

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Тернопільський національний економічний університет  
Факультет комп'ютерних інформаційних технологій  
Кафедра комп'ютерної інженерії

**Пліський Артем Вікторович**

**Система керування COM портом засобами середовища  
Delphi /  
COM port management system using Delphi tools**

напрямок підготовки: 6.050102 - Комп'ютерна інженерія  
фахове спрямування - Комп'ютерні системи та мережі  
Бакалаврська робота

Виконав студент групи КСМз-41/2  
А.В. Пліський

Науковий керівник:  
Климчук О.І.

Тернопіль - 2018

## ЗМІСТ

Вступ.....	3
1 Аналіз предметної області.....	4
1.1 Огляд стандарту RS-232.....	4
1.2 Різновиди кабелів підключення .....	8
1.3 Аналіз технічного завдання .....	16
2 Дослідження передачі даних стандарту RS-232 .....	17
2.1 Управління потоком .....	17
2.2 Перетворення рівнів RS-232 в TTL рівень.....	18
3 Проектування програмного засобу.....	28
3.1 Розробка інтерфейсу програмного засобу.....	28
3.2 Розробка функцій програмного засобу.....	29
3.3 Тестування та верифікація програмного засобу.....	34
Висновки .....	39
Список використаних джерел .....	40

Змн.	Лист	№ докум.	Підпис	Дата	Система керування COM портом засобами середовища Delphi / COM port management system using Delphi tools	Літ.	Арк.	Акрушів
						8	81	
Розробив		Пліський				ТНЕУ. ННІОТ. КСМзкп-41/2		
Перевір.								
Консульт.		Паздрій І.Р.						
Н. Контр.		Гураль І.В.						
Затвердив		Березький О.М						

## ВСТУП

RS-232 - популярний протокол, застосовуваний для зв'язку комп'ютерів з модемами та іншими периферійними пристроями.

RS-232 - інтерфейс передачі інформації між двома пристроями на відстані до 20 м. Інформація передається по проводах з рівнями сигналів, що відрізняються від стандартних 5В, для забезпечення більшої стійкості до перешкод. Асинхронна передача даних здійснюється з встановленою швидкістю при синхронізації рівнем сигналу стартового імпульсу.

Інтерфейс RS-232-C був розроблений для простого застосування, однозначно визначається по його назві "Інтерфейс між термінальним обладнанням і зв'язковим обладнанням з обміном по послідовному двійковому коду". Кожне слово в назві значиме, воно визначає інтерфейс між терміналом (DTE) і модемом (DCE) з передачі послідовних даних.

Пристрої для зв'язку по послідовному каналу з'єднуються кабелями з 9-ма або 25-ти контактними роз'ємами типу D. Зазвичай вони позначаються DB-9, DB-25, CANNON 9, CANNON 25 і т.д. Роз'єми типів розетки і штирів. Кожен пін позначений і пронумерований. Розташування пінів представлено нижче.

Асоціація електронної промисловості (EIA) розвиває стандарти з передачі даних. Стандарти EIA мають префікс "RS". "RS" означає рекомендований стандарт, але зараз стандарти просто позначаються як "EIA" стандарти. RS-232 був введений в 1962. Стандарт розвивався і в 1969 представлена третя редакція (RS-232C). Четверта редакція була в 1987 (RS-232D, відома також під EIA-232D). RS-232 ідентичний стандартам МККТТ (CCITT) V.24 / V.28, X.20bis / X.21bis і ISO IS2110.

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			3

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд стандарту RS-232

RS-232 - популярний протокол, застосовуваний для зв'язку комп'ютерів з модемами та іншими периферійними пристроями.

RS-232 - інтерфейс передачі інформації між двома пристроями на відстані до 20 м. Інформація передається по проводах з рівнями сигналів, що відрізняються від стандартних 5В, для забезпечення більшої стійкості до перешкод. Асинхронна передача даних здійснюється з встановленою швидкістю при синхронізації рівнем сигналу стартового імпульсу.

Інтерфейс RS-232-C був розроблений для простого застосування, однозначно визначається по його назві "Інтерфейс між термінальним обладнанням і зв'язковим обладнанням з обміном по послідовному двійковому коду". Кожне слово в назві значиме, воно визначає інтерфейс між терміналом (DTE) і модемом (DCE) з передачі послідовних даних.

Пристрої для зв'язку по послідовному каналу з'єднуються кабелями з 9-ма або 25-ти контактними роз'ємами типу D. Зазвичай вони позначаються DB-9, DB-25, CANNON 9, CANNON 25 і т.д. Роз'єми типів розетки і штирів. Кожен пін позначений і пронумерований. Розташування пінів представлено нижче.

Асоціація електронної промисловості (EIA) розвиває стандарти з передачі даних. Стандарти EIA мають префікс "RS". "RS" означає рекомендований стандарт, але зараз стандарти просто позначаються як "EIA" стандарти. RS-232 був введений в 1962. Стандарт розвивався і в 1969 представлена третя редакція (RS-232C). Четверта редакція була в 1987 (RS-232D, відома також під EIA-232D). RS-232 ідентичний стандартам МККТТ (ССТТ) V.24 / V.28, X.20bis / X.21bis і ISO IS2110. На рисунку 1.1 наведено позначення пристроїв передачі даних по стандарту RS-232.

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			4

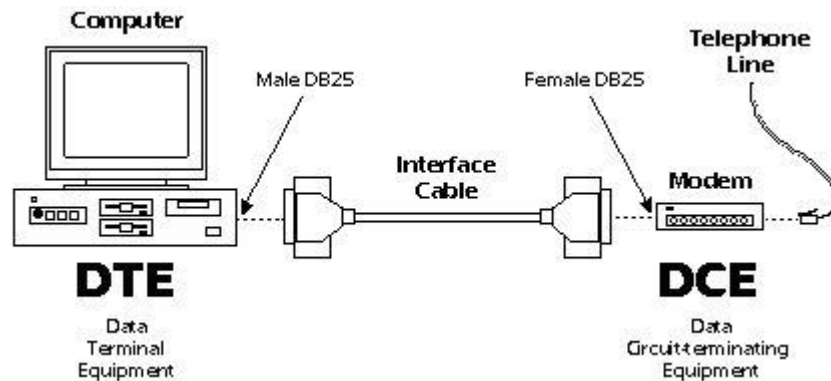


Рисунок 1.1 - Позначення пристроїв

В RS-232 використовуються два рівні сигналів: логічні 1 і 0. Логічний 1 іноді позначають MARK, логічний 0 - SPACE. Логічний 1 відповідають негативні рівні напруги, а логічному 0 - позитивні. Відповідні значення напруг представлені в таблиці 1.1 та 1.2.

Таблиця 1.1 - Рівні сигналів даних

Рівень	Передавач	Приймач
логічний 0	Від +5 В до +15 В	Від +3 В до +25 В
логічний 1	від -5 В до -15 В	Від -3 В до -25 В
Не визначений	Від -3 В до +3 В	

Таблиця 1.2 - Рівні сигналів на виході та на вході

Сигнал	На виході пристрої (driver)	На вході пристрої (terminator)
"Off"	Від -5 В до -15 В	від -3 В до -25 В
"On"	Від 5 В до 15 В	від 3 В до 25 В

На рисунку 1.2 наведено схему передачі сигналів з вказанням напруг. Сигнали після проходження по кабелю ослаблюються і спотворюються.

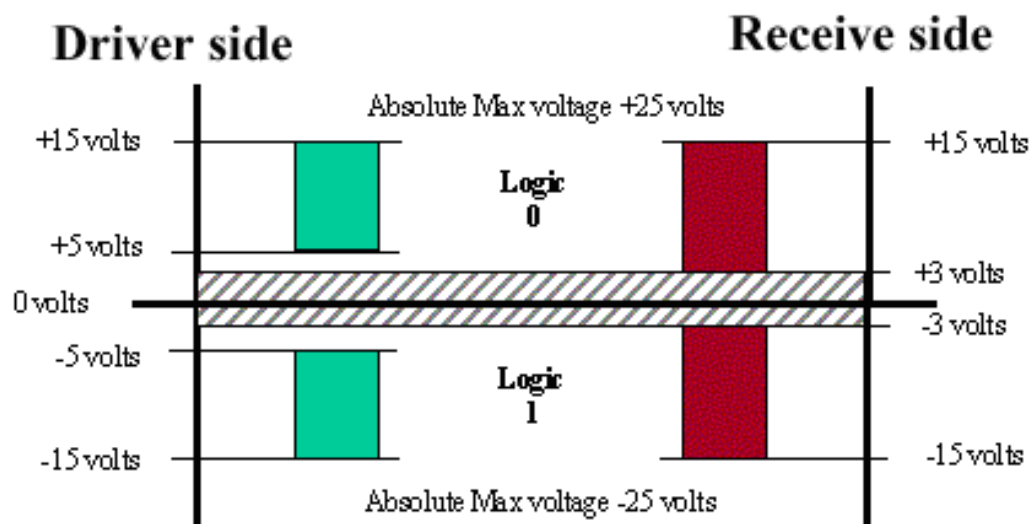


Рисунок 1.2 Схема передачі сигналів

Ослаблення зростає зі збільшенням довжини кабелю. Цей ефект сильно пов'язаний з електричною ємністю кабелю. За стандартом максимальна навантажувальна ємність становить 2500 пФ. Типова погонна ємність кабелю становить 130 пФ, тому максимальна довжина кабелю обмежена приблизно 17 метрами.

Перед з'єднанням двох пристроїв через RS-232, кожен з яких живиться від різних джерел рекомендується вирівняти напруги між їх сигнальними землями перед підключенням. Контакти роз'ємів для DB25 Розетка (мама) та їхня розпіновка приведені в таблиці 1.3.

Таблиця 1.3 - Контакти роз'ємів для DB25 Розетка

Контакт	Позначення	Напрямок	Опис
1	SHIELD	---	Shield Ground - захисна земля, з'єднується з корпусом пристрою і екраном кабелю
2	TXD	->	Transmit Data - Вихід передавача
3	RXD	<-	Receive Data - Вхід приймача
4	RTS	->	Request to Send - вихід запиту передачі даних
5	CTS	<-	Clear to Send - вхід дозволу терміналу передавати дані

Контакт	Позначення	Напрямок	Опис
6	DSR	<-	Data Set Ready - вхід сигналу готовності від апаратури передачі даних
7	GND	---	System Ground - сигнальна (схемна) земля
8	CD	<-	Carrier Detect - вхід сигналу виявлення несучої віддаленого модему
9-19	N / C	-	-
20	DTR	->	Data Terminal Ready - вихід сигналу готовності терміналу до обміну даними
21	N / C	-	-
22	RI	<-	Ring Indicator - вхід індикатора виклику (дзвінка)
23-25	N / C	-	-

Розпіновка контактів для роз'ємів DB9 Розетка (мама) наведені в таблиці 1.3.

