коротке повідомлення на номер абонента (за вказівкою викладача). Приклад відправлення *SMS*-повідомлення наведено нижче.

AT+CMGS="*****" (*відправка *SMS*: вказати номер та натиснути Enter*)

> Hello, friend (*модем виводить символ очікування тексту «>», далі набрати текст повідомлення*)

+CMGS: 118 (*в кінці тексту ввести CTRL+z*)

OK (*модем повідомляє про виконання команди відправлення повідомлення*)

AT+CNMI=2,1,0,0,0 (*формат повідомлення модема про отримання *SMS**)

OK

Після отримання повідомлення з телефону абонента прочитати його можна за допомогою утиліти *HyperTerminal*. Нижче показано повідомлення модема про прийняте *SMS*-повідомлення, команда на читання та команда на видалення *SMS*-повідомлення з пам'яті модема:

+CMTI: "SM",2 (*повідомлення модема про отримання СМС, яке знаходиться в 2-му регістрі пам'яті SIM-картки*)

AT+CMGR=2 (*читання SMS з 2-го регістру*)

+CMGR:"REC UNREAD","+38*****", "05/05/15, 17:35:54+16"

Privet, modem!

OK

AT+CMGD=2 (*очищення 2-го регістру*)

OK

AT+CMGR=2 (*читання пустого регістру*)

OK

3.5.2. Взаємодія ПЛК та GSM-модема

Для отримання первинних навичок з управління *GSM-модемом* з боку ПЛК за допомогою *AT*-команд пропонується використати готовий проект, в якому реалізовано відправлення двох *AT*-команд: команда перевірки готовності модема та команда перевірки реєстрації модема в мережі.

Проект складається з двох *POU* – головного *PLC_PRG* (для ініціації *COM*-порту) і ФБ *send_command_string* (для формування *AT*-команд). Оскільки використовується готовий проект, його лістинг не наводиться, а надаються лише коментарі щодо роботи структурних елементів проекту. У проекті реалізована відправка *AT*-команд за допомо-

гою бібліотеки *SysLibCom.lib*, яка входить до набору стандартних бібліотек *CoDeSys*.

Спочатку в основному блоці реалізовано відкриття потрібного порту з необхідними параметрами налаштування. Якщо порт відкритий та готовий до роботи, то можна формувати команду і посилати її в порт, після цього отримувати відповідь від модема. Читання та запис повинні відбуватися в різних циклах контролера, тому для читання та запису використаний оператор *CASE*, який дозволяє організувати розділення читання та запису за циклами контролера. Змінна *STATUS* визначає що повинен робити контролер – слухати порт або послати вибрану команду. Для відправлення в порт можна заздалегідь підготувати перелік команд або написати додатковий блок, який формуватиме команду і розраховуватиме контрольну суму. У цьому прикладі використовуються дві заздалегідь сформовані команди. Усередині оператора *CASE* викликається блок відправлення та приймання повідомлень.

Для спостереження за ходом виконання програми в проекті створено візуалізацію процесу відправлення команд та отримання відповідей модема. На рис. 3.5 показано фрагмент візуалізації проекту, на якому зображені стан модему та відповіді на *AT*-команди.

Якщо необхідно перевірити роботу ПЛК з відправки запитів до модема, то можна відключити модем від порту *RS-232 Debug* і підключити кабель прошивання ПЛК. Далі запустити програму сканування *COM*-портів і налаштуватись на потрібний *COM*-порт ПК. У разі правильної роботи ПЛК в сканері *COM*-порту з'являтимуться такі повідомлення:

- 1) 41 54 0D – команда перевірки готовності модему, тобто *ASCII*-коди символів <A>, <T>, <CR>;
- 2) 41 54 2B 43 52 45 47 3F 0D – команда перевірки реєстрації модему в мережі, тобто *ASCII*-коди символів <A>, <T>, <+> <C>, <R>, <E>, <G>, <?>, <CR>.


