

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-05

Дипломний проект

**здобувача освіти денної форми навчання
РП.05.08.000.ДП**

***ДЕМЧЕНКА
ВЛАДИСЛАВА СЕРГІЙОВИЧА***

**м. Одеса
2022 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: **121 «Інженерія програмного забезпечення»**

Освітня програма: **«Розробка програмного забезпечення»**

Група: **4РП-05**

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

Розробка програмного забезпечення
USB-частотоміру для технічних вимірювань

Проектний матеріал складається з пояснювальної записки на 81 сторінках та графічного (презентаційного) матеріалу на 16 аркушах (слайдах).

Дипломник _____ (Демченко В.С.)

Керівник _____ (Кривченко А.А.)

Консультанти:

з економічної частини _____ (Копайгородська Т.Г.)

з охорони праці _____ (Чорновол Н.І.)

з дотримання вимог ЄСКД _____ (Петрашова В.І.)

старший консультант _____ (Скорнякова О.В.)

До захисту допущений

Голова циклової комісії _____ (Скорнякова О.В.)

Завідувач відділення _____ (Суліма Ю.Ю.)

Захист « » _____ 2022 р.

Протокол ДКК № _____

Оцінка ДКК _____

Секретар ДКК _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення _____ комп'ютерних систем _____ Комісія _____ КТ і П
Спеціальність _____ 121 «Інженерія програмного забезпечення» _____

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР _____
Беркань І.В. _____

“ _____ ” _____ 2022 р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувачеві (здобувачці) освіти _____ Демченку Владиславу Сергійовичу _____

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) _____ Розробка програмного забезпечення USB-частотоміру для
технічних вимірювань _____

затверджена наказом по коледжу від “ _____ ” _____ 202 _____ р. № _____

2. Термін здачі закінченого проекту (роботи) _____

3. Вихідні дані до проекту (роботи) _____ 1. Технічні характеристики і специфікації сучасних частотомірів

2. Зв'язок з ПК через інтерфейс USB; 3. Стандартна схема перетворювача ADC100;

4. Використати у якості АЦП ІМС TLC1549; 5. Передбачити гальванічну розв'язку ланцюгів

на оптронах; 6. Передбачити підсилення вхідних сигналів не менш ніж в 100 разів;

7. Використати в схемі підсилювача ІМС LM358; 9. Розробку програмного забезпечення віртуального частотоміру виконувати у ICP RAD Studio або Visual Studio під ОС сімейства Windows

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

1. Аналіз існуючих засобів для вимірювання частоти;

2. Вибір і аналіз схем аналого-цифрового перетворення;

3. Розробка блоку сполучення з ПК для USB-частотоміру;

4. Розробка програмного забезпечення USB-частотоміру;

5. Економічна частина; 6. Охорона праці

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

1. Структурна схема USB-частотоміру на базі ПК

2. Блоки вхідного підсилювача та АЦП USB-частотоміру

3. БСА отримання даних з USB-порту

4. Приклади роботи програмного забезпечення USB-частотоміру

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1-4	Кривченко А.А.		
5. Економічна частина	Копайгородська Т.Г.		
6. Охорона праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		

7. Дата видачі завдання _____ Кривченко Анастасія Анатоліївна

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1.	Вступ. Постановка задачі проектування		
2.	Огляд і аналіз існуючих промислових частотомірів		
3.	Аналіз технічного завдання		
4.	Вибір і аналіз схем аналого-цифрового перетворення		
5.	Аналіз схемотехніки сучасних АЦП		
6.	Аналіз необхідності вживання в системі вимірювання АЦП з розрядністю, більшою 12		
7.	Вибір джерела опорної напруги		
8.	Розробка структурної схеми приладу		
9.	Розробка принципової електричної схеми підсилювача та блоку АЦП для послідовного порту		
10.	Вибір і обґрунтування складу засобів для розробки		
11.	Розробка алгоритмів, проектування і опис інтерфейсу, складання програмного забезпечення приладу		
12.	Техніко-економічне обґрунтування розробки		
13.	Розробка заходів техніки безпеки		

Дипломник

_____ (підпис)

Керівник

_____ (підпис)

ЗМІСТ

Вступ.....	6
1 Аналіз існуючих засобів для вимірювання частоти.....	9
1.1 Використання електронно-рахункових частотомірів.....	9
1.1.1 Класифікація і принцип дії частотомірів.....	9
1.1.2 Призначення і використання апаратного частотоміра ЧЗ-67.....	12
1.2 Віртуальні вимірювальні прилади на основі ПК.....	14
1.2.1 Інтерфейсні плати внутрішнього виконання.....	17
1.2.2 Зовнішні інтерфейсні пристрої.....	17
1.2.3 Віртуальний частотомір VFM.....	18
1.3 Висновки за розділом.....	20
2 Вибір і аналіз схем аналого-цифрового перетворення.....	21
2.1 Схемотехніка сучасних АЦП.....	21
2.2 8-розрядний АЦП з послідовним інтерфейсом.....	26
2.3 10-розрядний АЦП з послідовним інтерфейсом.....	28
2.4 12-розрядний АЦП з послідовним інтерфейсом.....	30
2.5 Аналіз необхідності вживання АЦП з розрядністю, більшою 12.....	32
2.6 Висновки за розділом.....	33
3 Розробка блоку сполучення з ПК для USB-частотоміру.....	34
3.1 Розробка принципової електричної схеми блоку АЦП.....	35
3.2 Розробка принципової електричної схеми вхідного підсилювача.....	41
4 Розробка програмного забезпечення USB-частотоміру.....	45
4.1 Склад технічних і програмних засобів для розробки.....	45
4.1.1 Технічні засоби розробки.....	45
4.1.2 Середовище розробки ПЗ.....	46
4.2 Створення інтерфейсу користувача ПЗ USB-частотоміру	51
4.3 Розробка алгоритмів і створення обробників подій.....	52
4.3.1 Реалізація процедури вимірювання вхідних сигналів.....	55
4.3.2 Реалізація функції знаходження частоти.....	58

4.3.3	Розробка і опис цифрової шкали.....	62
4.3.4	Реалізація керування протоколом та запуск програми.....	65
4.4	Опис інтерфейсу і роботи ПЗ USB-частотоміру	65
4.5	Тестування розробленого програмного забезпечення.....	68
5	Економічна частина.....	70
5.1	Резюме.....	70
5.2	Визначення трудомісткості розробки програмного забезпечення.....	70
5.3	Розрахунок ціни програмного продукту.....	73
6	Охорона праці.....	75
6.1	Вступ.....	75
6.2	Коротка характеристика і основні вимоги безпеки до мікроклімату виробничих приміщень, освітлення, шуму, вібрації, ультразвуку, інфразвуку, виробничих випромінювань, небезпека ураження електричним струмом.....	75
6.2.1	Мікроклімат виробничих приміщень.....	76
6.2.2	Освітлення виробничих приміщень.....	77
6.2.3	Заходи щодо захисту від дії шуму та вібрації.....	77
6.2.4	Безпека праці.....	78
6.2.5	Електробезпека.....	78
6.3	Пожежна безпека.....	79
	Висновки.....	80
	Перелік використаних джерел.....	80
	Додаток.....	80

ВСТУП

Вимірювання частоти електричного сигналу широко застосовується у радіотехніці, зв'язку, навігації, телекомунікаціях тощо. Вимірювання частоти це одне з найбільш точних вимірювань на даний момент, адже існують високоточні стандарти частоти. Частотні сигнали володіють доброю завадостійкістю і вимірювання виконуються з отриманням результату в дискретній формі. Це сприяє широкому застосуванню частотних сигналів. Задача швидкого і точного вимірювання частоти особливо актуальна при побудові інформаційно-вимірювальних систем на основі датчиків з частотними виходами, при використанні уніфікованих перетворювачів аналогових величин в частоту, а також в приладобудівній галузі.

Найбільше розповсюдження при побудові сучасних частотомірів отримали методи вимірювання частоти прямого відліку. Існує два режими роботи таких частотомірів: в першому режимі за ціле число періодів зразкової частоти підраховується число періодів вимірюваної частоти; в другому режимі за ціле число періодів частоти підраховується число періодів зразкової частоти. Абсолютна максимальна методична похибка дискретизації по модулю u в першому випадку рівна періоду вимірюваної величини, в другому випадку – зразкової величини. З метою зменшення цієї похибки розроблено багато методів і прийомів. Наприклад, метод синхронізації початку зразкового інтервалу часу з початком першого періоду вимірюваної частоти; однократна чи багаторазова розтяжка інтервалу часу і інші. Проте, до недоліків подібних методів і прийомів можна віднести наступне: збільшення часу вимірювання частоти; виконання додаткових аналогових частин схеми; використання елементів схем з відносно не високою стабільністю (наприклад, генераторів ударного збудження) і інші.

У наш час продовжуються розробки методів вимірювання частоти, які дозволяють більш точно і за менший час вимірювати частоту. Після цифрового розповсюдження мікропроцесорної техніки і швидкодіючих АЦП було

					РП 05.08.000.00 ДП ПЗ	Арх.
Зам.	Арх.	Не введено	Голов.	Датум		6

розроблено ряд способів вимірювання частоти, який оснований на швидкому перетворенні Фур'є, обчислення частоти на основі миттєвих відліків, уточнення за допомогою поточного усереднення, уточнення шляхом вагової функції і інші. Проте запропоновані методи складні в реалізації.

Треба зазначити, що не дивлячись на стрімкий технічний прогрес, велика кількість існуючих цифрових частотомірів не влаштовують користувачів, оскільки вони або дуже важкі за своєю конструкцією, або мають недостатньо високі характеристики, або мають високу вартість.

За рахунок використання сучасної елементної бази, мікропроцесорної техніки та програмного забезпечення досягається збільшення кількості виконуваних функцій, зменшення ваго-габаритних показників, стійкості до вібрацій, прискорення, підвищеної вологості, підвищеного рівня шумів, різких перепадів температури, а використання кварцового резонатора у зразковому генераторі підвищує точність частотоміра (похибка зменшується до похибки кварцового резонатора).

Віртуальним вимірювальним приладом є більш менш складне програмне забезпечення, встановлене на персональному комп'ютері, і інтерфейсний пристрій, що дозволяє ПК дістати доступ до тих фізичних величин і процесів, які він повинен буде обробляти. Як правило, у жості такого інтерфейсу виступає аналого-цифровий перетворювач з одним або декількома входами, можливо, забезпечений пристроєм нормування вхідного сигналу.

Віртуальні вимірювальні прилади на основі персонального комп'ютера майже завжди використовуються з більш менш розвиненим графічним інтерфейсом, що дозволяє обрати режим за допомогою клавіатури або миші через рівноманітні меню. Як буде показано у ході проектування, дуже зручно створювати невеликі застосунки, спеціально призначені для рішення того або іншого практичного завдання. Зазвичай вони створюються на універсальних поширених мовах програмування, таких як C++, Java, Delphi. Невелика структуризація цих програм дозволяє за умови перестановки лише відповідного драйвера застосовувати і промислові інтерфейсні пристрої, і ті, що можна

зібрати самостійно з окремих елементів. У Розділі 1 буде проведено порівняння обох варіантів, завдяки чому буде обрано рішення, найбільш відповідне технічному завданню.

Окрім виконання програм збору даних, користувач віртуального вимірювального приладу може експортувати результати вимірювань у розвиненіші програмні застосунки, наприклад електронні таблиці Excel або програми побудови діаграм. Ці офісні додатки роблять зрозумілими найабстрактніші записи або масиви даних, виділяючи в них непомітні на перший погляд тенденції або взаємні зв'язки. І, зрештою, файли цифрових даних, отримані при записі вимірюваних фізичних параметрів, можуть передаватися по лініях зв'язку, зокрема по електронній пошті і через Internet.

Можна розраховувати на те, що створюване програмне забезпечення USB-частотоміру для технічних вимірювань надасть широкі можливості, причому буде мати нижчу вартість, ніж класичний вимірювальний прилад (частотомір). Наприклад, такий USB-частотомір відповідає потребам учбових лабораторій при виконанні студентами лабораторних робіт з електрорадіовимірювань, коли кількість вимірювальних приладів зазвичай обмежена. Крім того, це дає можливість повернути до активної і корисної роботи персональні комп'ютери, які морально застаріли через швидку еволюцію обчислювальної техніки та політику компаній-виробників ПК і програмного забезпечення.