Таблиця 1.3 - Контакти роз'ємів для DB9 Розетка (мама)

Контакт	Обозн.	напрямок	опис
1	CD	<-	Carrier Detect
2	RXD	<-	Receive Data
3	TXD	->	Transmit Data
4	DTR	->	Data Terminal Ready
5	GND	---	System Ground
6	DSR	<-	Data Set Ready
7	RTS	->	Request to Send
8	CTS	<-	Clear to Send
9	RI	<-	Ring Indicator

Також існує можливість розпіновки для стандарту RJ-45, розпіновка наведена в таблиці 1.4.

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			7

Таблиця 1.4 - Контакти роз'ємів для RJ-45

Контакт	Обозн.	напрямок	опис
1	RI	<-	Ring Indicator
2	CD	<-	Carrier Detect
3	DTR	->	Data Terminal Ready
4	GND	---	System Ground
5	RxD	<-	Receive Data
6	TxD	->	Transmit Data
7	CTS	<-	Clear to Send
8	RTS	->	Request to Send

З'єднання коннектора RJ-45 чітко не стандартизовано. Даний варіант один з можливих.

## 1.2 Різновиди кабелів підключення

Серед різновидів підключення наявні можливості нуля модемного підключення кабеля по стандарту RS-232 схема підключення приведена на рисунку 1.3.

Розглянемо спочатку DSR сигнал (конт.6). Цей вхід сигналу готовності від апаратури передачі даних. У схемі з'єднань вхід замкнутий на вихід DTR (конт.4).

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			8



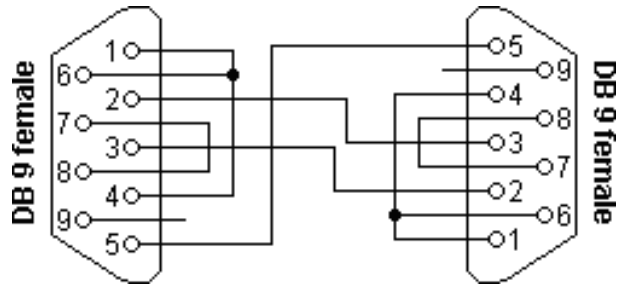


Рисунок 1.3 – Трьох провідний мінімальний

Це означає, що програма не бачить сигналу готовності іншого пристрою, хоча він є. Аналогічно встановлюється сигнал на вході CD (конт.1). Тоді при перевірці сигналу DSR для контролю можливості з'єднання буде встановлено вихідний сигнал DTR.

Це відповідає 99% комунікаційного програмного забезпечення. Під цим мається на увазі, що 99% програмного забезпечення з цим нуль-модемним кабелем візьмуть перевірку сигналу DSR.

Аналогічний спосіб застосовується для вхідного сигналу CTS. В оригіналі сигнал RTS (конт.7) встановлюється і потім перевіряється CTS (конт.8). Поєднання цих контактів призводить до неможливості зависання програм через відсутність відповіді на запит RTS (Рисунок 1.4).

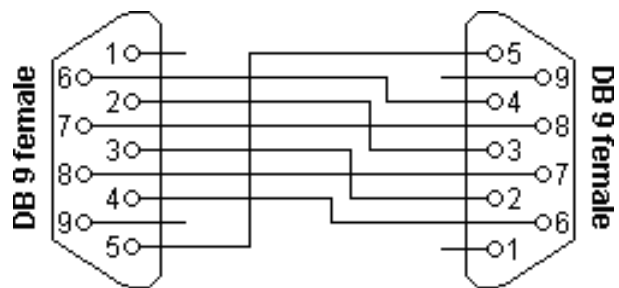


Рисунок 1.4 – 7-дротовий повний

Найефективніших повний нуль-модемний кабель з сімома проводами. Тільки сигнали індикатора виклику і визначення несучої не підключені.

Цей кабель не дозволяє використовувати попередній метод контролю передачі даних. Основна несумісність перехресне з'єднання сигналів RTS і

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			9

CTS. Спочатку ці сигнали використовувалися для контролю потоком даних по типу запит / відповідь. При використанні повного нуль-модемного кабелю більш немає запитів. Ці сигнали застосовуються для повідомлення іншій стороні чи є можливість з'єднання.

Контакти 2 і 3 на 9-ти вивідному роз'ємі D типу протилежні цим же контактам на 25-ти контактному роз'ємі. Тому, якщо з'єднати контакти 2-2 і 3-3 між роз'ємами D25 і D9, вийде комунікаційний кабель. Контакти сигнальної землі Signal Ground (SG) також повинні бути підключені між собою. Згідно таблиці 1.4-1.5.

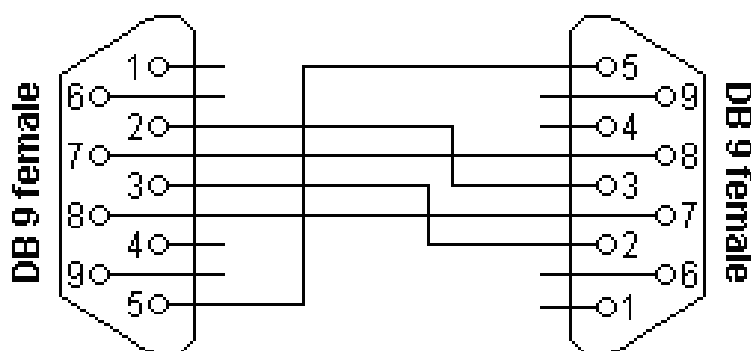


Рисунок 1.5 - 5-провідний з керуванням потоком

Можна знайти або виготовити багато типів кабелів для зв'язку по інтерфейсу RS-232. У цьому нуль модемном кабелі використовується тільки 5 проводів: сигнали даних TXD, RXD, сигнал GND і керуючі сигнали RTS CTS для управління потоком.

Все DTE-DCE кабелі прямого з'єднання, контакти з'єднуються один до одного. Кабелі DTE-DTE і DCE-DCE крос-кабелі.

1. DTE - DCE називається 'прямий кабель'
2. DTE - DTE називається 'нуль-модемний кабель'
3. DCE - DCE називається 'Tail Circuit Cable'

Опис повного нуль-модемного кабелю представлений в таблицях 1.5-1.6.

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			10

Таблиця 1.5 - З'єднання D9- D9

DB9-1		DB9-2	
Receive Data	2	3	Transmit Data
Transmit Data	3	2	Receive Data
Data Terminal Ready	4	6 + 1	Data Set Ready + Carrier Detect
System Ground	5	5	System Ground
Data Set Ready + Carrier Detect	6 + 1	4	Data Terminal Ready
Request to Send	7	8	Clear to Send
Clear to Send	8	7	Request to Send

Опис стандартів передачі даних через інтерфейси D25-D25 наведено в таблиці 1.6.

Таблиця 1.6 - З'єднання D25-D25

DB25-1		DB25-2	
Receive Data	3	2	Transmit Data
Transmit Data	2	3	Receive Data
Data Terminal Ready	20	6 + 8	Data Set Ready + Carrier Detect
System Ground	7	7	System Ground
Data Set Ready + Carrier Detect	6 + 8	20	Data Terminal Ready
Request to Send	4	5	Clear to Send
Clear to Send	5	4	Request to Send

Опис стандартів передачі даних через інтерфейси D9-D25 наведено в таблиці 1.7. Даний з'єднувач RS-232 може бути використаний для перевірки послідовного порту кмпьютера.

Таблиця 1.7 - З'єднання D9-D25

DB9		DB25	
Receive Data	2	2	Transmit Data
Transmit Data	3	3	Receive Data
Data Terminal Ready	4	6 + 8	Data Set Ready + Carrier Detect
System Ground	5	7	System Ground
Data Set Ready + Carrier Detect	6 + 1	20	Data Terminal Ready
Request to Send	7	5	Clear to Send
Clear to Send	8	4	Request to Send

Сигнали даних і управління з'єднані. У цьому випадку дані, що передаються відразу повертаються. Комп'ютер перевіряє власний потік. Це може бути використано для перевірки функціонування порту RS-232 зі стандартним термінальним програмним забезпеченням.

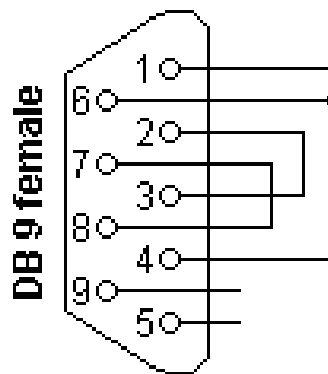


Рисунок 1.6 - DB 9 мама

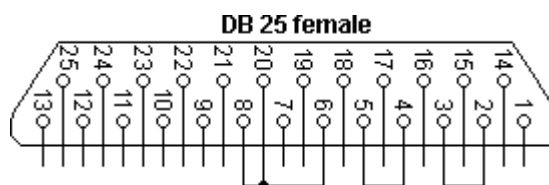


Рисунок 1.7 - DB 25 мама

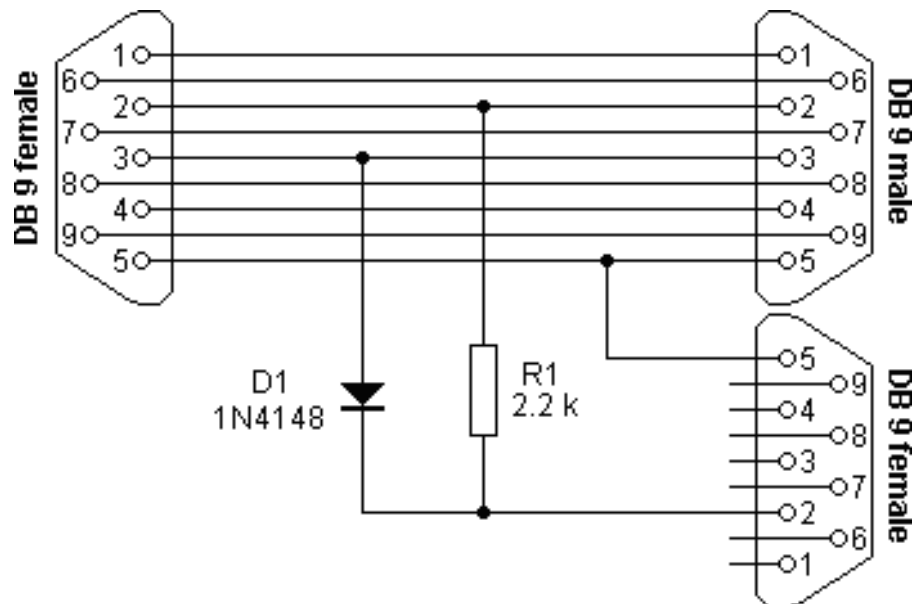


Рисунок 1.8 - Кабель контролю (моніторингу) RS-232

Контроль зв'язку з RS-232 між двома пристроями за допомогою комп'ютера можливий за допомогою кабелю, зображеного на рис. Два роз'єму підключаються до пристроїв, а третій підключається до спостерігача комп'ютера. Цей кабель приймає інформацію від двох джерел тільки на один приймальний порт RS-232. Тому, якщо обидва пристрої почнуть одночасну роботу, контрольована інформація на вході комп'ютера буде порушена. У більшості випадків зв'язок здійснюється в напівдуплексному режимі. Для цих режимів цей кабель буде працювати без проблем.