 <p>Ответ на команду AT</p> <p>Проверка регистрации в сети +CREG:<n>,<m></p> <p><n> = 0</p> <p>Расшифровка: 0 – отчет о регистрации в сети выключен 1 – отчет о регистрации в сети включен 2 – отчет о регистрации в сети включен</p> <p><m> = 1</p> <p>Расшифровка: 0 – не зарегистрирован 1 – зарегистрирован 2 – идет поиск сети 3 – в регистрации отказано 4 – неизвестная сеть 5 – зарегистрирован в роуминге</p>	<p>Время ожидания ответа: 100 мс</p> <p>Интервал запросов: 50 мс</p> <p>Номер текущей ошибки: 0</p> <p>Всего ошибок: 310</p>
	<p>Ответ на команду AT:</p> <p>OK</p>
	<p>Ответ на команду AT+CREG?:</p> <p>+CREG: 0,1</p> <p>OK</p>

Рисунок 3.5 – Фрагмент візуалізації проекту взаємодії ПЛК та модема

3.6. Оформлення результатів проведення заняття та контрольні питання

Складіть звіт про виконану лабораторну роботу, де наведіть схему з'єднань, умови обміну даними. За підсумками проведеного заняття студент повинен знати основи роботи з модемами за допомогою AT-команд з боку ПК та ПЛК.

Крім того, студент повинен відповісти на контрольні питання, приклади яких наведено нижче:

- 1) Якого типу бувають команди з переліку AT-команд?
- 2) Чому при з'єднанні модема та ПЛК по інтерфейсу RS-485 потрібно вимикати режим «ЛУНА»?
- 3) Які налаштування потрібно зробити для підключення модема до ПК за допомогою програми *HyperTerminal*?
- 4) Які режими обміну даними забезпечує GSM-модем?
- 5) Які повідомлення і в якому форматі може відправляти модем у випадку виникнення помилок?
- 6) Яка процедура роботи ПЛК з відправки AT-команд за допомогою бібліотеки *SysLibCom.lib*?

Лабораторна робота 4

ВЗАЄМОДІЯ ПЛК150 *ОВЕН* З *GSM*-МОДЕМОМ ДЛЯ ВІДПРАВЛЕННЯ *SMS* НА ТЕЛЕФОН АБОНЕНТА

4.1. Мета та час проведення лабораторної роботи

На цю лабораторну роботу відповідно до робочої програми вивчення дисципліни «Комп'ютерні мережі» відводиться 2 акад. год. аудиторного та 2 акад. год. позааудиторного часу (СРС).

Цілі лабораторної роботи:

- закріпити на практиці отримані теоретичні знання про стандарти на послідовні інтерфейси *RS-232* і *RS-485* на прикладі взаємодії ПЛК та модема;
- ознайомитися з реалізацією прямого управління з боку ПЛК150 *ОВЕН* інтерфейсом *RS-232 Debug* для передачі *SMS*-повідомлень за допомогою *GSM*-модема на прикладі використання бібліотек *Sms.lib* та *SmsOwenLib.lib*.

4.2. Апаратне забезпечення лабораторної роботи

Лабораторна робота виконується на робочих місцях з ПК та підключеними до них ПЛК, які встановлені на стендах. Для організації зв'язку з телефоном абонента використовується *GSM*-модем типу *TC35 UART/232* виробництва компанії *Siemens AG* (Німеччина).

Параметри налаштування послідовного інтерфейсу для зв'язку з модемом мають такі значення: 9600, 8-n-1, без управління потоком даних. Для зв'язку ПЛК і модема використовується інтерфейсний кабель типу *DCE-DTE* (прямий, мінімальний) зі з'єднувачем *DB9-M* – для підключення до модема і *RJ12 (6P6C)* – для підключення до інтерфейсу *RS-232 Debug* на передній панелі *ПЛК150 ОВЕН*. Схеми з'єднань пристроїв та кабелю наведені на рис. 3.3 та 3.4.

4.3. Програмне забезпечення лабораторної роботи

Для управління послідовним портом *RS-232 Debug* та передавання *SMS*-повідомлень необхідно використати проект з програмою користу-

вача, яка створена в середовищі *CoDeSys*. При цьому використовуються бібліотеки, які написані спеціально для контролерів *OBEH*. Пропонується використати готові проекти, які необхідно скоректувати відповідно до завдання. Перевірка та моніторинг *AT*-команд, які надсилає ПЛК, здійснюється за допомогою програми *COM Port Toolkit*.