У даному дипломному проекті будуть розроблені програмно-апаратні засоби USB-частотоміру для технічних вимірювань (віртуальний частотомір). Цей вимірювальний прилад можна буде використовувати для виконання вимірювання частот електричних сигналів в діапазоні від 1 Гц до 250 кГц при напрузі вхідного сигналу від 50 мВ до 50 В з похибкою вимірювань біля 1%.

					РП 05. 08 000. 00 ДП ПЗ	Арх.
Зач.	Арх.	Не одоб.	Голов.	Директ.		8

1 АНАЛІЗ ІСНУЮЧИХ ЗАСОБІВ ДЛЯ ВИМІРЮВАННЯ ЧАСТОТИ

У даному розділі будуть розглянуті існуючі програмно-апаратні комплекси для проведення за їх допомогою вимірювань частоти електричних коливань в умовах електро-радіовимірвальної лабораторії або у радіо-аматорстві, у тому числі електронно-рахункові частотоміри класичного типу.

1.1 Використання електронно-рахункових частотомірів

При вимірюванні різних величин (сили струму, напруги, опору, електричної потужності, ємності, частоти, фази, індуктивності) зручно користуватися універсальними приладами, що дозволяє підвищити зручність проведення комплексних вимірювань, добитися деякої уніфікації, економії засобів, отримання результатів одного класу точності.

В умовах учбової лабораторії для цих цілей можна скористатися таким приладом, як частотомір, проте для практичних вимірювань різних електричних величин за допомогою частотоміра потрібні відповідні пристрої сполучення. Так, для вимірювання опору і ємності можуть бути використані RC-генератори, в яких залежність частоти від опору або ємності достатньо точно описується емпіричною функцією. Для вимірювання індуктивності, відповідно, використовуються LC-генератори. Схеми таких генераторів зустрічаються у всіх популярних радіотехнічних довідниках.

1.1.1 Класифікація і принцип дії частотомірів

Розглянемо електронний частотомір як основний пристрій для проведення вимірювань частоти постійної і змінної напруги, проте не забуватимемо, що використання такого частотоміру на жаль не дозволяє нам реєструвати зміни електричних величин в часі і в подальшому аналізувати їх.

Частотомір – вимірвальний прилад для визначення частоти періодичного процесу або частот гармонійних складових спектру сигналу.

Частотоміри класифікують:

					РП 05.08.001.00 ДП ПЗ	Арх.
						9
Зм.	Арх.	Наблюд.	Генер.	Дата		

- По методу вимірювання – прилади безпосередньої оцінки (напр. аналогові) і прилади порівняння (напр. резонансні, гетеродинні, електронно-рахункові);
- По фізичному сенсу вимірюваної величини – для вимірювання частоти синусоїдальних коливань (аналогові), вимірювання частот гармонійних складових (гетеродинні, резонансні, вібраційні) і вимірювання частоти дискретної дії (електронно-рахункові, конденсаторні);
- По виконанню (конструкції) – цитові, переносні і стаціонарні;
- По сфері застосування частотоміри діляться на два великі класи засобів вимірювань – прилади електровимірювань і радіовимірювальні прилади. Слід зауважити, що кордон між цими групами приладів великий прозорий. До групи приладів електровимірювань входять аналогові стрілочні частотоміри різних систем, вібраційні, а також, частково, конденсаторні і електронно-рахункові частотоміри. До групи радіовимірювальних приладів входять резонансні, гетеродинні, конденсаторні і електронно-рахункові частотоміри.

Призначення електронно-рахункових частотомірів (див. рис.1.1) – обслуговування, регулювання і діагностика радіоелектронного устаткування рівного призначення, контроль роботи радіосистем і технологічних процесів. Приклади таких приладів: ЧЗ-54, ЧЗ-57, Ф5137, ЧЗ-84, ЧЗ-63 М та ін.



Рисунок 1.1. Зовнішній вигляд електронно-рахункового частотоміру

Принцип дії електронно-рахункових частотомірів (ЕРЧ) заснований на підрахунку кількості імпульсів, сформованих вихідними ланцюгами з періодичного сигналу довільної форми, за певний інтервал часу. Інтервал часу вимірювання також задається методом підрахунку імпульсів, узятих з внутрішнього кварцового генератора ЕРЧ або із зовнішнього джерела (наприклад стандарту частоти). Таким чином ЕРЧ є приладом порівняння, точність вимірювання якого залежить від точності еталонної частоти.

Спрощена структурна схема електронно-рахункового частотоміра в режимі виміру частоти змальована на рис.1.2, а в режимі вимірювання часових інтервалів – на рис.1.3.

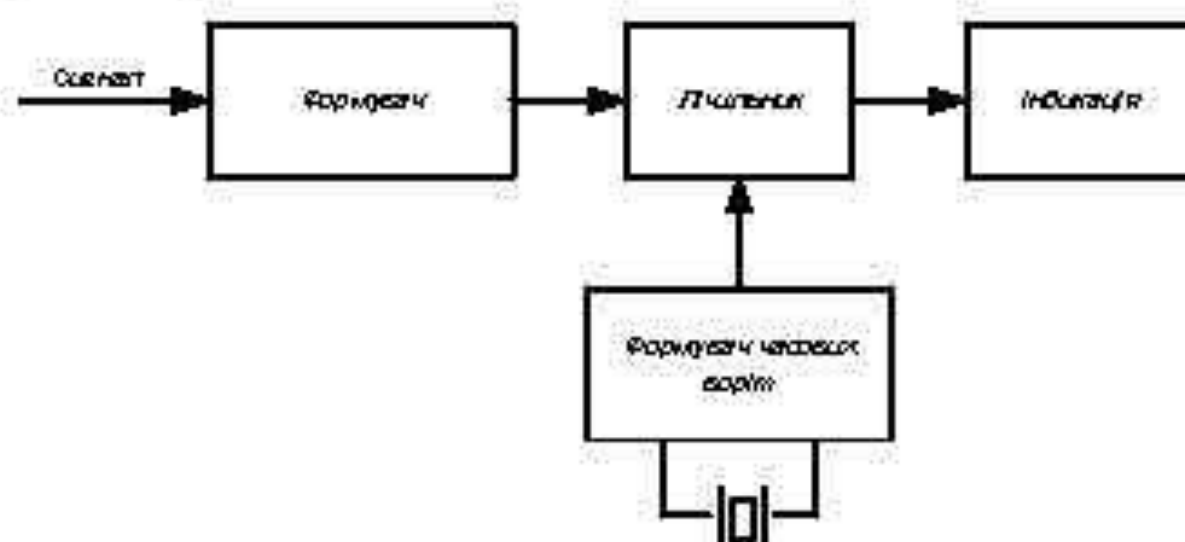


Рисунок 1.2. Спрощена структурна схема ЕРЧ при вимірюванні частоти

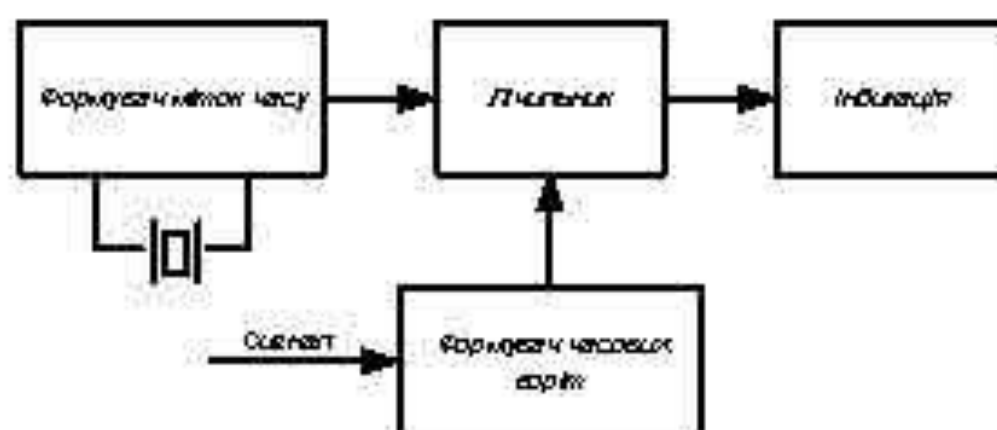


Рисунок 1.3. Спрощена структурна схема ЕРЧ при вимірюванні періоду

ЕРЧ є найбільш поширеним видом частотомірів завдяки своїй універсальності, широкому діапазону частот (від доль герца до десятків мегагерц) і високої точності. Для підвищення діапазону до сотень мегагерц – десятків гігагерц використовуються додаткові блоки – дільники частоти і переносники частоти.

Більшість ЕРЧ окрім частоти дозволяють вимірювати період слідування імпульсів, інтервали часу між імпульсами, відношення двох частот, а також можуть використовуватися як лічильники кількості імпульсів.

Деякі ЕРЧ (наприклад ЧЗ-64) поєднують в собі електронно-рахунковий і гетеродинний методи вимірювання. Це не лише підвищує діапазон вимірювання, але і дозволяє визначити несучу частоту імпульсно-модульованих сигналів, що простим методом рахунку не доступно.

Основні нормовані характеристики частотомірів:

- Діапазон вимірюваних частот;
- Допустима погрешність вимірювання (власної точності);
- Чутливість;
- Для ЕРЧ – не стабільність частоти кварцового генератора.

1.1.2 Призначення і використання апаратного частотоміра ЧЗ-67

Зовнішній вигляд апаратного електронно-рахункового частотоміру ЧЗ-67 показано на рис. 1.4, його характеристики представлені нижче.



Рисунок 1.4. Зовнішній вигляд частотоміру ЧЗ-67

Частотомір електронно-рахунковий ЧЗ-67 призначений для:

- вимірювання частоти синусоїдальних і частоти слідування імпульсних сигналів;
- вимірювання періоду синусоїдальних і періоду слідування імпульсних сигналів;
- вимірювання тривалості імпульсів і інтервалів годині;
- вимірювання відношення частот електричних сигналів;
- підсумовування електричних сигналів;
- ділення частоти електричних сигналів;
- вимірювання напруги опорних частот;
- роботи із змінними блоками.

Прилад за умовами експлуатації призначений для роботи в умовах:

- температура навколишнього середовища від 243 до 323 К (від мінус 30 до +50°C);
- підвищена вологість до 98% при температурі до 308 К (+35°C).

Прилад живиться від мережі змінного струму напругою (220 ± 22) В частотою $(50 \pm 0,5)$ Гц, (220 ± 11) В або (115 ± 6) В частотою $(400-12 \pm 25)$ Гц.

В приладі передбачена можливість роботи із змінними блоками і іншими приладами. Вживання змінних блоків і інших приладів дозволяє проводити вимірювання частоти в широкому діапазоні і значно розширює можливості приладу. При роботі із змінним блоком підсилювачем широкосмутовим ЯЗЧ-3 1/1 прилад вимірює частоту синусоїдальних сигналів в діапазоні від 0,1 до 60 МГц при рівні вхідного сигналу від 1 мВ до 10 В. При роботі із змінним блоком перетворювачем частоти ЯЗЧ-41 прилад вимірює частоту синусоїдальних сигналів в діапазоні від 0,1 до 1 Гц при рівні вхідного сигналу від 0,05 до 1 В. При роботі із змінним блоком перетворювачем частоти ЯЗЧ-42 прилад вимірює частоту синусоїдальних сигналів в діапазоні від 1 до 5 Гц при рівні вхідного сигналу від 0,2 до 10 мВ. При роботі із змінним блоком перетворювачем частоти ЯЗЧ-43 прилад вимірює частоту синусоїдальних сигналів в діапазоні від 4 до 12 Гц при рівні вхідного сигналу від 0,2 до 5 мВ. При роботі із

змінним блоком перетворювачем частоти автоматичним ЯЗЧ-72 прилад вимірює частоту синусоїдальних сигналів від 0.3 до 7 ГГц при рівні вхідного сигналу від 0.2 до 5 мВт. При роботі із змінним блоком перетворювачем частоти автоматичним ЯЗЧ-72 або перетворювачем частоти ЯЗЧ-42 і перетворювачем частоти Ч5-13 вимірюється частота синусоїдальних сигналів в діапазоні від 10 до 78.33 ГГц при рівні вхідного сигналу від 0,1 до 5 мВт (10 – 37.5) ГГц, від 0.5 до 5 мВт (37.5 - 70) ГГц і від 1 до 5 мВт (70 – 78.33) ГГц. При роботі із змінним блоком перетворювачем частоти ЯЗЧ-87 прилад вимірює частоту синусоїдальних сигналів і несучу частоту імпульсно-модульованих сигналів від 0.07 до 12 ГГц при рівні вхідного сигналу від 0.1 до 5 мВт. При роботі із змінним блоком перетворювачем частоти ЯЗЧ-88 прилад вимірює частоту синусоїдальних сигналів і несучу частоту імпульсно-модульованих сигналів від 8 до 18 ГГц при рівні вхідного сигналу від 0.4 мВт до 5 мВт.