Довжина кабелю впливає на максимальну швидкість передачі інформації. Довший кабель має велику ємність і відповідно для забезпечення надійної передачі нижчу швидкість. Велика ємність призводить до того, що зміна напруги одного сигнального проводу може передатися на інший суміжний сигнальний провід. Максимальною відстанню зазвичай вважається рівним 15 м, але це не встановлено в стандарті. Ми рекомендуємо використовувати на відстанях до 50 м, але це залежить від типу використовуваного обладнання та характеристик кабелю.

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			13

Таблиця 1.8 - Максимальна довжина кабелю

Швидкість [бод]	Макс. довжина [фути]	Макс. довжина [метри]
19 200	50	15
9 600	500	150
4 800	1000	300
2 400	3000	900

Швидкість передачі інформації по RS-232 вимірюється в Бодах. Ця одиниця названа на честь Еміля Бодо (Jean Maurice-Emile Baudot) (1845-1903), французького інженера по телеграфії, винахідника першого друкувального пристрою для телеграфу (телетайпа), представленого на Міжнародній Телеграфній конференції в 1927. Максимальна швидкість відповідно до стандарту 20000 Бод. Однак сучасне обладнання може працювати значно швидше. Не має значення на скільки швидке (повільне) ваше з'єднання - максимальне число читання за секунду можна встановити за допомогою використовуваного програмного забезпечення.

При передачі по послідовному каналу контроль парності може бути використаний для виявлення помилок при передачі даних. При використанні контролю парності надсилаються повідомлення підраховують число одиниць в групі біт даних. Залежно від результату встановлюється біт парності. Приймальний пристрій також підраховує число одиниць і потім звіряє біт парності.

Для забезпечення контролю парності комп'ютерну систему на безпечній повинні однаково робити підрахунок біта парності. Тобто, визначитися встановлювати біт при парному (even) або непарному (odd) кількості одиниць. При контролі на парність біти даних і біт парності завжди повинні містити парне число одиниць. В протилежному випадку відповідає для контролю на непарність.

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			14

Часто в драйверах доступні ще дві опції на парність: Mark і Space. Ці опції не впливають на можливість контролю помилок. Mark означає, що пристрій завжди встановлює біт парності в 1, а Space - завжди в 0.

Перевірка на парність - це найпростіший спосіб виявлення помилок. Він може визначити виникнення помилок в одному бите, але при наявності помилок в двох бітах вже не помітить помилок. Також такий контроль не відповідає на питання який біт помилковий. Інший механізм перевірки включає в себе Старт і Стоп біти, циклічні перевірки на надмірність, які часто застосовуються в з'єднаннях Modbus.

У цьому прикладі показана структура даних, що передаються з синхронізуючим тактовим сигналом. У цьому прикладі використовується 8 біт даних, біт парності і стоп біт. Така структура також позначається 8E1.

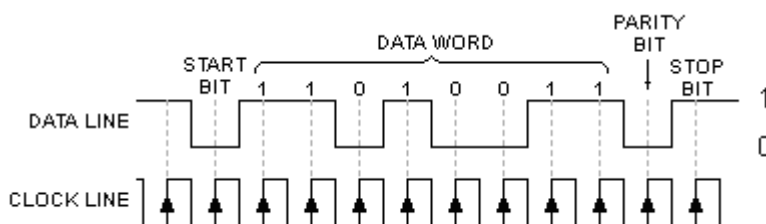


Рисунок 1.9 - Тактовий сигнал - для асинхронної передачі

Примітка: Тактовий сигнал - для асинхронної передачі це внутрішній сигнал

Сигнальна лінія може знаходитися в двох станах: включено і вимкнено. Лінія в стані очікування завжди включена. Коли пристрій або комп'ютер хочуть передати дані, вони переводять лінію в стан вимкнено - це установка Старт біта. Біти відразу після Старт біта є б'юітами даних.

Стоп біт дозволяє пристрою або комп'ютера провести синхронізацію при виникненні збоїв. Наприклад, перешкода на лінії приховала Старт біт. Період між старт і стоп битами постійний, відповідно до значення швидкості обміну, числу біт даних і біта парності. Стоп біт завжди

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			15

включений. Якщо приймач визначає вимкненому стані, коли повинен бути присутнім стоп біт, фіксується поява помилки.

Стоп біт не просто один біт мінімального інтервалу часу в кінці кожної передачі даних. На комп'ютерах зазвичай він еквівалентний 1 або 2 бітам, і це повинно враховуватися програмі драйвера. Хоч, 1 стоп біт найбільш загальний, вибір 2 біт в гіршому випадку трохи сповільнить передачу повідомлення.

Є можливість установки значення стоп біта рівним 1.5. Це використовується при передачі менше 7 бітів даних. У цьому випадку не можуть бути передані символи ASCII, і тому значення 1.5 використовується рідко.

### 1.3 Аналіз технічного завдання

В результаті аналізу технічного завдання булор виділено наступні цілі проектування:

- розробка зручного графічного інтерфейсу;
- забезпечення можливості ручного управління кожним піном інтерфейсу;
- дослідження стандарту RS-232;
- дослідження та використання функцій для роботи із стандартом RS-232;

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			16



## 2 ДОСЛІДЖЕННЯ ПЕРЕДАЧІ ДАНИХ СТАНДАРТУ RS-232

### 2.1 Управління потоком

Управління потоком представляє управляти переданими даними. Іноді пристрій не може обробити дані, що приймаються від комп'ютера або іншого пристрою. Пристрій використовує управління потоком для припинення передачі даних. Можуть використовуватися апаратне або програмне управління потоком.

Апаратний протокол управління потоком RTS / CTS. Він використовує додатково два дроти в кабелі, а не передачу спеціальних символів по лініях даних. Тому апаратне керування потоком не уповільнює обмін на відміну від протоколу Хон-Хoff. При необхідності послати дані комп'ютер встановлює сигнал на лінії RTS. Якщо приймач (модем) готовий до прийому даних, то він відповідає установкою сигналу на лінії CTS, і комп'ютер починає посланку даних. При неготовності пристрою до прийому сигнал CTS не встановлюється.

Програмний протокол управління потоком Хон / Хoff використовує два символи: Хон і Хoff. Код ASCII символу Хон - 17, а ASCII код Хoff - 19. Модем має маленький буфер, тому при його заповненні модем посилає символ Хoff комп'ютера для припинення посланки даних. При появі можливості прийому даних надсилається символ Хон і комп'ютер продовжить пересилання даних. Цей тип управління має перевагу в тому, що не вимагає додаткових ліній, тому щосимволи передаються по лініях TD / RD. Але на повільних з'єднаннях це може привести до значного уповільнення з'єднання, тому що кожен символ вимагає 10 бітів.

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			17

## 2.2 Перетворення рівнів RS-232 в TTL рівень

Два типи пристроїв RS-232, 1488 і 1489 використовуються і зараз. Це ранні представники цього стандарту. Пристрої живиться потужними джерелами живлення, оскільки відповідно до стандарту RS-232 передавачі повинні були забезпечувати мінімальний +5В сигнал низького рівня і мінімальний -5В сигнал високого рівня. Ці рівні сигналів забезпечували стійкість до перешкод після передачі по проводах до приймача. Але це потребувало наявності двополярного джерела живлення, і тому багато материнських плат включали в себе джерело негативної напруги виключно для живлення пристроїв типу 1488 і 1489.

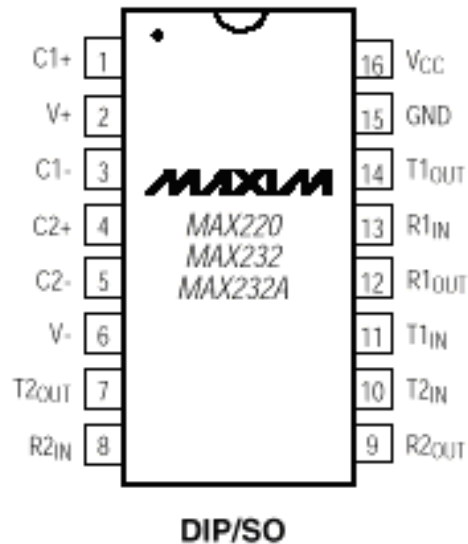
Сімейство мікросхем MAX220-MAX249 лінійних приймально-передавачів призначені для інтерфейсів EIA / TIA-232e і V.28 / V.24, особливо в пристроях, де відсутня напруга  $\pm 12\text{В}$ .

Альтернативна мікросхема ICL232. Це здвоєний приймально-передавач відповідна специфікаціям RS-232C і V.28. Для живлення мікросхеми потрібно тільки напруга +5В. Напруги +10В і -10В перетворюються з 5В-го за допомогою двох ємнісних перетворювачів напруги.

Мікросхема MAX232 швидко стала індустріальним стандартом. Багато розробники використовують її, незважаючи на те, що параметри мікросхем з однополярним живленням значно покращилися з часом.

Конфігурація висновків MAX232: представлена на рисунку 2.1.