4.4. Короткі теоретичні відомості

Перший приклад демонструє можливість застосування *ПЛК150 OBEH* як засіб диспетчеризації та моніторингу автономних технологічних об'єктів. В першому прикладі реалізована відправка повідомлень про спрацьовування будь-якого з шести дискретних датчиків типу «сухий контакт». Це можуть бути датчики доступу в приміщення, загазованості, досягнення граничних значень (рівень, тиск, температура тощо).

Другий приклад демонструє можливість роботи *GSM*-модема одночасно в двох режимах: передавання *SMS*-повідомлень та обмін даними по *CSD*-каналі. Обмін даними по *CSD*-каналі здійснюється періодично для скачування архівів з *flash*-пам'яті ПЛК, а *SMS*-повідомлення ПЛК відправляє у випадку аварій на автономному технологічному об'єкті.

Бібліотеки для реалізації описаних прикладів надаються безкоштовно компанією *OBEH* для ПЛК 100-ої серії власного виробництва.

До складу бібліотеки *Sms.lib* входить ФБ *GSM_Modem_SMS*, який здійснює відкриття та налаштування *COM*-порту ПЛК, відправлення *SMS*-повідомлення та скидання біта у разі успішного відправлення *SMS*-повідомлення.

Бібліотека *SmsOwenLib.lib* призначена для роботи на контролерах *OBEH* із зовнішнім модемом (необов'язково *GSM*-модемом): для відправлення та отримання *SMS*-повідомлень, фіксації вхідного виклику та первинної конфігурації самого модема. Вона має у своєму складі окремі ФБ для захоплення та звільнення порту ПЛК, конфігурування модема, для відправлення та прийому повідомлень. Особливістю цієї бібліотеки є необхідність наявності підключеного до середовища *CoDeSys* ПЛК,

тому що бібліотечні компоненти працюють лише всередині ПЛК. Тому відладка проектів неможлива в емуляторі *CoDeSys*. Крім того, ця бібліотека потребує підключення до проекту стандартної бібліотеки *UNM.lib*, яка слугує для управління інтерфейсними мережними портами контролера. Вказана бібліотека працює з програмними модулями типу *Universal Network Module [VAR]*, які додатково створюються та налаштовуються в конфігурації ресурсів ПЛК. Для модуля *UNM* створюється необхідний інтерфейс. В даному випадку це буде модуль *Debug RS-232 [SLOT]* зі стандартними параметрами налаштування: 9600, 8-n-1, без управління потоком даних, які співпадають з параметрами налаштування самого модема. Програмні модулі, які створені в конфігурації ресурсів ПЛК, мають адреси для вказівників, починаючи з нульової.

4.5. Послідовність проведення лабораторної роботи

4.5.1. Підготовка ПЛК для відправлення SMS-повідомлень за допомогою бібліотеки SMS.lib

Скопіюйте файл з проектом з сервера та завантажте проект у середовище *CoDeSys 2.3*. Нижче наведено лістинг проекту з блоком оголошених змінних та власне з програмою.

Блок об'явлених перемених для *POU PLC_PRG*.

```
PROGRAM PLC_PRG
VAR
    sPhone: ARRAY[0..3] OF STRING(15);
    bSend: BOOL;
    rsv: STRING;
    GSM_Modem_SMS1: GSM_Modem_SMS;
    q1: BOOL;
    q2: BOOL;
    F_TRIG1: F_TRIG;
    R_TRIG1: R_TRIG;
    Textn: STRING;
    in: BOOL;
END_VAR
```

Блок с текстом программы *POU PLC_PRG*.

(*Приклад відправлення SMS на телефон абонента у разі спрацьовування будь-якого з шести дискретних датчиків, які підключені до входів ПЛК, номер порту RS-232 Debug – 4*)

```
sPhone[0]:='+38*****';
in:=i1 OR i2 OR i3 OR i4 OR i5 OR i6;
IF i1=TRUE THEN Textn:='Авария-вход 1';END_IF;
IF i2=TRUE THEN Textn:='Авария-вход 2';END_IF;
IF i3=TRUE THEN Textn:='Авария-вход 3';END_IF;
IF i4=TRUE THEN Textn:='Авария-вход 4';END_IF;
IF i5=TRUE THEN Textn:='Авария-вход 5';END_IF;
IF i6=TRUE THEN Textn:='Авария-вход 6';END_IF;

F_TRIG1(CLK:=in , Q=>q1 );
R_TRIG1(CLK:=in , Q=>q2 );
bSend:=NOT(q1 OR q2);
GSM_Modem_SMS1(Time_rcves_SMS:=#5s, Time_delete_SMS:=#5s,
com_num:=COM4, sPhone_in:=sPhone, send_SMS_text:=Textn,
Text_or_PDU:=TRUE, bSend:=bSend, rcvedSMS=>rsv );
```