Прилад може застосовуватися для налаштування, випробувань і калібрування різного роду приймально-передаючих трактів, фільтрів, генераторів, для налаштування систем зв'язку і інших пристроїв.

1.2 Віртуальні вимірні прилади на основі ПК

Для проведення дослідження багатьох електричних процесів звичайних мультиметрів (авометрів) і навіть частотомірів недостатньо, оскільки в цих випадках виникає необхідність відстежувати в часі і реєструвати досить швидкі зміни сигналу. Використання для цих цілей електронно-променевих осцилографів доцільно, але все одно не дозволяє зберігати результати, аналізувати і використовувати їх в розрахунках. Вихід – створення віртуального вимірного приладу на основі ПК – персонального комп'ютера, блоків сполучення і спеціального програмного забезпечення.

В даному випадку під «віртуальністю» мається на увазі той факт, що деяка частина функцій або вузлів вимірного приладу реалізується за допомогою персонального комп'ютера. Віртуальні вимірні прилади можна умовно розбити на дві категорії. Прилади першої групи «повністю віртуальні» і

фактично складається з однієї програми для ПК, а функції по вводу-виводу вимірюваного або генерованого сигналу покладаються на штатні апаратні засоби, наприклад, звукову карту [1][2]. Прилади другої групи мають окрім «виртуальної» частини ще і матеріальну. До комп'ютера в цьому випадку підключається жорсткий пристрій, який є невід'ємною частиною виртуального приладу. Такі пристрої реалізують деякі функції або вузли вимірювального приладу, які по тих або інших причинах не можна покласти на комп'ютер. Ці пристрої можуть мати різну складність – від складних промислових інтерфейсних плат, що вбудовуються в комп'ютер, до простих пристроїв, що підключаються до комп'ютера через USB, COM або LPT порт [3],[4],[5].

При порівнянні реальних і віртуальних приладів, окрім можливостей, що надаються, і режимів роботи, треба також приймати до уваги і їх основні характеристики – точність і швидкодію.

Точність віртуального приладу визначається не лише кількістю цифр пікселів, яке виводиться на екран управляючою програмою. До речі, ці цифри можуть бути помилковими, якщо не прийняті деякі заходи метрологічного характеру. Одним з основних критеріїв є розрядність аналого-цифрового перетворювача. Цей параметр визначає ступінь роздільної здатності при вимірюванні, тобто ту найменшу різницю між двома сусідніми значеннями, яку «відчуває» вимірювальний прилад. Наприклад, восьмирозрядний АЦП здатний формувати 2^8 , або 256 рівних значень вихідного сигналу (коду). Якщо його повна шкала складає 5 В, він зможе розрізнити два рівні вхідної напруги, що відрізняються приблизно на 20 мВ; це відповідає чутливості хорошого стрілочного гальванометра класу 0,4 або більшості осцилографів. Простий розрахунок показує, що вхідна напруга 4 В може бути зміряна з точністю близько 0,5%, а напруга 100 мВ – лише з точністю близько 20%. Тут виявляється відоме емпіричне правило «останньої третини шкали», яке, ймовірно, відоме всім користувачам аналогових мультиметрів і залишається настільки ж актуальним в цифрову епоху. 12-розрядний АЦП з можливістю формування на виході $2^{12} = 4096$ рівних значень зможе змряти напругу 4 В з

точність близько 0,03%, а 100 мВ - з точністю близько 1,2%. Зрозуміло, ці розрахунки вірні за умови, що всі електронні компоненти в АЦП мають допуски, відповідні вказаним величинам. Не варто, наприклад, розраховувати на середню точність 1% (що було б велими непогано) у вимірвальних ланцюгах, зібраних на звичайних резисторах з допуском 5%. Крім того, велике значення для точності всього пристрою має точність характеристик джерела опорної напруги, що працює з АЦП. Крім того, треба враховувати немінучий температурний дрейф, вплив якого може виявитися істотним при проведенні вимірювань в польових умовах в рідний час року. З вищесказаного виходить, що для отримання високоточних вимірювань недостатньо просто зібрати АЦП з 12-и розрядами або більше (зараз є і 16-, і навіть 24-розрядні). Так само, як і будь-який інший засіб вимірювання, віртуальний прилад повинен повертатися та калібруватися настільки часто, наскільки це необхідно, за винятком випадків, коли проводяться порівняльні або відносні вимірювання. Ось чому в даному проекті розглядаються лише АЦП з числом розрядів від 8 до 12, що вже зараз знайшли найрізноманітніше вживання. Що стосується швидкодії, слід вказати, що більшість віртуальних приладів, доступних за ціною, мають відносно вузьку смугу робочих частот. Це викликане, з одного боку, витратами часу, необхідними АЦП для виконання кожного перетворення «аналог-цифра», а з іншого боку – витратами часу, необхідними для програмної обробки результатів. Дешеві віртуальні прилади найчастіше обмежені за швидкістю вимірювань (від декількох десятків до декількох десятків тисяч відліків в секунду). Цього більше ніж достатньо, наприклад, для реєстрації кризових заряду або розряду акумуляторів, а також для вимірювання метеорологічних параметрів. Але для правильного відображення форми звукового сигналу на екрані віртуального осцилографа або для виконання серйозного спектрального аналізу ресурсів буде потрібно більше.

Користувач подібних приладів завжди повинен чітко уявляти можливості свого устаткування і програмного забезпечення (ПЗ) і враховувати їх, перш ніж робити остаточні висновки з отриманих результатів.

Існуючі готові рішення коштують від декількох десятків до декількох тисяч доларів залежно від пропонованих можливостей. Найкращі з них далеко перевершують можливості звичайних емпіричальних приладів, а найекономічніші рішення часто діють на широту їх потенційних застосувань. Розглянемо існуючі комерційні і професійні рішення в цій сфері.

1.2.1 Інтерфейсні плати внутрішнього виконання

Пристрої, виконані на платах, які встановлюються в слоти материнської плати ПК, дозволяють досягти в реальному масштабі часу частоти дискретизації в декілька десятків мегагерц і розрядності 16 або 24 біт, причому часто для декількох вхідних каналів одночасно.

Ці пристрої жорстко підключені до шини центрального процесора і можуть працювати в режимі прямого доступу до пам'яті (ПДП – DMA), а отже, повністю використовувати можливості найшвидших процесорів. Крайні моделі навіть можуть бути забезпечені буферною пам'яттю об'ємом до декількох мегабайтів, що дозволяє реєструвати найшвидші процеси в реальному масштабі часу.

Компанія National Instruments завжди була піонером в даній сфері і зараз як і раніше займає підтримуючі позиції в цій області ринку, що стала в даний час полем жорсткої конкуренції.

Традиційними споживачами вказаної категорії виробів в розвинених країнах є промислові галузі і наукові організації, інакше кажучи, споживачі, які не економлять на засобах, що дозволяють отримати метрологічно бездоганні результати.

1.2.2 Зовнішні інтерфейсні пристрої

У випадках, коли швидкодія не має первинного значення, просте і економічне рішення представляють зовнішні інтерфейсні пристрої, що підключаються до ПК через послідовний або паралельний порти. До цього типу інтерфейсів можна віднести як мініатюрний АЦП, розташований в корпусі роз'єму DB25, так і класичний мультиметр, забезпечений роз'ємом RS232, а

також настільний прилад, що має на корпусі декілька роз'ємів і органів керування.

Фірма National Instruments пропонує оригінальне рішення, що знаходиться посередині між двома вказаними варіантами і прекрасно підходить для портативних ПК. Віртуальний мультиметр VirtualBench, розроблений цією фірмою, має інтерфейс USB та Wi-Fi і об'єднує в собі осцилограф, логічний аналізатор, генератор сигналів довільної форми, цифровий мультиметр, частотомір і систему цифрового вводу-виводу. Діалогове вікно у режимі осцилографу (підключено спеціальний plug-in) показано на рис. 1.5.

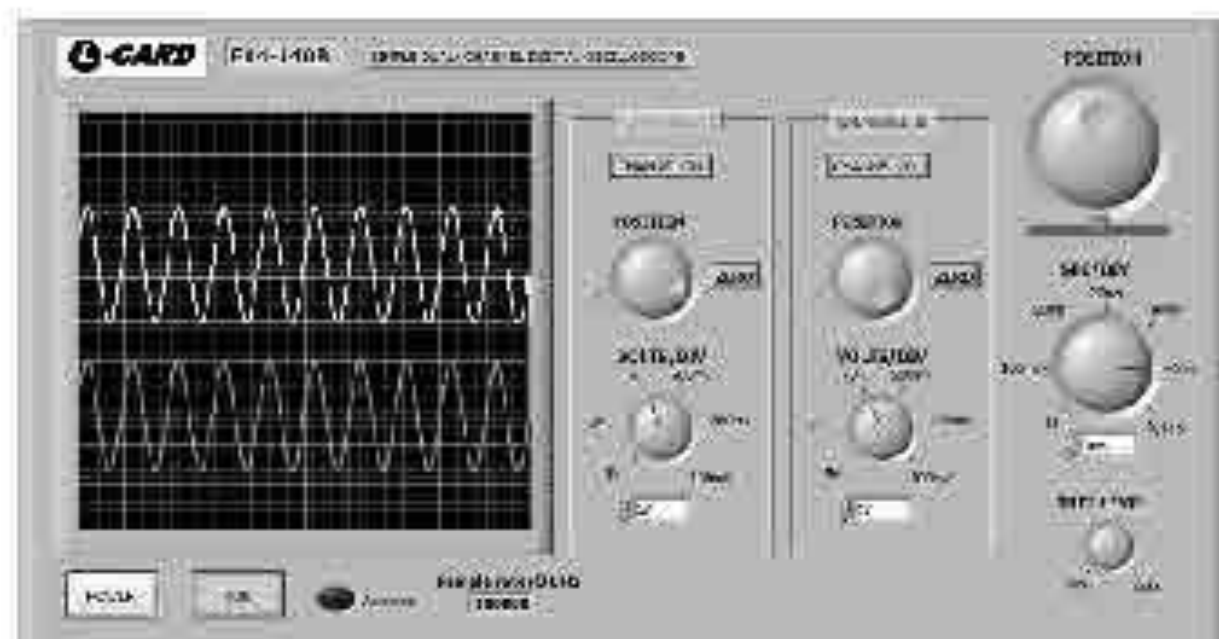


Рисунок 1.5. Діалогове вікно віртуального мультиметру фірми National Instruments

1.2.3 Віртуальний частотомір VFM

Цей вимірвальний прилад є аматорською конструкцією і описаний в журналі «Радиолюбби» [7]. У даній конструкції використовується мікроконтролер сімейства AVR AT90S2313 виробництва Atmel. Він зручний тим, що має у себе «на борту» аналоговий компаратор (на ньому реалізується вхідний формувач), швидкодіючі таймери-лічильники (підразунок імпульсів), послідовний порт (зв'язок з комп'ютером). Окрім цього, він містить 8-бітовий процесор RISC з швидкодією 10MIPS, 32 робочих регістра, 2 кілобайти ПЗП,

128 байт ОЗП, сторожовий таймер. Мікроконтролер має невелике енергоспоживання і може програмуватися внутрішньосхемно, без витягання з пристрою. За допомогою 25-контактного роз'єму пристрій підключається до СОМ-порту комп'ютера. Живлення схеми здійснюється від ліній DTR і CTS цього ж порту. Якщо порт комп'ютера не може забезпечити необхідну здатність (наприклад, це notebook) навантаження, то в цьому випадку живлення подається ззовні, через 13-й контакт роз'єму. Частотомір функціонує під керуванням комп'ютера, який подає йому команди через СОМ-порт. Назад, в комп'ютер, пристрій повертає відповіді на команди і дані вимірювань. Частотомір має всього один зовнішній вхід для подачі вимірюваного сигналу. Проте, при цьому сигнал може бути як цифровим так і аналоговим. Під різні режими вимірювань мікроконтролер проводить переконфігурування своїх внутрішніх апаратних засобів.