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			18



CAPACITANCE ( $\mu\text{F}$ )					
DEVICE	C1	C2	C3	C4	C5
MAX220	4.7	4.7	10	10	4.7
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

Рисунок 2.1 - Структурна схема MAX232A

На структурній схемі MAX232A зображені подвоювач напруги і інвертор напруги + 10В в 10В. Ці напруги використовуються для формування сигналів відповідних RS-232. MAX232A дозволяє підключити два послідовних порту.

До початку 1960-х в Телепринтери для зв'язку на великі відстані застосовувалася струмова петля 60мА. У 1962 була представлена модель 33 телетайпа з 20мА струмового петлею. Після цього цей інтерфейс став широко використовуватися. Протягом еволюції інтерфейсу 20мА струмова петля застосовувалася багато в якому обладнанні. Схема наведена на рисунку 2.2. Цей інтерфейс став популярним через його низьку ціну при використанні на великих відстанях, а також високою завадостійкістю передачі даних.

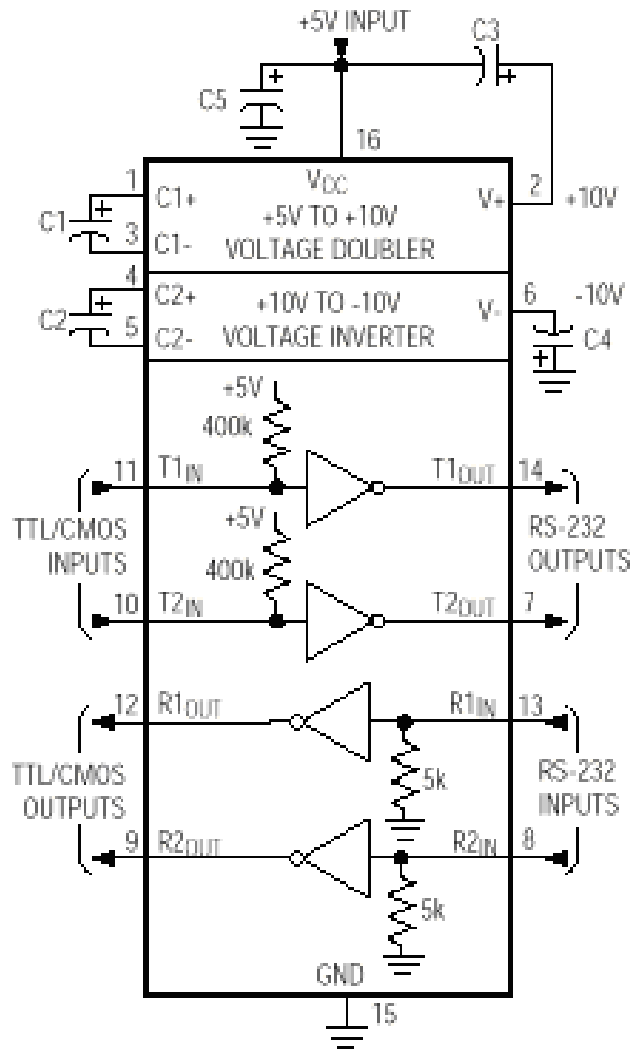


Рисунок 2.2 - Струмова петля

В інтерфейсі зі струмовою петлею електричним сигналом є струм, а не напруга. Струмова петля може працювати в дуплексному, напівдуплексному режимі, а також в активному або пасивному режимі.

Цей стандарт дозволяє передавати данні на відстані до 600 м із швидкістю до 19.2 кБод.

Основні особливості:

- велика дальність ніж у RS-232
- стійкість передачі даних
- відстані до 600 м
- швидкість передачі до 19.2 кБод

Одночасна двунправленна передача даних можлива за цією схемою. Для цього режиму необхідні два генератора струму 20мА. Схема наведена на рисунку 2.3. Наприклад, карта ІВМ адаптера послідовного інтерфейсу має в своєму складі тільки один генератор струму. В цьому випадку для створення повного з'єднання другий пристрій має мати генератор струму для створення другої струмової петлі.

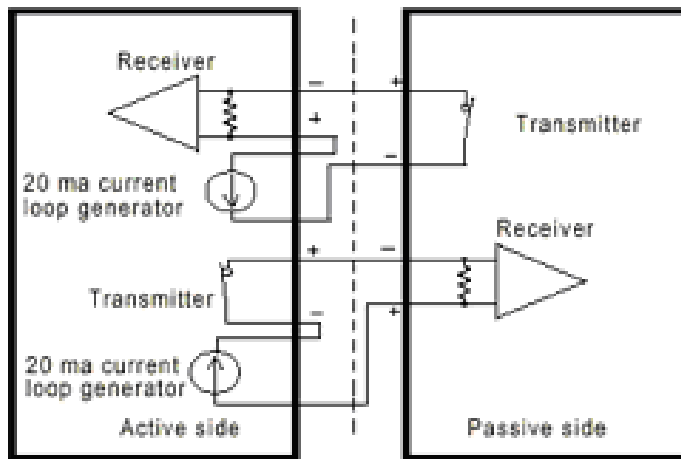


Рисунок 2.3 - Повнодуплексна схема 20 мА

Основними елементами 20 мА струмової петлі є джерело струму, струмовий ключ і струмовий детектор. Передавач - це струмовий ключ, а приймач - детектор струму. Схема, яка містить джерело струму називається активною стороною, інші елементи інтерфейсу – пасивною (рисунки 2.4). У симплексній схемі передавачі та приймачі розташовуються послідовно в одній струмовій петлі. При роботі одного передавача обидва приймачі приймають дані.

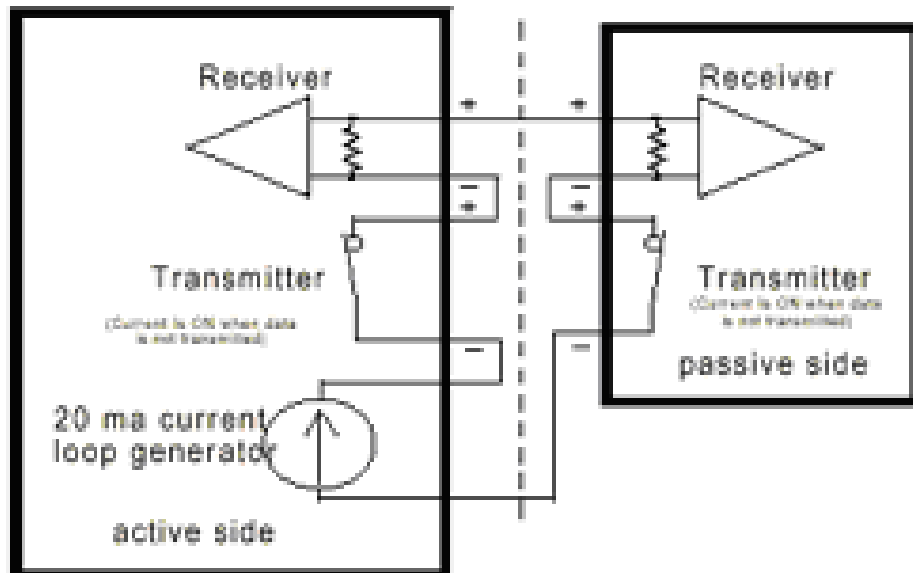


Рисунок 2.4 - Симплексна схема 20 мА

На рис нижче представлені рівні сигналів інтерфейсу RS-232 і їх відповідність з інтерфейсом струмового петлі 20 мА. Для струмового петлі наявність струму відповідає пасивному стану (відсутність передачі даних, рисунок 2.5).

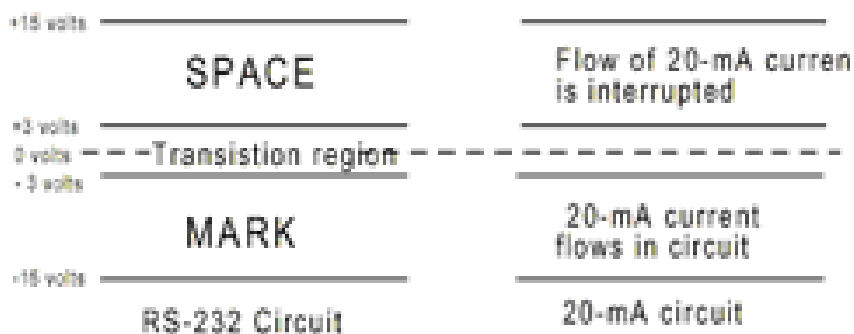


Рисунок 2.5 - Порівняння рівнів RS-232 і 20 мА струмового петлі

Ця схема згадується тут тому, що іноді її плутають з 20мА струмового петлюю. Призначення даної схеми - передача сигналу від віддаленого аналогового датчика через струмового сигнал. Для передачі сигналу потрібно тільки два дроти і джерело живлення датчика. Для живлення датчика використовується джерело напруги 24В. Віддалений датчик змінює струм в

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			22

петлі відповідно до виміряним параметром. На послідовному резистори RL цей струм перетвориться в напругу, яке далі може бути оброблено (рисунок 2.6).

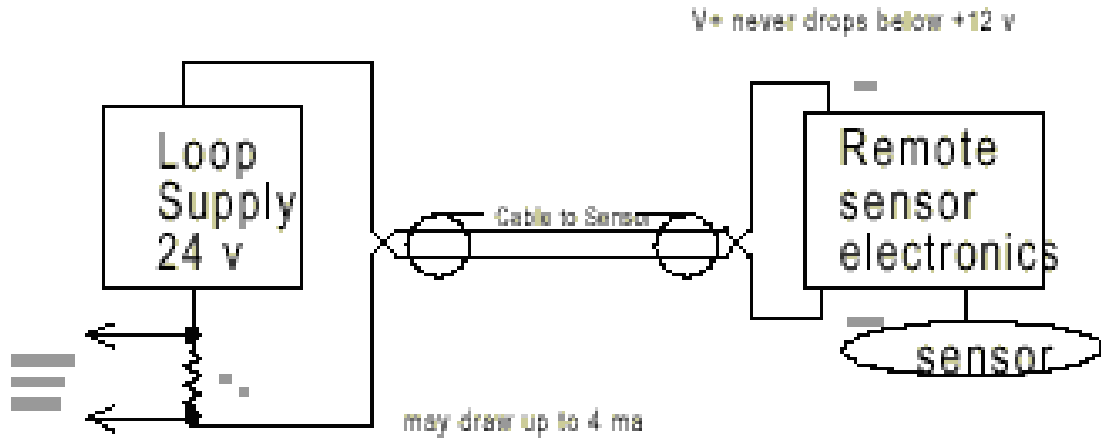


Рисунок 2.6 - Схема перетворювача аналогової струмової петлі 4 - 20 мА

Це ще один приклад комбінування аналогової і цифрової струмового петлі схеми 4 - 20 мА. Для цієї струмового петлі застосовується комунікаційний протокол HART®. HART® протокол використовується для інтелектуальних віддалених перетворювачів, сумісних з аналогової струмового петлею 4-20 мА, а також мають цифровий обмін на ті ж проводам. Це здійснюється за рахунок застосування двотонального частотного сигналу (FSK) сигналу з рівнями 4-20 мА.