Запропонований проект є навчальним і має на своїй меті лише демонстрацію можливостей *ПЛК150 OBEH* з управління послідовним портом *RS-232 Debug* для відправлення за допомогою *GSM*-модема *SMS*-повідомлень. Тому розглянемо тільки основні елементи програми. В організаційному блоці *PLC_PRG* послідовно реалізовано такий алгоритм:

- заповнення масиву телефонними номерами абонентів;
- алгоритм оцінки стану дискретних входів та формування коротких повідомлень;
- запуск тригера для формування команди на відправлення повідомлення через бібліотечний функціональний блок з бібліотеки *SMS.lib*.

Отже, відкомпілюйте завантажену в *CoDeSys* програму, підключіться до ПЛК за допомогою програмного комунікаційного модуля

Gateway та завантажте готовий проєкт в ПЛК. Для завантаження коду програми використовуйте інтерфейс *Ethernet*. При необхідності проведіть налагодження проєкту. Запустіть ПЛК на виконання програми користувача і не відключайтеся середовищем *CoDeSys* від ПЛК для спостереження за змінними процесу. Імітуйте спрацьовування датчика, використовуючи емулятор дискретних сигналів. Далі за допомогою стільникового телефону приймайте повідомлення від ПЛК.

4.5.2. Прийом повідомлень програмою *HyperTerminal*

Приймачем *SMS*-повідомлень може бути також автоматизоване робоче місце (АРМ) оператора, тобто ПК з підключеним до нього *GSM*-модемом. Тобто, якщо в якості абонента використовувати ПК з підключеним до нього модемом, необхідно використати додаткове прикладне програмне забезпечення. Але, як і раніше, можна використати програму *HyperTerminal*.

Після завантаження програми в ПЛК перевірте утилітою *HyperTerminal* наявність повідомлень в буфері *COM*-порту ПК. Для цього запустіть утиліту і відповідним чином налаштуйте з'єднання. Використовуйте *AT*-команди для читання прийнятого повідомлення.

+СМТІ: "SM",2 (*сообщение модема о полученном СМС, сообщение во 2-й ячейке*)

AT+СMGR=2 (*чтение СМС из 2-й ячейки*)

+СMGR: "REC UNREAD", "+38*****", "10/10/14,17:35:54+16"

Privet,modem!

OK

AT+СMGD=2 (*очистка 2-й ячейки*)

OK

AT+СMGR=2 (*чтение пустой ячейки*)

OK

Переконавшись в отриманні повідомлень, відключіться від *COM*-порту ПК і закрийте програму.

4.5.3. Підготовка ПЛК для відправлення SMS-повідомлень за допомогою бібліотеки *SmsOwenLib.lib*

Скопіюйте файл з проектом з сервера і завантажте проект у середовище *CoDeSys 2.3*. Нижче наведений лістинг проекту з блоком оголошених змінних та власне з програмою.

Блок об'явлених змінних для *POU PLC_PRG*.

```
PROGRAM PLC_PRG
VAR
  open_port_1: OpenPort;
  dev_Num: WORD := 0;
  cfg: FB_SMS_CFG;
  cfg_ok: BOOL;
  sms_send: FB_SMS_SR;
  sms_flag_send: BOOL;
  in_ring_flag: BOOL;
  in_sms_flag: BOOL;
END_VAR
```

Організаційний блок *PLC_PRG*, який створений в середовищі *CoDeSys* на мові *CFC*, представлений на рис. 4.1.