Основні технічні характеристики вимірювального приладу такі:

- Робочі режими - вимірювання частоти, вимірювання періоду, вимірювання інтервалу часу, підрахунок числа імпульсів;
- Прилад дозволяє вимірювати по аналоговому входу частоту періодичних сигналів, які мають один екстремум позитивної полярності за період. Діапазон вимірюваних частот від 0,1 Гц до 250 кГц при напрузі вхідного сигналу від 0,1 до 30 В;
- Прилад дозволяє вимірювати по цифровому входу частоту цифрових сигналів в діапазоні від 0,1 Гц до 5 МГц. Напрята логічного нуля вимірюваного сигналу повинна знаходитися в діапазоні 0...2 В, логічної одиниці - в діапазоні 2,5...30 В;
- Час рахування 1 мс, 10 мс, 100 мс, 1 с, 10 с;
- Прилад дозволяє вимірювати одиничний і усереднений період періодичних сигналів позитивної полярності в діапазоні від 10 мкс до 10 с при напрузі вхідного сигналу від 0,1 до 30 В;
- Період міток часу (роздільна здатність) 0,1 мкс;

- Число усереднюваних періодів 10, 100, 1000, 10000.



Рисунок 1.6. Інтерфейс управляючої програми приладу VFM

Керувача програма (рис.1.6) працює під керуванням ОС сімейства Windows. Вона реалізує підтримку всіх заявлених режимів вимірювання. Окрім цього, за її допомогою можна вести журнал вимірювань.

1.3 Висновки за розділом

Після аналізу різних варіантів побудови та організації вимірювальних приладів – частотомірів, описаних у даному розділі, можна відзначити, що для виконання технічного завдання найбільш придатне використання віртуального частотоміру на основі програмно-апаратних засобів для ПК, а саме – АЦП, який буде сполучений з послідовним портом USB та ПЗ, що в базовому рішенняні буде представляти собою віртуальний прилад для вимірювання частоти електричного струму. Проте, функціональність такого приладу може бути істотно розширена для інших застосувань. Таким чином, можна розраховувати на те, що створюване програмне забезпечення USB-частотоміру надасть набагато ширші можливості, причому за ціною буде дешевше класичних вимірювальних приладів, що мають схожий рівень технічних характеристик. Такий підхід дозволить обмежитися мінімальними витратами, що відповідає, наприклад, потребам учбової лабораторії або аматорським застосуванням, де кількість вимірювальних приладів обмежена. Окрім того, це дасть можливість повернути до активної і корисної роботи морально застаріли ПК.

Зм.	Арх.	Наблюд.	Генер.	Датум

РП 05.08.001.00 ДП ПЗ

Арх.

20

2 ВИБІР І АНАЛІЗ СХЕМ АНАЛОГО-ЦИФРОВОГО ПЕРЕТВОРЕННЯ

Існує безліч способів для перетворення аналогового сигналу – електричної напруги або струму, що змінюється плавно і безперервно, в потік цифрових даних, що представляє собою дискретну кодовану послідовність імпульсів. На практиці найчастіше використовується аналого-цифрове перетворення за допомогою імпульсно-кодової модуляції (ІКМ). В цьому випадку процес починається з представлення безперервного сигналу у вигляді послідовності відліків, які беруться через певний проміжок часу (накільки кажучи, з певною частотою дискретизації). Цю функцію виконує схема, що називається пристроєм вибірки-зберігання. Запам'ятовувачи миттєве значення вхідного сигналу (найчастіше на конденсаторі), цей пристрій забезпечує збереження величини узятго відліку на час процесу оцифрування, який полягає у представленні амплітуди кожного відліку у формі двійкового кодового слова з певною кількістю розрядів. Спосіб, використовуваний для виконання такої оцифрування, і визначає можливості, складність і вартість аналого-цифрового перетворювача.

2.1 Схемотехніка сучасних АЦП

Головна особливість сучасних АЦП полягає у організації управління по одно- або двопровідній послідовній шині (SPI, Microwire, I2C і тому подібне), а не через паралельний інтерфейс, що вимагає наявності одного виводу мікросхеми на кожен розряд шини управління. Звичайно, такий спосіб передачі бітів даних – один за іншим по одному дроту – обмежує швидкість обміну інформацією, хоча і тут можна досягти швидкості передачі даних до 1 Мбіт/с. На практиці, з врахуванням властивостей і можливостей схем дискретизації і квантування, не варто розраховувати на подолання бар'єру в декілька десятків тисяч вимірювань в секунду, що в середньому відповідає частоті дискретизації 20 кГц. Тому при розрядності від 8 до 12 біт вони прекрасно підходять для рішення більшості задач у області створення віртуальних вимірювальних

					РП 03.03.002.00 ДП ПЗ	Арх.
Зм.	Арх.	Наблюд.	Підпис	Дата		21

прикладів. На рис. 2.1 наведена структурна схема стандартних послідовних АЦП, виконаних в корпусі з восьма виводами. Логічний контролер з вбудованим тактовим генератором управляє роботою схеми перетворення, що функціонує за принципом послідовного наближення. Цей принцип полягає в поступовому покроковому накопиченні в проміжному регістрі даних двійкового кодового слова, відповідного співвідношенню вхідної аналогової напруги (різниці між рівнями напруги на диференціальних входах ANALOG +IN і ANALOG -IN) і опорної напруги (різниці між рівнями напруги на диференціальних входах REF+ і REF-).

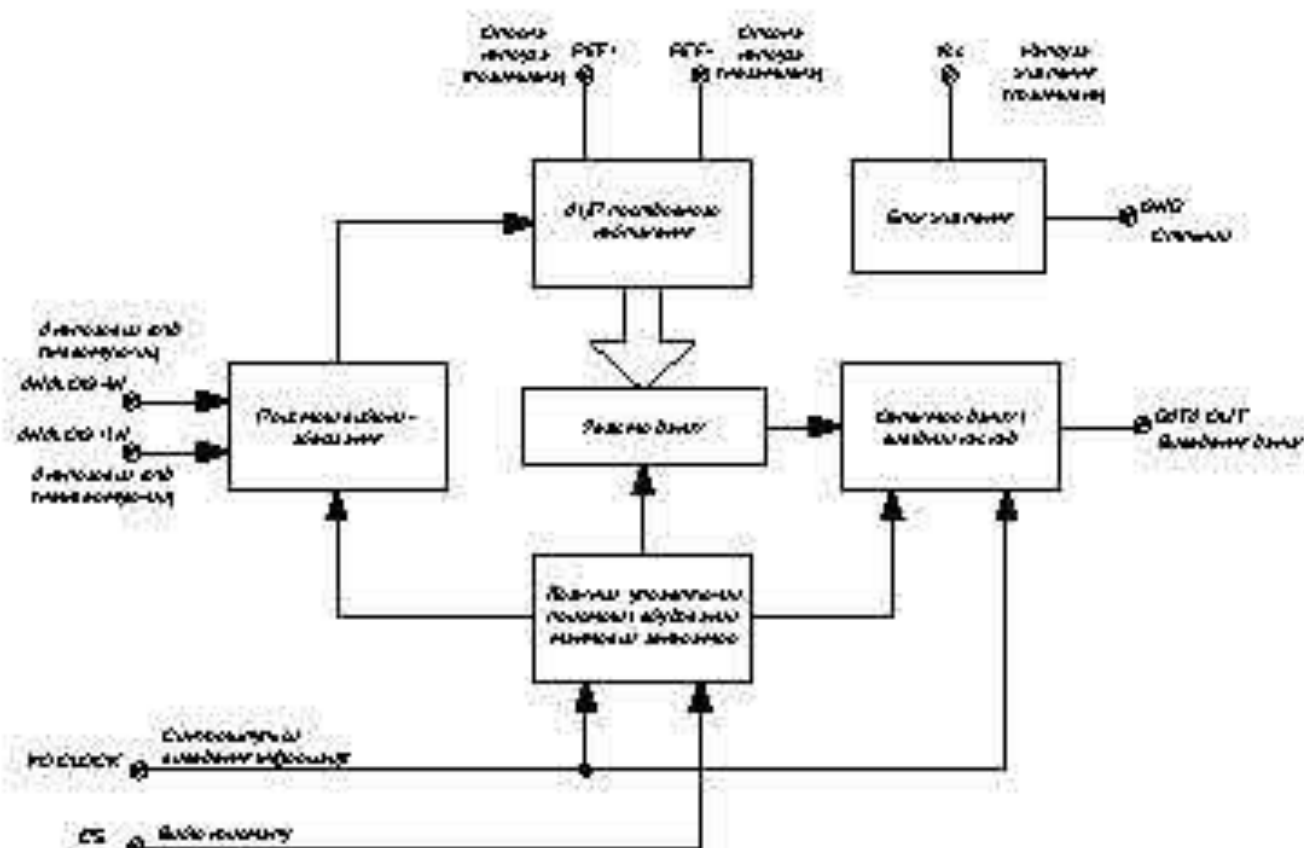


Рисунок 2.1. Структурна схема послідовного АЦП

Цикл перетворення починається з фіксації рівня вхідної аналогової напруги в пристрої вибірки-зберігання, а потім перетворювач починає формувати вміст регістра. Процес накопичення кодового слова займає деякий час, що називається часом перетворення.

На практиці як мінімум один з входів ANALOG -IN або REF- тежню логічно сполучений із спільним дротом (GND), унаслідок чого лише дві

моделі послідовних АЦП можуть працювати за схемою з диференціальним виходом. Після закінчення процесу перетворення селектор даних, розташований перед вихідним каскадом, починає послідовно вибирати біти інформації, що містяться в регістрі даних, і потім також послідовно подає їх на вихід DATA OUT. Частота вибірки і формування вихідних імпульсів визначається зовнішнім тактовим сигналом I/O CLOCK.

Для правильного функціонування АЦП необхідна наявність деякої зовнішньої управляючої системи, яка повинна формувати послідовності синхронізуючих сигналів. В більшості випадків аналого-цифрове перетворення починається у момент подачі відповідного сигналу на вхід CS (вибір кристала). Само по собі перетворення виконується за декілька десятків мікросекунд. Після цього необхідно подати потрібну кількість тактових імпульсів на вхід I/O CLOCK, щоб вивести результат перетворення через вивід DATA OUT. Хоча величина тактової частоти виведення даних для деяких моделей має обмеження знизу (наприклад, 100 кГц), вона все-таки набагато нижче, ніж внутрішня тактова частота перетворення. Як правило, швидкість послідовного виведення інформації залежить лише від швидкості, з якою управляюча система може обробляти біти даних, що надходять.