Утиліта Hercules SETUP - корисний термінал послідовного порту (RS-232 або RS-485), протоколів UDP / IP і TCP / IP (клієнт або сервер). Може використовуватися з оригінальними пристроями Ethernet (конвертери Serial / Ethernet, RS-232 / Ethernet буфери або контролери введення / виведення) for the UDP Setup (рисунок 2.7.). Утиліта була створена для власних потреб, але зараз вона включає в себе багато додаткових функцій і поширюється Freeware.

Основні корисні частини:

- Термінал послідовного порту - підтримка COM5 і вище;

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			23

- TCP / IP клієнтський термінал;
- TCP / IP серверний термінал;
- UDP термінал.

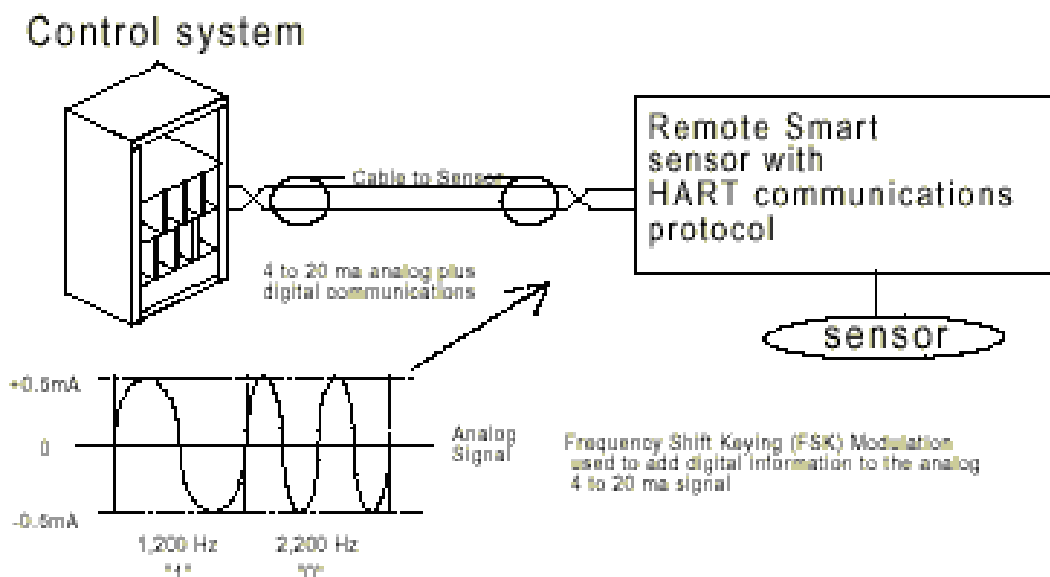


Рисунок 2.7 - Схема аналогової струмової петлі 4 до 20 мА з цифровою передачею даних по HART протоколу

Основні особливості:

- Не потребує інсталяції, тільки один .EXE файл.
- Працює з віртуальним послідовним портом, (COM12 наприклад)
- Можна використовувати прості Макро функції, що включає посилку HEX команд. Макро функції зберігаються в регістрах, і Hercules запам'ятовує їх.
- Термінал послідовного порту показує стану і може керувати сигналами модему (CTS, RTS, DTR, DSR, RI, CD)
- Можна пересилати файли і зберігати отримані дані в LOG файлі.
- Підтримка TEA - безпечну TCP / IP, щоб отримати клієнта і сервера TCP / IP, тестових режимів ..
- Підтримка передачі даних TCP / IP в TCP / IP сервері або клієнта TCP / IP.

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			24



- Підтримка передачі даних UDP / IP в UDP / IP терміналі
- Підтримка мережевого віртуального терміналу NVT (Network Virtual Terminal) в тестовому режимі.
- Застосування Telnet додатково з NVT дозволяє конфігурувати послідовний порт (RFC2217), проводити ідентифікацію пристрою, підтвердження передачі даних і ін.
- Ця утиліта поширюється FREEWARE, можна використовувати її і поширювати без всяких обмежень.

Terminal - це простий емулятор терміналу послідовного порту (COM). Може застосовуватися для комунікаціїс різними пристроями, такими як модеми, роутери, GSM телефони (рисунок 2.8). Дуже корисна утиліта для налагодження додатків для з'єднань по послідовному каналу.

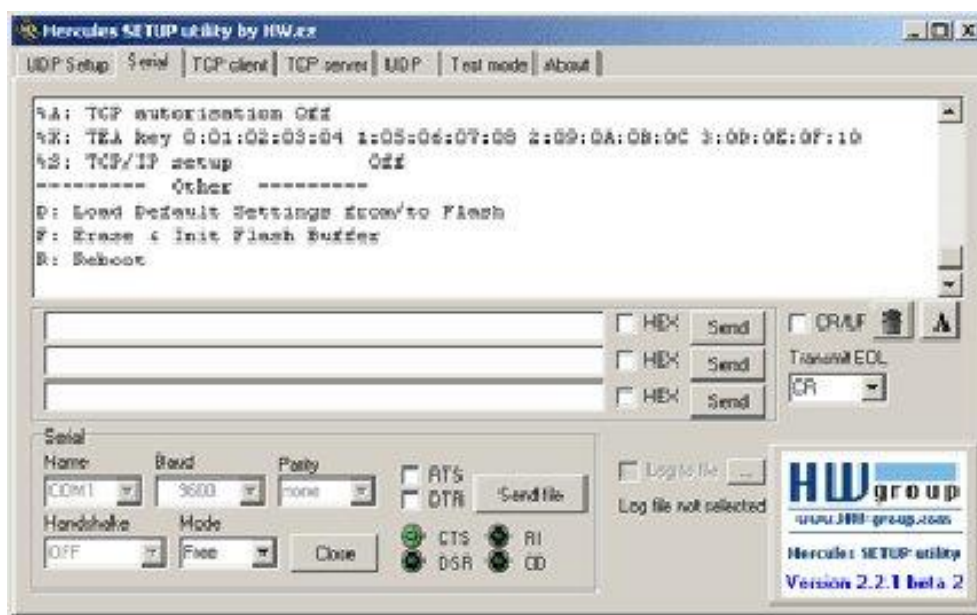


Рисунок 2.8 - Головне вікно Terminal

Основні особливості:

- маленький розмір файлу small .exe 246k;
- проста посилка файлу;
- лічильник символів;
- до 6 COM портів;

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			25

- швидкість обміну до 256Кбіт / с;
- запис в log файл (hex & string);
- передача макросів.

Tera Term (Pro) - вільно поширюваний емулятор терміналадля MS-Windows, як показано на рисунку 2.9.

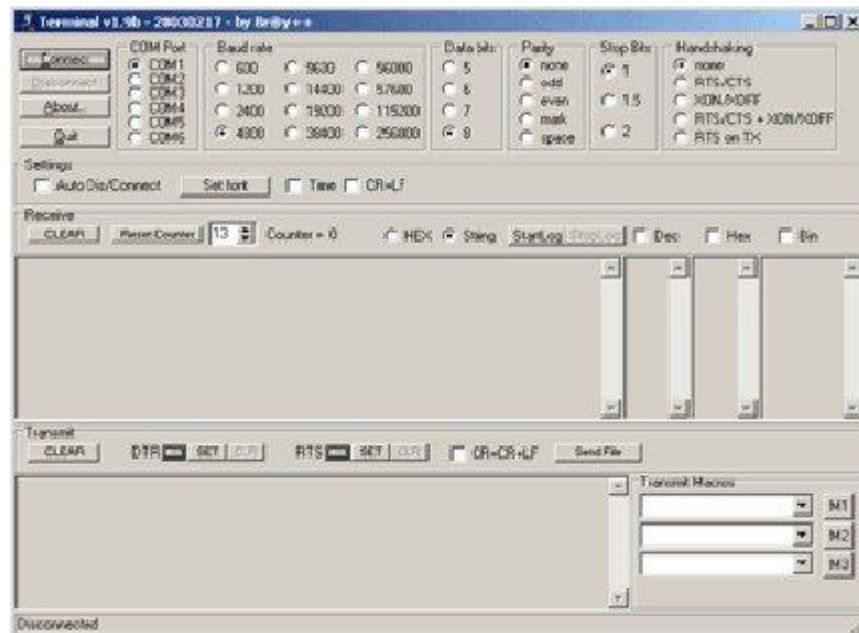


Рисунок 2.9 – Головне вікно Tera Term

Основні особливості:

- Підтримка портів вище COM4.
- Швидкість обміну 14400.
- Гарячі клавіші користувача "EditCLS", "EditCLB", "ControlOpenTEK" і "ControlCloseTEK".
- Дії "clearscreen", "code2str", "enablekeyb", "filemarkptr", "fileseekback", "filestrseek2", "findclose", "findfirst", "findnext", "getdir", "kmtfinish", "kmtget", "makepath", "sendkcode", "setdir", "setexitcode", "str2code" and "testlink".
- Макрооператор "%".
- Макро: коментар, який можна включити в будь-який рядок.
- Pass-through printing directly to a port

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			26

- Спеціальні опції в файлі установки "AutoFileRename", "BackWrap", "Beep", "EnableStatusLine", "MaxComPort", "PassThruDelay", "PassThruPort", "PrnConvFF", "RussPrint" і "TitleFormat".
- Нове діалогове вікно [Setup] TCP / IP.
- Лист історії хостів.
- Використання кодувань російської мови (CP 866 і ISO 8859-5).
- Нові іконки.
- Можливість виділення тексту.
- Виконання команд меню за допомогою гарячих клавіш, які визначаються користувачем.
- Tera Term розширений інтерфейс (рисунок 2.10).