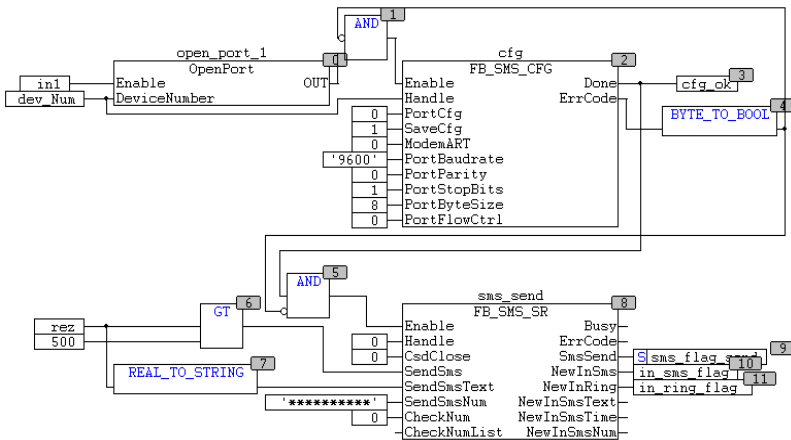


Рисунок 4.1 – Програма для відправлення SMS-повідомлень

У програмі спочатку відкривається інтерфейс зв'язку з модемом (in1), далі налаштовується модем. Якщо немає помилок відкриття порту, то після отримання сигналу готовності порту активується блок відправлення та приймання SMS-повідомлень. У випадку наявності помилок налаштування модему виконання алгоритму повторюється. Якщо модем вже раніше налаштований і його параметри не будуть змінюватись, то можна не використовувати цей ФБ у програмі або викликати його тільки один раз при підключенні нового модема. У цьому прикладі CSD-режим вимкнений (всі вхідні виклики будуть скинуті). На вхід SendSms ФБ *sms_send* подається десятизначний номер одержувача.

Для ініціювання процедури відправлення SMS-повідомлення в програмі передбачений наступний алгоритм, якщо параметр (rez) в якійсь момент часу стане більшим значення уставки (500), формується сигнал (вхід SendSms ФБ *sms_send*). Текстом повідомлення буде саме значення параметра rez. Номер телефона отримувача повідомлення фіксований.

На виході блока SendSms формуються сигнали, які показують статус блока, наявність вхідних викликів, прийняття SMS-повідомлень (текст, час приймання та номер телефона відправника). У разі успішного відправлення повідомлення на вихід SmsSend ФБ *sms_send* буде поданий одиничний імпульс (значення True буде встановлено на один цикл роботи ПЛК).

4.6. Оформлення результатів проведення заняття та контрольні питання

Складіть звіт про виконану лабораторну роботу, де надайте схему з'єднань, умови обміну даними. За підсумками проведеного заняття студент повинен знати основи роботи з GSM-модемами за допомогою AT-команд з боку ПЛК.

Крім того, студент повинен відповісти на контрольні питання, приклади яких наведено нижче:

- 1) Які інтерфейси можна використовувати для підключення GSM-модема до ПЛК150 ОВЕН?
- 2) Яким чином реалізується управління GSM-модемом (DCE) з боку ПЛК (DTE)?
- 3) Якого типу бувають команди з переліку AT-команд?

Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ

для проведення лабораторних занять з курсу

«Комп'ютерні мережі»

(у двох частинах)

для студентів напряму підготовки 050202

«Автоматизація та комп'ютерно-інтегровані технології»

денної та заочної форм навчання

Частина 1

Укладачі: ПОДУСТОВ Михайло Олексійович
ЛИСАЧЕНКО Ігор Григорович
ЛОБОЙКО В'ячеслав Олексійович
ШУТИНСЬКИЙ Олексій Григорович

Відповідальний за випуск М.О. Подустов

Роботу до видання рекомендувала Н. М. Самойленко

Редактор Л.А. Пустовойтова

План 2015 р., поз. 164

Підп. до друку __.__.15. Формат 60×84 1/16. Папір офсетний.

Друк – ризографія. Гарнітура Times New Roman.

Ум. друк. арк. 3,0. Наклад 50 прим. Зам. № __. Ціна договірна.

Видавничий центр НТУ „ХП”.

Свідоцтво про державну реєстрацію ДК № 3657 від 24.12.2009 р.

61002, Харків, вул. Фрунзе, 21.

ТОВ "Видавництво"Підручник НТУ "ХП".

Свідоцтво про державну реєстрацію ДК № 3656 від 24.12.2009 р.

61002, Харків, вул. Фрунзе, 21.