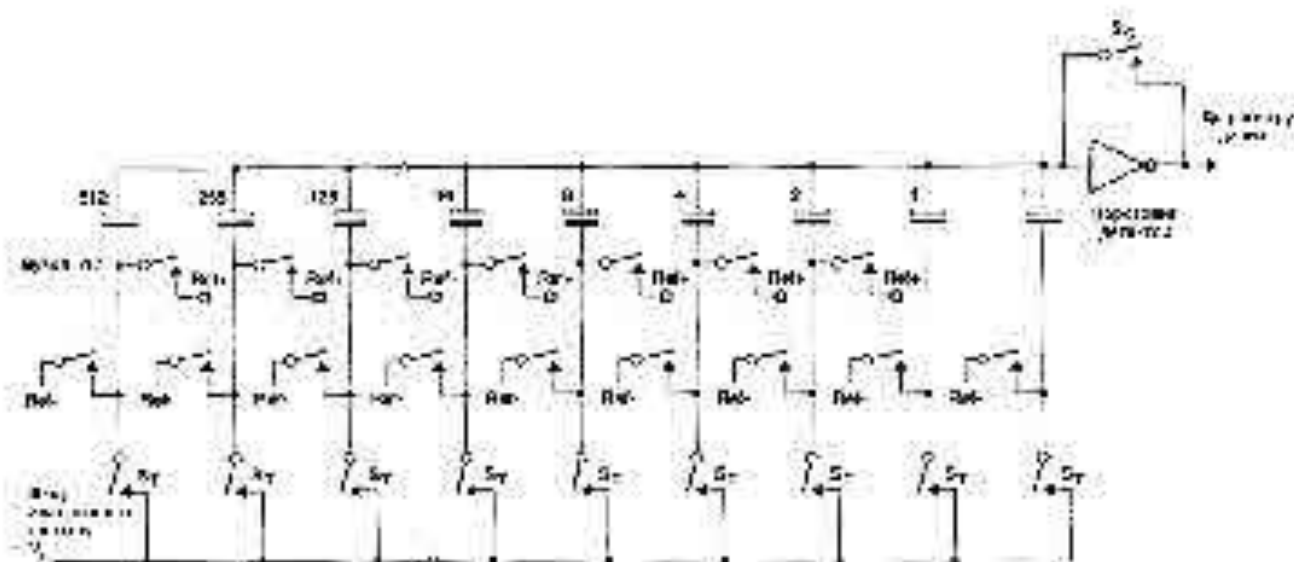


Рисунок 2.2. Функціональна схема АЦП послідовного наближення з комутувачою матрицею конденсаторів

Основовою функціональною схемою АЦП послідовного наближення (рис. 2.2) служить комутувача матриця конденсаторів. Значення кожного біта даних визначається пороговим детектором залежно від величини заряду кожного з вагових конденсаторів, що складають матрицю АЦП. Рис. 2.2 відповідає десятирозрядному АЦП. По малюнку видно, що число конденсаторів, що містяться в матриці, має бути на одиницю більше, ніж число розрядів. Відносна вага кожного ланцюга матриці конденсаторів визначається відповідно до послідовності ступенів числа два, тобто від 1 до 512 для десятирозрядного АЦП. На початку циклу перетворення логічний управляючий пристрій замикає всі вкл. S_T і S_C , унаслідок чого всі конденсатори одночасно заряджають до рівня вхідної напруги V_i (режим вибірки). Потім всі вкл. розмикаються (режим зберігання) і пороговий детектор починає формувати біти даних, порівнюючи напругу REF- з напругою на кожному з конденсаторів матриці.

Першою обробляється напруга на конденсаторі, встановленому в ланцюзі матриці з максимальною вагою (в даному випадку вага 512). Для цього вузол 512 підключається до джерела напруги REF+, а решта всіх вузлів підключається до REF-. Якщо напруга в спільній кінці більше порогу спрацювання детектора, приблизно рівного $V_{CC}/2$, то біт даних встановлюється в нульовий стан, а вузол 512 підключається до джерела напруги REF-. Якщо ж напруга в спільній кінці менше порогу, біт даних встановлюється в одиничний стан, а вузол 512 підключається до джерела REF+ до завершення процесу перетворення. Та ж послідовність операцій повторюється для 1 ланцюга з вагою 256, потім з вагою 128 – і так до тих пір, поки не буде визначено значення кожного розряду регістра даних.

Слід пам'ятати, що n-розрядний АЦП може сформувати лише 2^n різних двійкових кодових слів (256 для 8 розрядів, 1024 для 10 розрядів, 4096 для 12 розрядів і так далі). Різниця між двома сусідніми кодовими словами відповідає аналоговій вазі молодшого значущого розряду (МЗР, або LSB в іноземній літературі). Зміна вхідної напруги на величину, меншу в порівнянні з цією різницею, може бути не відображена у вихідних даних. Деякі перетворювачі,

крім того, вносять додаткову погрешність і формують результат з точністю до одного або навіть до двох МЗР. Така дискретність результату перетворення є слідством характерної для АЦП передавальної характеристики у вигляді «сходинки» (рис. 2.3а).

Абсолютно очевидно, що в цьому випадку, як і при будь-якому іншому процесі аналого-цифрового перетворення, отриманий результат містить деяку погрешність, звану помилкою квантування. Причини її виникнення пояснюються в графіку, приведеному на рис. 2.3б.

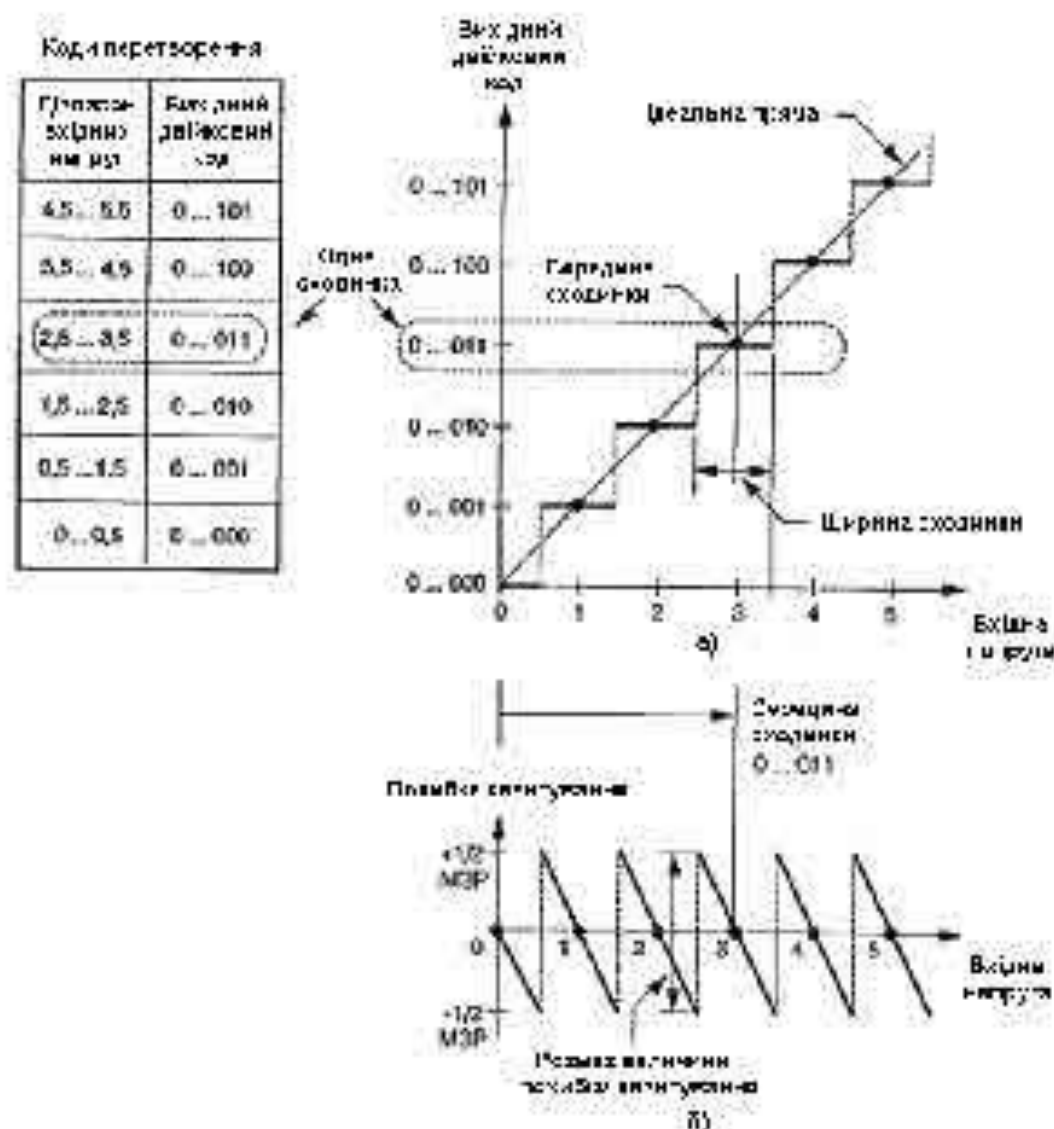


Рисунок 2.3. Передавальна характеристика АЦП і таблиця кодів перетворення

Помилка квантування з'являється в результаті заміни дійсної величини відліку вхідного сигналу її дискретним еквівалентом у вигляді вихідного коду і

визначається ж різниця між графіками ступінчастої передавальної характеристики і ідеальної прямої. Максимальна величина помилки для передавальної характеристики, приведеної на рис. 2.3а, дорівнює половині величини кроку квантування (половині аналогової ваги МЗР).

Важливо враховувати цю особливість, яка, якщо узяти для прикладу 8-розрядний АЦП з повною шкалою 5 В, виявляється в неминусій потребності перетворення величиною до 20 мВ. І хоча ця величина складає лише 0,4%, її слід брати до уваги.

2.2 8-розрядний АЦП з послідовним інтерфейсом

8-розрядні послідовні АЦП є і найдавнішими і найпростішими у вживанні. Якщо вони працюють в останній третині своєї повної шкали (що можна забезпечити за допомогою попереднього підсилювача із зсувом), їх роздільна здатність з точністю 1/256 цілком відповідає точності, потрібній в багатьох додатках віртуального вимірювального приладу.

Компоненти такого типу пропонують багато виробників, але до якої небудь стандартизації і уніфікації в цій області ще далеко. Розташування виводів і протоколи зв'язку в різних типах подібних АЦП різні. Можливо, виробники роблять це для того, щоб утруднити заміну компонентів на виробі конкурентів.

8-розрядний послідовний АЦП, який використовуватиметься в прикладах, був вибраний, з одного боку, зважаючи на його широке поширення і цілком прийнятну ціну, а з іншого боку з огляду на те, що існують 10- і 12-розрядні моделі, повністю сумісні з ним по розташуванню виводів. При нагоді це може спростити проблему модернізації друкованих плат.

На рис. 2.4 наведена схема розташування виводів АЦП TLC 549 Texas Instruments. Саме ця модель розглянута нижче в її практичному вживанні.

Такий АЦП має лише один аналоговий вхід (ANALOG IN) і два входи для підключення опорної напруги (REF+ і REF-). За другий аналоговий вхід можна вважати вивід GND – спільний. Якщо при цьому вивід REF- теж підключений

до спільного дроту, то на виході АЦП формуватиметься байт вихідного коду, рівний 00000000 при нульовій напрузі на аналоговому вході, і 11111111 - при вхідній напрузі, рівній опорній, яка подається на вивід REF +.



Рисунок 2.4. Розташування виводів АЦП TLC 549, TLC 1549

Протокол зв'язку цього АЦП досить простий, його часові діаграми приведені на рис. 2.5.

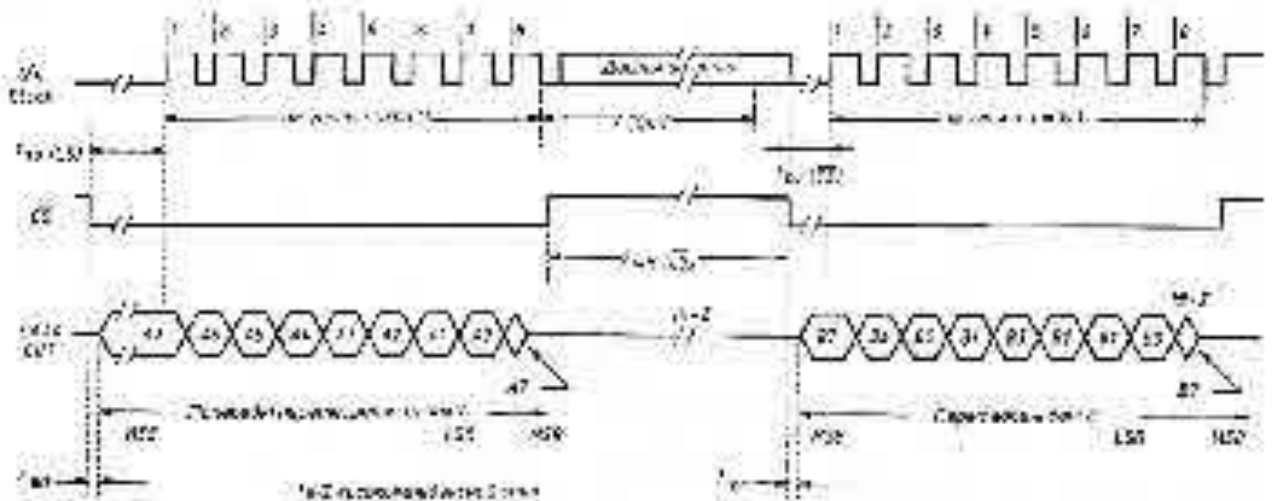


Рисунок 2.5. Протоколи зв'язку АЦП TLC 549

Під час переходу сигналу на виводі CS від високого до низького рівня в регістр введення даних поміщається результат попереднього перетворення. Тому рекомендується виконати «порожнє перетворення» відразу після ввімкнення пристрою або в разі виконання двох перетворень, розділених допустимим інтервалом часу. Перше лічбове значення вихідного коду в цих випадках буде невірним. Наступне перетворення буде правильним, воно почнеться по першому передньому фронту імпульсу, що поступить на вивід CS після низького рівня. Важливо, щоб на виводі CS був високий рівень протягом всього часу перетворення, але, оскільки цей процес триває всього декілька

десятків мікросекунд, дану умову можна вважати за автоматично виконувану, якщо АЦП управляється процесором з програмою, написаною на істотно менш швидкій мові, ніж Assembler.