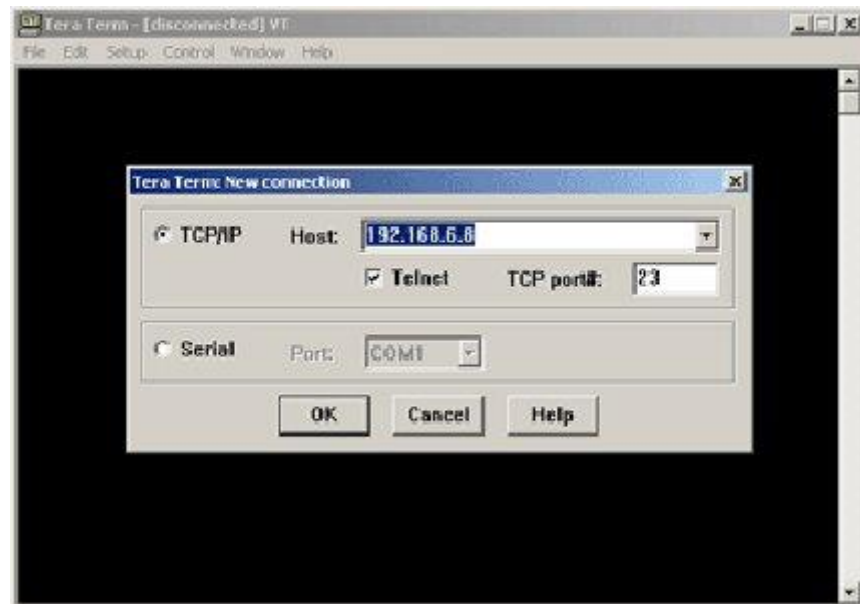


Рисунок 2.10 - Term розширений інтерфейс.

Підтримує емуляцію VT100, telnet з'єднання, з'єднання по послідовному порту і т.д.

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			27

### 3 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ

#### 3.1 Розробка інтерфейсу програмного засобу

При проектуванні програмного засобу було розроблено зручний графічний інтерфейс, що демонструє роботу програми та надає можливість подати сигнал на будь-який з пінів інтерфесу. Головне вікно додатку на етапі розробки приведенне на рисунку 3.1.

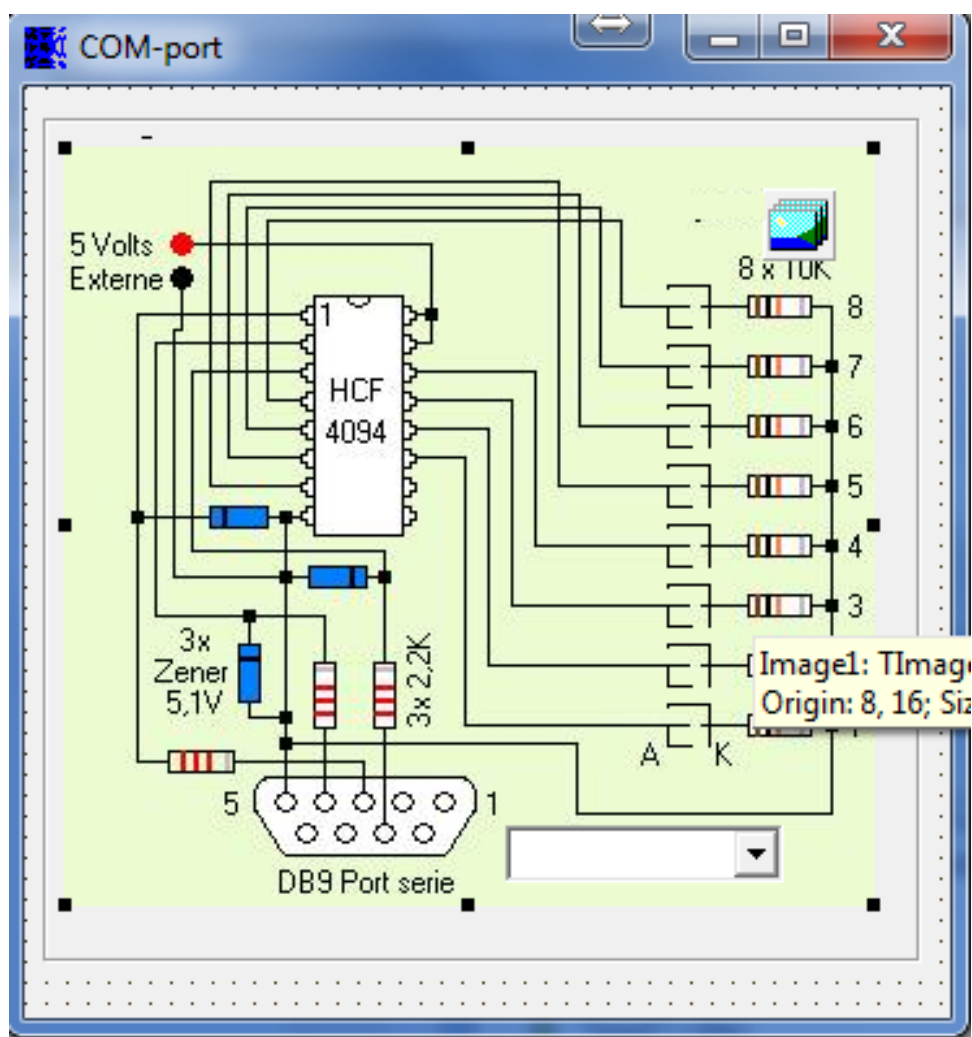


Рисунок 3.1 - Головне вікно додатку на етапі розробки

При проектуванні графічного інтерфейсу було використано компоненти наступних класів:

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			28

- TGroupBox;
- TImageList;
- TComboBox;
- TImage;

Основні функції, що були використані наведені в модулях: Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, PORTUtil, ImgList, ExtCtrls, jpeg;

Величезна кількість зовнішніх пристроїв зв'язуються з комп'ютером за допомогою RS-232.

Для використання програмного засобу існує необхідність наявності в комп'ютері com-порта. Існує необхідність передавати і приймати по ньому байти. У загальному випадку постановка задачі буде наступною: необхідно відправляти потрібний рядок і необхідно реагувати на прихід рядки в порт від зовнішнього пристрою.

Приклад містить інтерфейсну форму, а також модуль для роботи з портом. Модуль узятий з реально працюючої програми, але (для спрощення), з нього були викинуті всілякого виду перевірки, залишилась лише структура обробника даних.

При моделюванні роботи пристрою треба використовувати велику кількість перевірок (після кожної API-функції), очищати буфер порту і задавати його ємність і т.д. Але для початку (наприклад, щоб послати що-небудь в модем) необхідно розробити ряд процедур та функцій. Відмінна особливість прикладу - організація роботи прийому даних з порту за подією, якою програма чекає в паралельному потоці.

### 3.2 Розробка функцій програмного засобу

Модуль для роботи з портом містить чотири процедури:

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			29

- PortInit - ініціалізація роботи і запуск потоку прийому даних з порту.
- KillComm - власне знищення потоку прийому і самого порту.
- WriteComm - запис в порт.

Процедура PortInit будується наступним чином.

Для початку необхідно створити порт і отримати його ідентифікаційний номер (хоча, строго кажучи, створити файл і отримати хендл). Зробити це можна однією функцією CreateFile:

```
CommHandle: = CreateFile ( 'COM1', GENERIC_READ or
GENERIC_WRITE, 0, nil,
OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL or
FILE_FLAG_OVERLAPPED, 0);
```

CommHandle - хендл, тобто номер створеного каналу. Тип - THandle. Надалі працюємо в програмі тільки з нею.

Перший параметр 'COM1' - власне ім'я порту. Його відповідно можна міняти (необхідно звернути увагу це не тип string, а тип PChar). Решта установки досить стандартні і міняти їх часто не доведеться.

Тепер необхідно налаштувати параметри порту, а так само маску. Маска - це опис такої події, яку порт буде чекати, і за яким вестиметься обробка подій. В розглянемо окремий випадок - прихід символу "повернення каретки".

SetCommMask (CommHandle, EV\_RXFLAG); - встановлюємо маску EV\_RXFLAG - "обробка за певним символу". Іншими словами, як тільки в порт прийде необхідний символ - то програма почне обробляти цю подію. При відстеженні кількох подій вони задаються через or (логічне або).

Сам символ задаємо в DCB-структурі. DCB- структура - це керуюча структура обраного порту. Ключова необхідність. Необхідно її заповнити. Власне в ній визначаються настройки порту.

GetCommState (CommHandle, DCB); - отримуємо поточний DCB.

DCB.BaudRate: = CBR\_9600; - встановлюємо швидкість роботи.

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
							30
Змн.	Арк.	№ докум.	Підпис	Дата			

DCB.Parity: = NOPARITY; - встановлюємо відсутність перевірки на парність.

DCB.ByteSize: = 8; - 8 біт в переданому байті.

DCB.StopBits: = OneStopBit; - одиночний стоп-біт.

DCB.EvtChar: = chr (13); - задаємо символ для SetCommMask. В даному випадку - повернення каретки.

SetCommState (cId, DCB); - тепер прописуємо виправлене DCB.

Природно це не всі параметри DCB, а тільки найважливіші та часто використовувані. Всю структуру DCB, а так само всі значення параметрів можна отримати за допомогою опису структури.

Тепер займемося організацією прийому інформації з порту. Для того щоб не зациклять програму на постійному зчитуванні порту з одного боку і в той же час завжди бути готовим прийняти інформацію, запускаємо процедуру читання порту в окремому потоці:

```
CommThread: = CreateThread (nil, 0, @ ReadComm, nil, 0, ThreadID);
```

Мінлива CommThread - хендл, але вже на потік.

ReadComm - власне процедура, за якою і проводиться обробка. "@" означає, що передаємо не ім'я процедури, а її адресу. ThreadID - ідентифікатор потоку.

Даний рядок запускає процедуру ReadComm паралельно із основною програмою.

Процедура ReadComm дозволяє зчитувати з порта дані. Тепер розглянемо процедуру ReadComm. Це звичайна процедура, входячи складу модуля. Весь її вміст зациклюється за допомогою while true do.

Запускаємо процедуру виключно в окремому потоці, так що б не відбулось зависання програмного засобу.

Процедура містить цикл і в циклі очікується подія:

```
WaitCommEvent (CommHandle, TransMask, @ Ovr);
```

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			31

Потік зупиняється і чекає якої-небудь події прописаного в SetCommMask (в процедурі PortInit). Як тільки подія відбулася програма продовжує виконання.