Кожен біт вхідних даних може бути підрахований на виводі DATA OUT, причому біти виводяться старшими розрядами (MSB) вперед по передньому фронту імпульсів на виводі I/O CLOCK.

Одна з переваг цього АЦП полягає в тому, що він може працювати з будь-якою частотою дискретизації, визначуваною управляючою системою – від одного вимірювання за декілька годин до двадцяти тисяч вимірювань в секунду.

2.3 10-розрядний АЦП з послідовним інтерфейсом

Роздільна здатність в 10 біт – це своєрідний компроміс між 8 – і 12-розрядними пристроями, що дозволяє усунути проблеми, пов'язані з недостатньою в деяких випадках точністю, так само як і з складністю у виготовленні і налаштуванні.

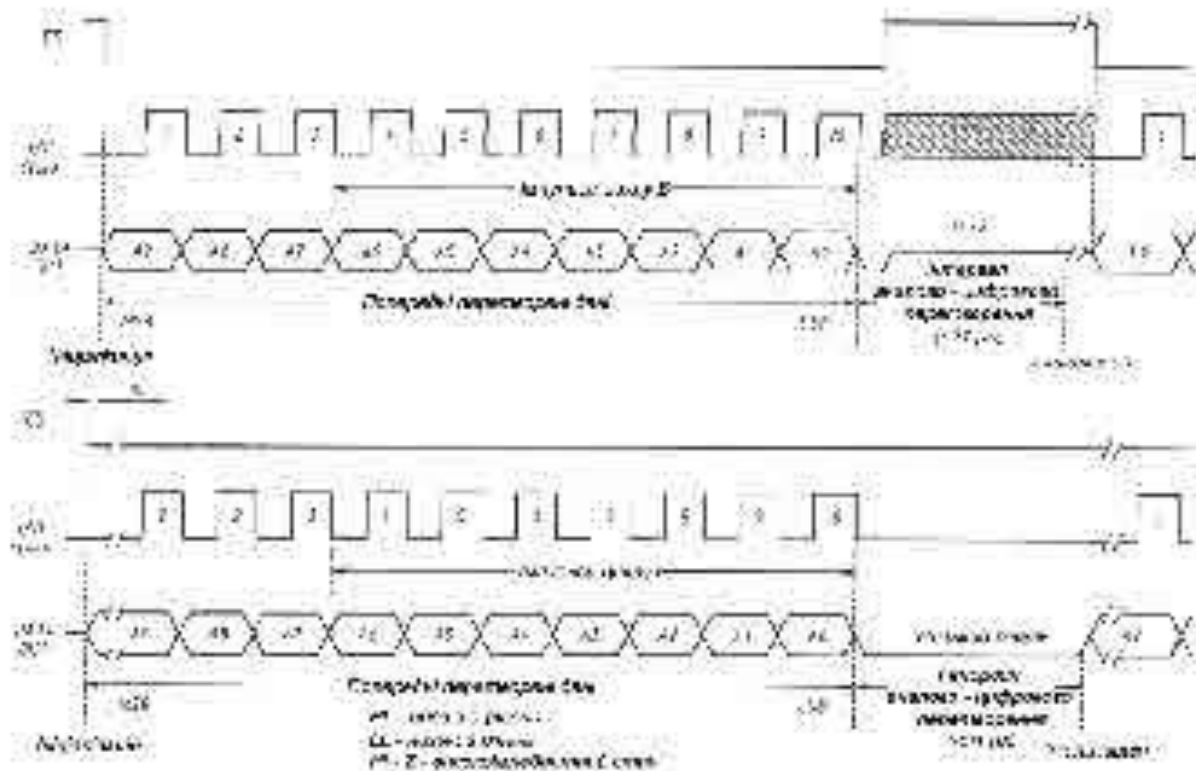


Рисунок 2.6. Протоколи зв'язку АЦП TLC 1549

Аналог 0-цифровий перетворювач TLC 1549 виробництва Texas Instruments зручний тим, що він сумісний по розташуванню виводів з TLC 549 (рис. 2.4), але цей компонент допускає шість різних протоколів зв'язку (три «швидкі» і три «повільні»). Протокол зв'язку, наведений на рис. 2.6, дуже схожий на протокол TLC 549, з тєю лише відмінністю, що тут використовуються 10 біт даних замість 8.

Прямий конкурент АЦП TLC 1549 – аналого-цифровий перетворювач MAX 1243 виробництва компанії MAXIM. Окрім своїх відмінних характеристик, він цікавий тим, що належить до сімейства повністю взаємозамінних АЦП, яке включає 12-розрядну версію MAX1241, а також тим, що за допомогою простої програми його можна легко переключити в 8-розрядний режим. Таке ж розташування виводів мають і раніші виробни компанії MAXIM (MAX 187, MAX 189), і сучасні моделі (MAX 1240, MAX 1242). Деякі з цих АЦП забезпечені вбудованим джерелом опорної напруги, що дає додаткові переваги при розширенні можливостей пристроїв. Існує два варіанти протоколу зв'язку, використовуваного при роботі з АЦП MAX 1243 (рис. 2.7).

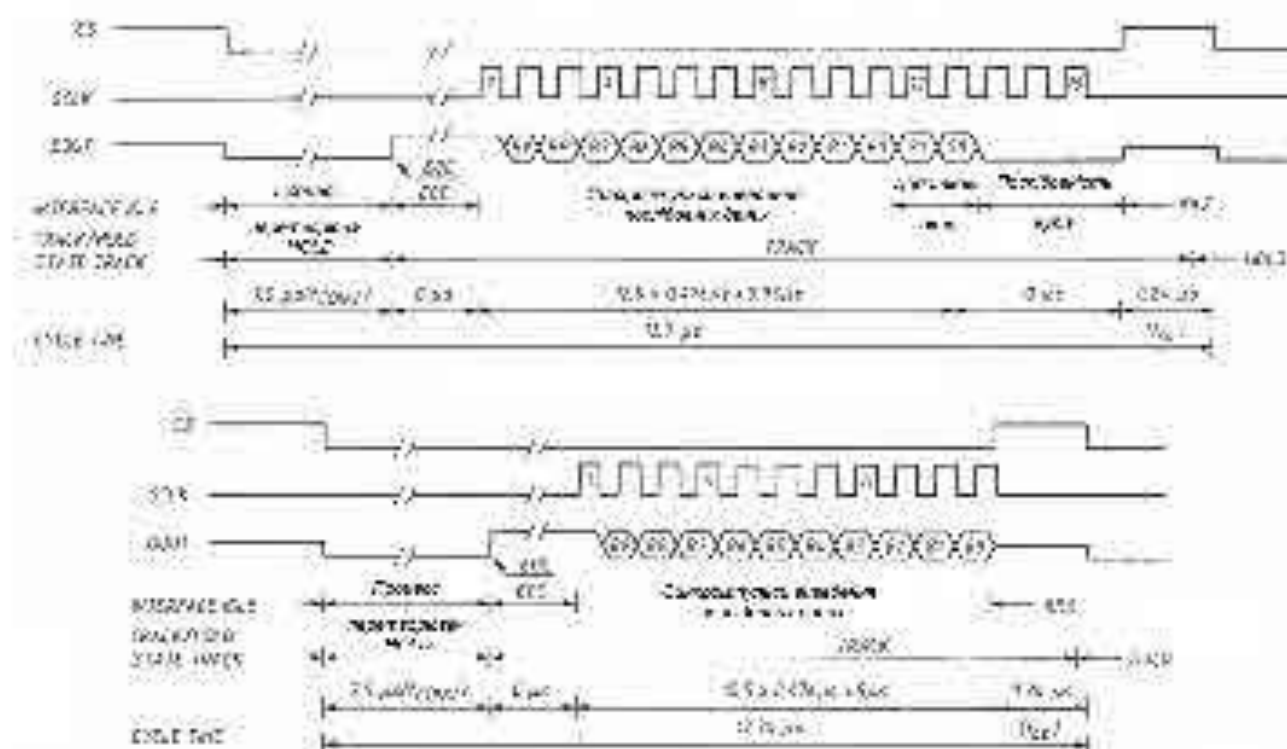


Рисунок 2.7. Протоколи зв'язку АЦП MAX 1243

Слід враховувати, що на початку кодової посылки присутній «одичинний» біт, в деякому роді стартовий; його потрібно виключити перед обробкою, наприклад подавши один додатковий тактовий імпульс.

На відміну від вищеописаних компонентів, АЦП MAX 1243 повинен виконати перетворення перед тим, як вивести слово даних. Отже, після подачі рівня логічного нуля на вивід CS (початок перетворення) треба чекати не менше 7,5 мкс або чекати переходу сигналу на виводі DOUT в стан логічної одиниці, перш ніж подати позитивний перепад напруги (фронт імпульсу) на вивід SCLK для початку виведення даних. Перевага такого способу роботи полягає в тому, що отримуваний результат відповідає поточному циклу перетворення, а не попередньому, як у вище описаних пристроїв. Крім того, при виведенні результату можна використовувати лише всі старші значущі розряди, навше кажучи, використовувати MAX 1243 в 8-розрядному режимі дуже високої точності. Для цього досить припинити передачу даних після біта B2, подавши на виведення CS сигнал логічної одиниці.

2.4 12-розрядний АЦП з послідовним інтерфейсом

Аналого-цифровий перетворювач MAX 1241, поєднує відповідний АЦП MAX 1243 по розташуванню виводів, є 12-розрядною версією, яка використовує протокол зв'язку, дуже, схожий з протоколом, представленим на рис. 2.8. Поєднує протокол зв'язку АЦП MAX 1241 наведений на рис. 2.9.

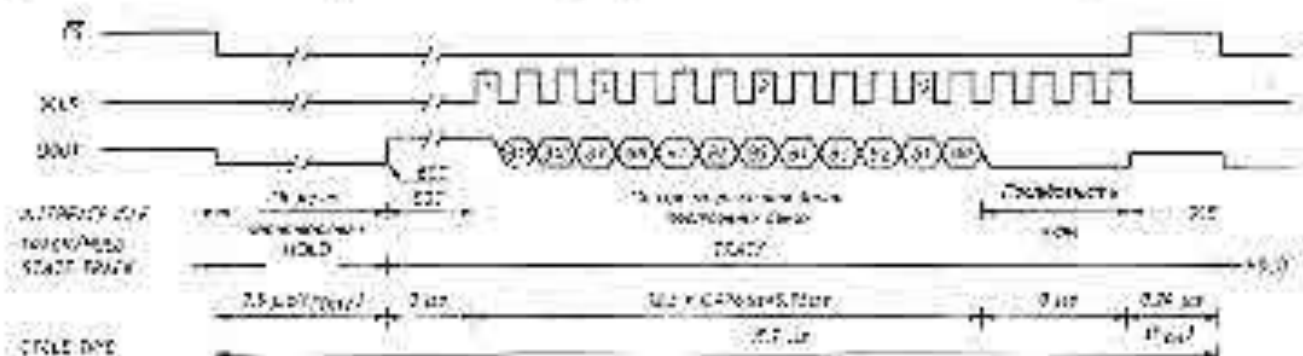


Рисунок 2.9. Протокол зв'язку АЦП MAX 1241

Серед напівпровідникових компонентів компаній Linear Technology і Vishay можна знайти 12-розрядні АЦП, які по розташуванню виводів схожі з

але ускладнює вживання програм на мові BASIC або навіть на мові PASCAL, що працюють на повільному процесорі. Звичайно, в нижньої межі тактової частоти є певні допуски на практичне використання, але все таки потрібна обережність при оцінці точності отримуваних результатів.

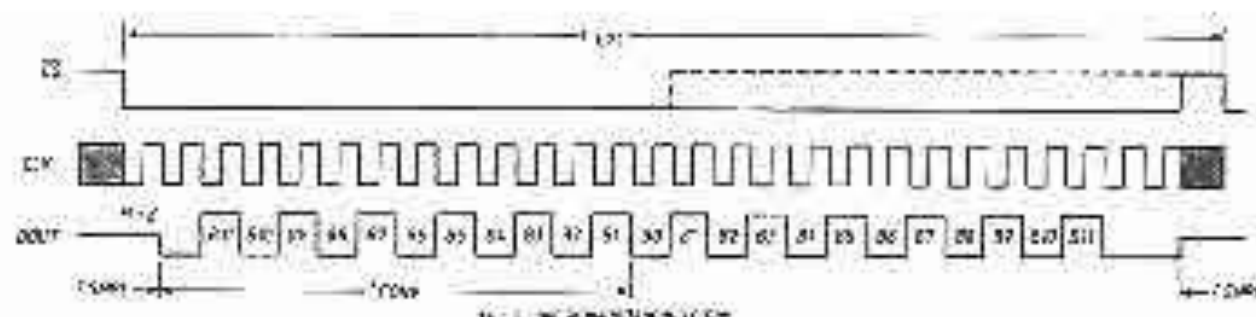


Рисунок 2.11. Протокол зв'язку АЦП LTC 1292

2.5 Аналіз необхідності вживання АЦП з розрядністю, більшою 12

Спробуємо відповісти на питання про те, наскільки виправдано вживання аналого-цифрових перетворювачів з розрядністю більшою 12 для вирішення відносно простих завдань при побудові віртуального вимірювального приладу – частотоміру.

У промисловості і науці широко застосовуються 16-, 24-розрядні пристрої і навіть пристрої з більшою розрядністю. Оптимальна експлуатація таких АЦП з високою роздільною здатністю 0,015% і 6 ppm = 6×10^{-6} (у інземній літературі часто зустрічається одиниця ppm – часток на мільйон – або 10^{-6}) передбачає особливу акуратність в питаннях, що стосуються високої точності і калібрування всього вимірювального ланцюга – від датчиків до пристроїв індикації і друку. У будь-якому випадку це передбачає великі витрати, що важливо. Для того, щоб оцінити ситуацію, слід взяти до уваги, що по ширині аркуша формату А4 навіть при роздільній здатності 600 dpi лазерний принтер зможе розмістити лише 7000 окремих крапок, а графічний екран вміщає, наприклад, 1024 крапки по вертикалі. Отже, за допомогою цих пристроїв неможливо графічно точно відтворювати результати вимірювань, зроблених шістнадцятирозрядними приладами, в десятки і сотні разів точнішими.

Вияток становлять випадки виведення особливих ділянок з великим збільшенням. Тому в даному проекті не розглядаються практичні конструкції АЦП з розрядністю більшою 12, які до того ж украй рідко випускаються в корпусах з вісьма виводами і забезпечені послідовним інтерфейсом.

Основна увага надалі буде приділена тому, як добитися вдалих результатів (найпростішими способами), використовувачи 8- або 10-розрядні АЦП.

2.6 Висновки за розділом

На основі розглянутих вище характеристик та особливостей роботи сучасних мікросхем АЦП можна зробити висновок, що для реалізації поставленого завдання найдоцільніше буде використання 10-розрядного АЦП. Роздільна здатність в 10 біт – це своєрідний компроміс між 8 – і 12-розрядними пристроями, що дозволяє усунути проблеми, пов'язані з недостатньою в деяких випадках точністю, так само як і з складністю у виготовленні і налаштуванні. Оберемо 10-розрядний аналого-цифровий перетворювач TL C 1549 виробництва Texas Instruments, який зручний тим, що він сумісний по розташуванню виводів з TL C 549, але цей компонент допускає цілий різний протоколів зв'язку (три «швидких» і три «повільних»), що дозволить дуже гнучко налаштувати систему. Окрім того, у жорстці джерела опорної напруги для цього АЦП найкраще підійде інтегральний стабілізатор LM2931.

3 РОЗРОБКА БЛОКУ СПОЛУЧЕННЯ З ПК ДЛЯ USB-ЧАСТОТОМІРУ

У даному розділі будуть розроблені структурні та принципові електричні схеми блоку сполучення з персональним комп'ютером для USB-частотоміру.

Згідно технічному завданню, USB-частотомір має бути сполучений з ПК через послідовний інтерфейс. Найдоцільніше буде використання універсального інтерфейсу USB, який встановлюється в усіх поколіннях ПК, починаючи з 2000-х років. Окрім того, для роботи з відносно слабкими електричними сигналами буде розроблено схему входного підсилювача.

Структурна схема вимрювального приладу на базі персонального комп'ютера, яка побудована на основі вивпадених у технічному завданні вимог та проведеного у розділах 1.2 аналізу, показана на рис.3.1.

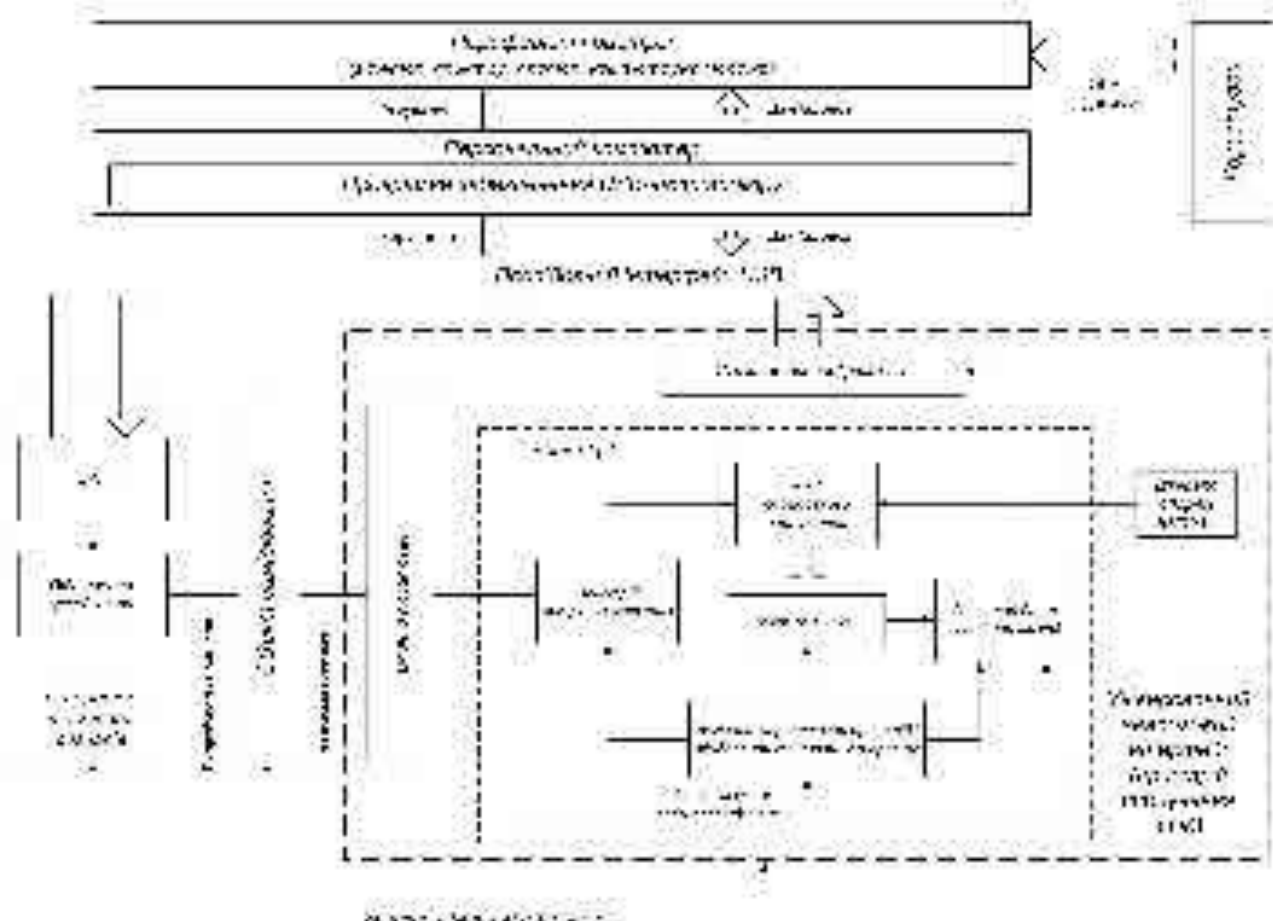


Рисунок 3.1. Структурна схема USB-частотоміру на базі ПК

Основне призначення USB-частотоміру – можливість вимірювання і спектрального аналізу частоти електричних коливань, а для роботи з будь-якими аналоговими сигналами від датчиків необхідні відповідні пристрої сполучення, які не входять в дану розробку, але описані в літературі [1], [2], [4].

3.1 Розробка принципової електричної схеми блоку АЦП

Принципова схема, що наведена на рис. 3.2, побудована на основі схеми промислового АЦП AD C 10 (рис. 1.7). Зокрема, застосований аналогічний вхідний каскад з дільником напруги. При використанні тих же номіналів резисторів R4 і R5 (100 кОм для AD C 10) можна забезпечити автоматичну сумісність цього пристрою зі всіма приставками – пристроями нормування сигналів, датчиками і тому подібне. Підбір резисторів вхідного дільника у будь-якому випадку вимагає акуратності.

Треба враховувати, що вхідний опір мікросхеми АЦП послідовного наближення (див. 2) має, принаймні під час виконання перетворення, майже лише ємнісний характер. При напрузі живлення 5 В струм витоку в межах 1 мкА відповідає активній складовій опору 5 МОм, що значно більше вхідного опору звичайного осцилографа. Ємнісна ж складова може досягати 30 пФ в мікросхемі TL C 1549. Це не сильно відрізняється від того, що властиве входу класичного осцилографа, але випадок, який нас цікавить, абсолютно особливий. Вхідна ємність утворена елементами пристрою вибірки-зберігання, і вона постійно змінюється. Інтегруючий фільтр, що утворюється цією ємністю С і опором R, включеним послідовно з входом, визначає час наростання сигналу, пропорційний добутку RC.

Якщо частота дискретизації дуже велика відносно частоти зрізу інтегруючого фільтру, то перетворення почнеться в той момент, коли еквівалентний конденсатор з ємністю С буде заряджений ще не повністю, і, отже, результат вимірювання буде не вірним.

Змк.	Арх.	Наблюд.	Голов.	Дата

РП 05. 08 003. 00 ДП ПЗ

Величини резисторів, вибрані для вхідних ланцюгів ADC 10 (рис. 3.2), були розраховані так, щоб гарантувати бажану точність на високих частотах дискретизації, на якій можуть працювати ці прилади (близько 20 кГц). Проте номінали резисторів можна збільшувати, якщо споживача влаштовують не настільки високі швидкості вимірювань. У звичайних випадках підійде номінал 100 кОм (з допуском 1% або менше).

Описуваний в проекті пристрій буде використовуватися для вимірювання параметрів сигналів відносно спільного або мережевого дроту. Але у разі ситуації можуть виникнути проблеми через те, що спільний дріт пристрою сполучений з корпусом ПК.

За винятком тих випадків, коли пристрій працює з переносним ПК з автономним живленням, його не можна підключати безпосередньо до ланцюгів, що не ізолюються від мережі або знаходяться під високим потенціалом.

Додавання декількох оптронів до даної схеми допомогло вирішити цю проблему, хоча і привело до незначного погіршення характеристик інтерфейсу.

Як і в параметри не має оптрон, в ньому завжди використовується оптичний принцип передачі сигналів без всякого гальванічного зв'язку. При цьому і джерело, і приймач оптичного випромінювання розміщені в єдиному корпусі. Для передачі випромінювання від джерела до приймача використовуються різні технології, опис яких виходить за рамки даного проекту. Особливості цих технологій визначають основні характеристики оптрона, забезпечуючи, зокрема, необхідний компроміс між ступенем ізоляції, коефіцієнтом передачі і швидкістю.