(TransMask and EV\_RXFLAG) = EV\_RXFLAG - цим виразом перевіряємо, а чи подія відбулася, що необхідно.

Необхідно зазначити, що в SetCommMask можна поставити за допомогою оператора or кілька подій. Тоді в обробнику після очікування треба відповідно передбачити і кілька if then. Що б прописати реакцію на кожну подію.

ClearCommError (CommHandle, Errs, @ Stat); - всупереч назві, ця функція очищає не помилки, а власне факт приходу події. Без неї RXFLAG так і залишиться висіти. Можна спробувати вгадати, що буде на наступному циклі прийому.

Потім йде власне прийом Kols: = Stat.cbInQue; - беремо кількість байт в буфері порту.

ReadFile (CommHandle, Resive, Kols, Kols, @ Ovr); - зчитуємо все в масив Resive.

Далі все отримане в Resive треба обробити.

Не треба з потоку звертатися до візуальних компонентів (наприклад, до Panel1 J). Краще рішення - щоб юніт по роботі з портом взагалі не бачив головний юніт і головну форму. Виведення даних на екран або на обробку можна зробити або за таймером або, наприклад, пославши призначене для користувача повідомлення (message), ну а в обробнику організувати виведення прийнятої інформації на екран.

Процедура WriteComm.

Власне вона робить запис в порт. У наведеному коді - один байт.

KolByte: = 1;

Transmit [0]: = chr (A);

WriteFile (CommHandle, Transmit, KolByte, KolByte, @ Ovr);

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			32



Змінна Kolbyte - це і є власне кількість байт, що посилаються. Послати рядок ще простіше. Необхідно заповнити KolByte кількістю символів в рядку, а замість масиву Transmite використовуємо строковий тип, тільки не в паскалівському форматі, а в PChar (він має інший формат рядка, або null-terminated string).

Треба відзначити, що у всіх функціях API використовується саме цей строковий стандарт, очевидно через написання самої операційки на Сі.

Процедура KillComm знищує створений об'єкт.

Закінчивши використання порту необхідно звільнити оперативну пам'ять та знищити використані дескриптори. TerminateThread (CommThread, 0); - "знищення" паралельного потоку прийому

CloseHandle (CommHandle); - "знищення" власне файлу-порту.

var

DCB: TDCB;

hPort: THandle;

begin

// 1. Відкриваємо файл

```
hPort = CreateFile (PChar ( 'COM' + IntToStr (<номер порту>)),
    GENERIC_READ + GENERIC_WRITE,
    0, nil,
    OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0);
```

// 2. Контроль помилок

```
if hPort = INVALID_HANDLE_VALUE then begin
```

```
// Виявлена помилка, порт відкрити не вдалося
```

```
exit;
```

```
end;
```

// 3. Читання поточних налаштувань порту

```
if GetCommState (hPort, DCB) then;
```

// 4. Налаштування:

// Швидкість обміну

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			33

```

DCB.BaudRate: = [швидкість обміну];
// Число біт на символ
DCB.ByteSize: = [розмір "байта" при обміні по СОМ порту - зазвичай
8 біт];
// Стоп-біти
DCB.StopBits: = [константа, яка визначає кількість стопбітів];
// Парність
DCB.Parity: = [константа, яка визначає режим контролю парності];
DCB.Flags: = 20625;
// 5. Передача налаштувань
if not SetCommState (hPort, DCB) then {помилка настройки порту};
// 6. Налаштування буферів порту (черг введення і виведення)
if not SetupComm (hPort, 16, 16) then {помилка настройки буферів};
// 7. Скидання буферів і черг
if PurgeComm (hPort, PURGE_TXABORT or PURGE_RXABORT or
PURGE_TXCLEAR or PURGE_RXCLEAR) then;
.....
// 8. Закриття порту
CloseHandle (hPort);
end;

```

Модем - по суті, той же порт. Зв'язок реалізується так само як і в попередньому коді. Використовуємо функцію GetModemStatus, замість EscapeCommFunction.

### 3.3 Тестування та верифікація програмного засобу

Тестування програмного забезпечення - перевірка відповідності між реальним і очікуваною поведінкою програми, що здійснюється на кінцевому

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			34

наборі тестів, обраному певним чином. У більш широкому сенсі, тестування - це одна з технік контролю якості, що включає в себе активності з планування робіт (Test Management), проектування тестів (Test Design), виконання тестування (Test Execution) і аналізу отриманих результатів (Test Analysis).

Для тестування проведемо тестовий запуск та виконання функціональних можливостей. Після запуску програми на екрані буде відображена форма, що на рисунку 3.2.

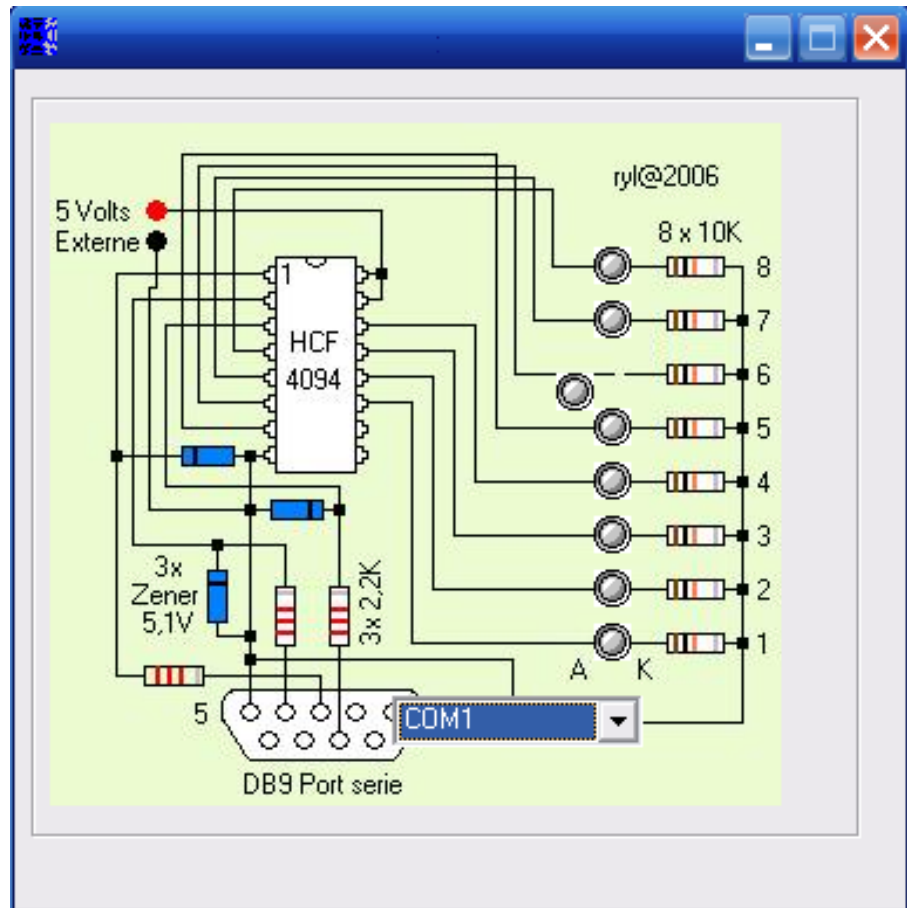


Рисунок 3.2 – Головне вікно додатку після запуску. Етап тестування.

Якість програмного забезпечення (Software Quality) - це сукупність характеристик програмного забезпечення, що відносяться до його здатності задовольняти встановлені і передбачувані потреби.

Верифікація (verification) - це процес оцінки системи або її компонентів з метою визначення чи задовольняють результати поточного етапу розробки умов, сформованим на початку цього етапу (IEEE). Тобто чи виконуються

цілі, терміни, завдання по розробці проекту, визначені на початку поточної фази.

Валідація (validation) - це визначення відповідності розробляється ПО очікуванням і потребам користувача, вимогам до системи (BS7925-1).

Також можна зустріти іншу інтерпретацію: процес оцінки відповідності продукту явним вимогам (специфікаціям) і є верифікація (verification), в той же час оцінка відповідності продукту очікуванням і вимогам користувачів - є валідація (validation).

При виконанні перевірки функціональних можливостей розробленого програмного засобу було отримано результати виконання тестового плану.

Було виявлено деякі проблеми сумісності при виконанні програмного продукту на Windows 10, причиною яких є відсутність певних API бібліотек.

Результат замикання одного піну зображено на рисунку 3.3.

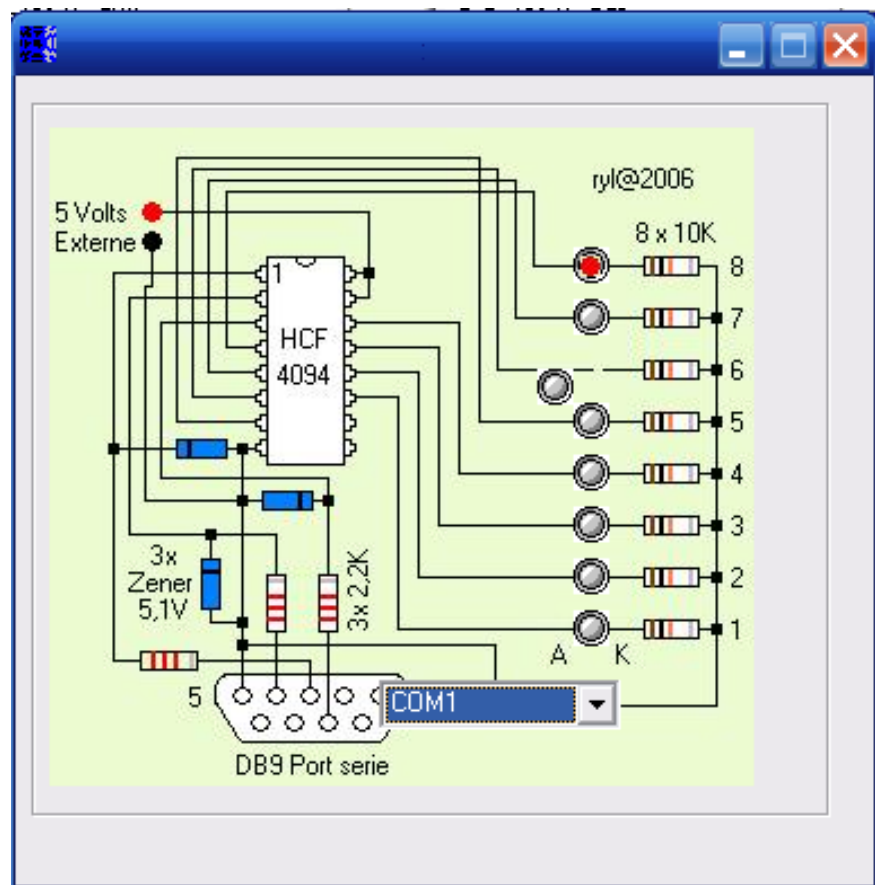


Рисунок 3.3 - Головне вікно додатку з встановленим сигналом. Етап тестування.

Тест план (Test Plan) - це документ, що описує весь обсяг робіт з тестування, починаючи з опису об'єкта, стратегії, розкладу, критеріїв початку і закінчення тестування, до необхідного в процесі роботи обладнання, спеціальних знань, а також оцінки ризиків з варіантами їх вирішення. З гідно тестового плану було перевірено використання кожного піну. На рисунку 3.4 наведено приклад замикання двох пінів.

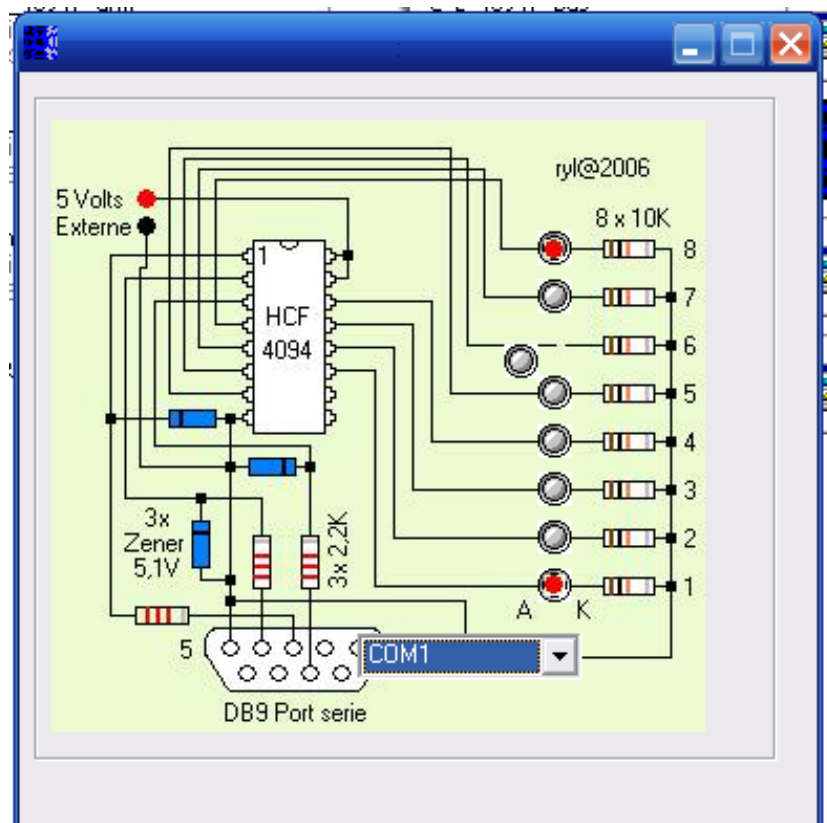


Рисунок 3.4 - Головне вікно додатку з двома встановленими сигналами.

#### Етап тестування.

У стандарті IEEE 829 перераховані пункти, з яких повинен (нехай - може) складатися тест-план:

Тест дизайн - це етап процесу тестування ПО, на якому проектується і створюються тестові випадки (тест кейси), відповідно до визначених раніше критеріями якості та цілями тестування.

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			37

Тестування проводилось на комп'ютері: Intel Pentium N4200 (1.1 - 2.5 ГГц) / RAM 4 ГБ / HDD 1 ТБ / Intel HD Graphics 505 / DVD Super Multi / LAN / Wi-Fi / Bluetooth.

Цілі тестування:

- підвищити ймовірність того, що додаток, призначене для тестування, буде працювати правильно при будь-яких обставинах.
- підвищити ймовірність того, що додаток, призначене для тестування, буде відповідати всім описаним вимогам.
- надання актуальної інформації про стан продукту на даний момент.

Етапи тестування:

- Аналіз продукту.
- Робота з вимогами.
- Розробка стратегії тестування і планування процедур контролю якості.
- Створення тестової документації.
- Тестування прототипу.
- Основне тестування.
- Стабілізація.
- Експлуатація.

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			38

## ВИСНОВКИ

Тестування проводилось на комп'ютері: Intel Pentium N4200 (1.1 - 2.5 ГГц) / RAM 4 ГБ / HDD 1 ТБ / Intel HD Graphics 505 / DVD Super Multi / LAN / Wi-Fi / Bluetooth.

Цілі тестування:

- підвищити ймовірність того, що додаток, призначене для тестування, буде працювати правильно при будь-яких обставинах.
- підвищити ймовірність того, що додаток, призначене для тестування, буде відповідати всім описаним вимогам.
- надання актуальної інформації про стан продукту на даний момент.

					ДП.КСМ. .00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гольштейн Б.С. Учебник для вузов: «Системы коммутации» / Б.С. Гольштейн // С-Пб.:БХВ-Санкт-Петербург, 2003.-318 с.:ил.
2. Руководство по Cisco IOS [Текст]. - СПб.: Питер, М.: Издательство «Русская Редакция», 2008. -784 с
3. Официальный сайт производителя оборудования Cisco Systems [Электронный ресурс] / Режим доступа – <http://www.cisco.com>.
4. Хьюкаби, Д. Руководство по конфигурированию коммутаторов Catalyst.: Пер. с англ. [Текст]/ Д. Хьюкаби, С. Мак-Квери– М.: Издательский дом «Вильямс» – 2004. – 560 с.
5. Основы организации сетей Cisco, том 1 [Текст].: Пер. с англ. - М.: Издательский дом «Вильямс», 2002. - 512 с.
6. Портнов, Э.Л. Оптические кабели связи [Текст] / Э.Л. Портнов– М. «Информсвязь», 2000 – 112 с.
7. Программа сетевой академии Cisco CCNA 3 и 4. Вспомогательное руководство, 3-е изд., с испр. [Текст]: Пер. с англ. – М.: ООО «И. Д. Вильямс», 2007. – 994.
8. Величко В.В. Телекоммуникационные системы и сети: Учебное пособие в 3-х томах. Том 3. – Мультисервисные сети [Текст] / В.В. Величко, Е.А. Субботин, В.П. Шувалов, А.Ф. Ярославцев; под ред. профессора В.П. Шувалова. – М.: Горячая линия – Телеком, 2005. – 592 с.
9. Росляков А.В. IP-телефония / А.В. Росляков, М.Ю. Самсонов, И.В. Шibaева– М.: Эко-Трендз, 2003. – 252 с.
10. Гольштейн Б.С. IP-телефония / Б.С. Гольштейн, А.В. Пинчук, А. Л. Суховицкий // М.: Радио связь, 2001.- 366с.:ил.
11. Гольштейн Б.С. Справочник по телекоммуникационным протоколам: «Протокол SIP» / Б.С. Гольштейн, А. А. Зарубин, В. В. Саморезов // С-Пб.:БХВ-Санкт-Петербург, 2005.-456с.:ил.

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			40



12. Принципы маршрутизации в Internet, 2-е издание [Текст]: Пер. с англ. М.: Издательский дом "Вильямс", 2001. - 448 с.: ил.
13. Руководство по Cisco IOS [Текст]. - СПб.: Питер, М.: Издательство «Русская Редакция», 2008. -784 с
14. Официальный сайт производителя оборудования Cisco Systems [Электронный ресурс] / Режим доступа – <http://www.cisco.com>.
15. Хьюкаби, Д. Руководство по конфигурированию коммутаторов Catalyst.: Пер. с англ. [Текст]/ Д. Хьюкаби, С. Мак-Квери– М.: Издательский дом «Вильямс» – 2004. – 560 с.
16. Транспортные сети и системы электросвязи. Системы мультиплексирования: Учебник для студентов ВУЗов по специальности «Телекоммуникации» [Текст] / Под ред. В.К. Стеклова. – К.; 2003 – 352 с.
17. Дональд, Дж. Стерлинг. Техническое руководство по волоконной оптике [Текст] / Дональд Дж. Стерлинг., пер. Московченко А. – Издательство «ЛОРИ» – 1998.
18. Основы организации сетей Cisco, том 1 [Текст].: Пер. с англ. - М.: Издательский дом «Вильямс», 2002. - 512 с.
19. Портнов, Э.Л. Оптические кабели связи [Текст] / Э.Л. Портнов– М. «Информсвязь», 2000 – 112 с.
20. Программа сетевой академии Cisco CCNA 3 и 4. Вспомогательное руководство, 3-е изд., с испр. [Текст]: Пер. с англ. – М.: ООО «И. Д. Вильямс», 2007. – 994.
21. Слепов Н.Н. Оптоволоконные системы дальней связи. Перспективы развития [Текст] // Н.Н. Слепов. - Электроника: НТБ – 2005. – Вып.6.
22. Величко В.В. Телекоммуникационные системы и сети: Учебное пособие в 3-х томах. Том 3. – Мультисервисные сети [Текст] / В.В. Величко, Е.А. Субботин, В.П. Шувалов, А.Ф. Ярославцев; под ред. профессора В.П. Шувалова. – М.: Горячая линия – Телеком, 2005. – 592 с.

					ДП.КСМ.	.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			41

23. Методичні рекомендації до виконання дипломного проекту з освітньо-кваліфікаційного рівня “Бакалавр” напряму підготовки 6.050102 «Комп’ютерна інженерія» фахового спрямування «Комп’ютерні системи та мережі» / О.М. Березький, Л.О.Дубчак, Р.Б. Трембач, Г.М. Мельник, Ю.М. Батько, С.В. Івасьєв / Під ред. О.М. Березького. -Тернопіль: ТНЕУ, 2016. – 60 с.

24. Методичні вказівки до написання техніко-економічного розділу дипломних проектів освітньо-кваліфікаційного рівня «бакалавр» напряму підготовки 6.050102 комп’ютерна інженерія/ І.Р. Паздрій – Тернопіль: ТНЕУ, 2014. – 37 с.

					ДП.КСМ. .00.00.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		