В більшості випадків джерелом випромінювання служить світлодіод з арсеніду галію, що працює в ближній інфрачервоній області спектру. При цьому спектр його випромінювання практично збігається з областю максимальної спектральної чутливості кремнієвих фотоприймачів, найбільш дешевих і широко поширених. У спеціальних оптронах можуть використовуватися і інші випромінювачі, такі як мікаторні лампи розжарювання або газорозрядні світловипромінювачі прилади, наприклад

на окоці. В якості фотоприймачів можна часто зустріти фоторезистори, ж, наприклад, в оптопарах – “лампа розжарювання – фоторезистор”, що широко застосовувалися років двадцять тому. Ці оптопари використовувалися в звукоєкстрактах в якості потенціометрів з електронним управлінням.

Найпоширеніші оптрони випускаються в корпусі DIP6 і будуються за схемою, відповідно до якої, інфрачервоний світлодіод оптично пов'язаний з фототранзистором, три виводи якого є виходами оптрона. Прийняте розташування виводів забезпечує максимальну відстань між входом і виходом. Очевидно, що при такій конструкції немає необхідності робити оптрон з воляцією, що витримує напругу 3000 В, якщо напруга пробов між контактними майданчиками або друкованими провідниками складає всього біля 500 В.

Вивід бази зазвичай не використовується, оскільки транзистор переходить в провідний стан в результаті фотоелектричного ефекту. Проте іноді зустрічаються схеми, де між базою і емітером включений резистор. Забезпечувачи швидке розсмоктування накопиченого в базі заряду, таке схемне рішення помітно покращує час спрацювання оптрона, але, на жаль, за рахунок зниження його коефіцієнта посилення по струму або коефіцієнта передачі. Треба відмітити, що цей резистор зменшує також і зворотний струм колектора.

В деяких випадках вивід бази можна використовувати для управління транзистором незалежно від стану світлодіода, але при цьому треба стежити, щоб не порушилися ізоляційні властивості оптрона.

Такий поширений оптрон, як ТЛ 111 (аналоги МСТ 2, Н11 А2 і т.п.) має напругу ізоляції 1500 В, смугу пропускання 300 кГц і коефіцієнт передачі струму близько 8%. Це означає, що при силі струму світлодіода 10 мА сила струму фототранзистора складатиме не більше 800 мкА.

Оптрон 4N28 при напрузі ізоляції 500 В має коефіцієнт передачі струму 10%, тоді як 4N25 має такий же коефіцієнт при напрузі ізоляції 2500 В.

В моделей типу SL 5500 (спеціальний телефонний оптрон) коефіцієнт передачі струму може складати до 40%, а напруга ізоляції дорівнює 3500 В при постійному струмі або 2500 В (ефективна) при змінному. Оптрон SL 5501, ціна якого ледве нижча, має коефіцієнт передачі струму не більше 15%.

Що стосується оптрона CNY 17-2, широко вживаного в телефонії, то його коефіцієнт передачі досягає 80%, напруга ізоляції складає 4400 В, а швидкість смуги пропускання – більше 1 МГц. Деякі оптрони, що випускаються в корпусах DIP8, використовують фотодіод, сполучений з транзистором, не чутливим до випромінювання, а необхідним лише для посилення струму фотодіода. Оскільки для правильної роботи на фотодіод має бути подана зворотна напруга зсуву, що створює відповідний зворотний струм, оптрони такого типу мають додатковий вивід для його підключення, що позначається VCC.

Таким чином можна отримати оптрони з досить високим коефіцієнтом передачі і з винятковою швидкодією 11 МГц для елементу CNW 136 компанії Hewlett Packard. Оптрон HPCL 4562 того ж виробника, спеціально призначений для передачі аналогових сигналів, має смугу пропускання 17 МГц при коефіцієнті передачі струму 200%.

Але найпопулярнішим рішенням для радикального збільшення коефіцієнта передачі струму є вживання складеного фототранзистора, побудованого по схемі Дарлінгтона. Це рішення використовується при виготовленні широко поширеного оптрона 4N33, що має високий коефіцієнт передачі струму – 500%, але смугу пропускання лише 30 кГц.

Більш швидкодіючий оптрон CNW 139 (виробник – Hewlett Packard) має рекордний коефіцієнт передачі 3000%, тобто посилення в 30 разів.

У схемі інтерфейсу, представлений на рис. 3.2, використано три найдешевші оптрони 4N33, але для поліпшення параметрів цього АЦП можна застосовувати якісніші і дорожчі компоненти. Вживання оптронів з великим коефіцієнтом передачі дозволяє забезпечити простоту рішень схемотехніки, досягнення якої є однією з цілей даного проекту.

					РП 05.08.003.00 ДП ПЗ	Арх.
Змк.	Арх.	Наблюд.	Голов.	Діагн.		39

У серйозніших проектах між оптронами і АЦП напевно виключатимуться логічні схеми з тригерами Шмітта, використовуваними в якості формувачів для швидкозмінюючихся сигналів. Малогабаритні послідовні АЦП здатні забезпечити великі вихідні струми, що дозволяє підключати світлодіод оптрона безпосередньо до їх виходів через резистор опором всього 2,7 кОм. Цього вистачає, щоб фототранзистор оптрона нормально працював з вихідними лініями USB-порту комп'ютера, отримуючи напругу живлення від лінії +5V через такий же резистор.

Для захисту від негативної напруги, що з'являється на виводах USB-порту комп'ютера і також впливає на світлодіоди через резистор опором 2,7 кОм, у схему додані діоди 1N4148.

Фототранзистори оптронів OP1 і OP2 формують управляючі сигнали, які впливають на входи АЦП, сполучені із спільним дротом резисторами опором 82 Ом. Такий номінал, що незначно малий і викликає споживання великої потужності, необхідний для підвищення крутості фронтів управляючих сигналів.

Таким чином, гальванічна розв'язка цілком забезпечена з боку цифрової частини. У якості стабілізатора вибраний компонент LM 2931, що має дуже мале падіння напруги на регулюючому елементі, що дозволяє обійтися джерелом живлення 5 В. Якщо ж є упевненість, що зовнішня напруга живлення буде завжди більше 5 В, то можна застосувати звичайний стабілізатор 78L05, зменшивши ємність конденсатора на виході стабілізатора до 10 мкФ.

При використанні зовнішнього джерела живлення вимоги до енергоспоживання стають менш жорсткими, тому джерело опорної напруги REF 25 Z може бути замінено на менш економічне, але дешевше і при цьому має точність 0,2%, а не 1% (наприклад, LT 1009 CZ виробництва компанії Linear Technology).

3.2 Розробка принципової електричної схеми вхідного підсилювача

За допомогою невеликого пристрою, який буде описано нижче, легко знімаються два основні обмеження, властиві простим АЦП: вхідний діапазон від 0 до +5 В і нездатність працювати із змінною напругою без постійної складової. Наприклад, пристрій зсуву нуля і підсилювач з коефіцієнтами посилення – 1, 10 і навіть 100 спільно розширить сферу застосування АЦП ADC 10 і того потужного програмного забезпечення, яке до нього додається. Цей АЦП можна буде використовувати навіть для оцифрування звуку.

Подача змінної напруги з нульовим середнім значенням (іншими словами, без постійної складової) на вхід аналого-цифрового перетворювача з вхідним діапазоном 0-5 В приводить до ефекту однопілітеріодного випрямлення. Результати вимірювань, виконаних без врахування цієї обставини, значайно, виявляться невірними. Дана ситуація аналогічна тій, коли промінь зенітного оцифрографа розташований в нижньому краї екрану, а не в середині, або коли перо реєстратора встановлене на один з країв паперу. У обох випадках регулювання положення променя або установки нуля дозволяє вирішити проблему шляхом додавання регульованої постійної напруги зсуву до вхідного сигналу.

Хоча вхідний діапазон 0-5 В і підходить для вирішення багатьох завдань, при вимірюваннях за допомогою 8-розрядного АЦП бажано використовувати максимально можливу кількість з 256 рівнів, тобто потрібно укластися в останню третину шкали. Наприклад, вхідна напруга, що змінюється в межах від 0 до 500 мВ, при перетворенні займе всього 25 рівнів, що відповідає точності 4%, тоді як АЦП напевно має точність не нижче 1%. 1-2-розрядні АЦП вирішують цю проблему, але вони теж не працюють з сигналами змінного струму.

Вхідний опір більшості мікросхем АЦП значно менше 1 МОм, і через це звичайний дільник 1/10 буде дуже сильно притікувати сигнали, амплітуда яких лише дещо більше рівня 5 В.

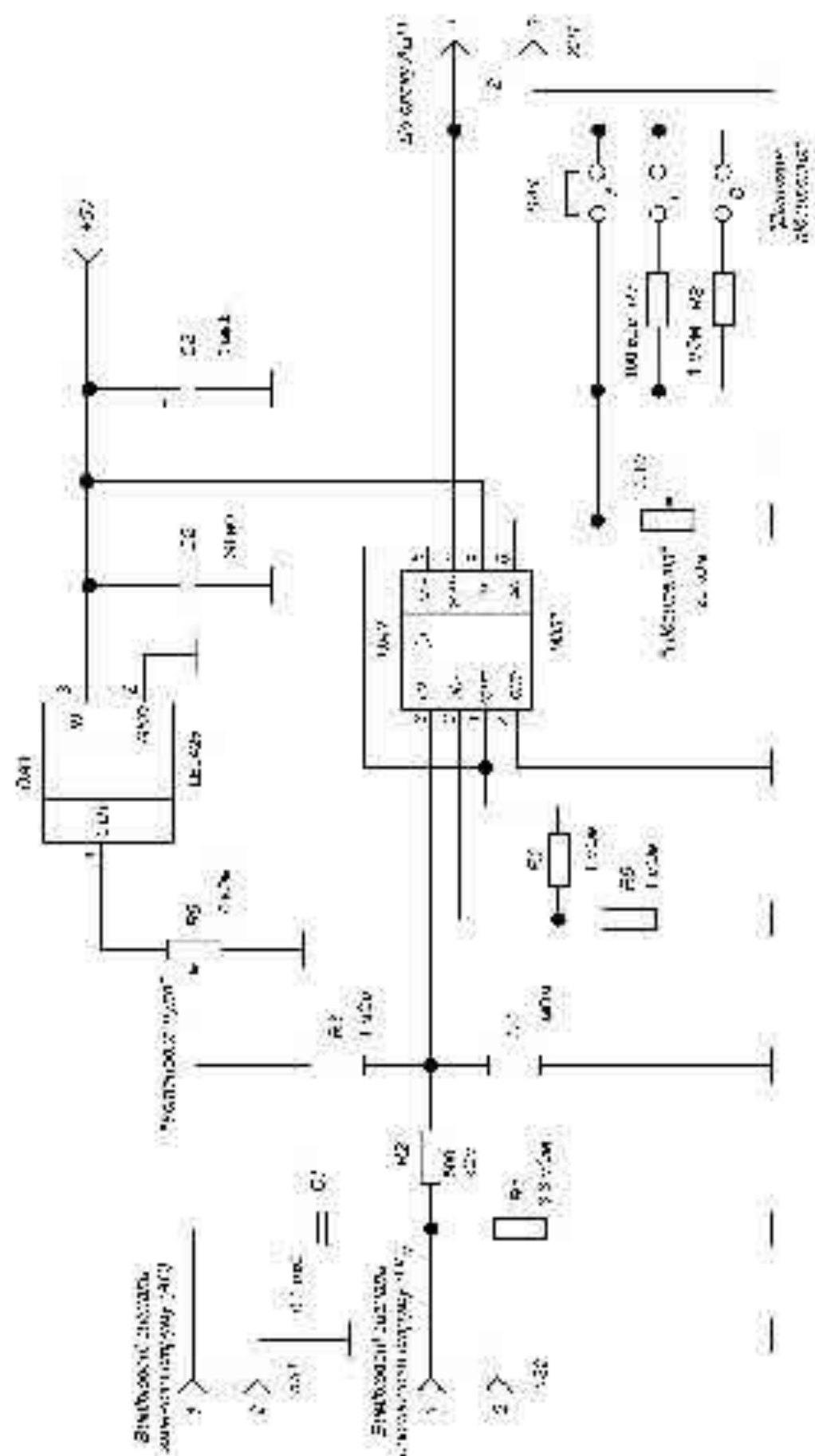


Рисунок 3.3. Принципова електрична схема вхідного підсилювача USB-частотоміру

Змк.	Арх.	Наблюд.	Підпис	Дата

РП 05.08.003.00 ДП ПЗ

Арх.

42

Попередній підсилювач з коефіцієнтом посилення 10 дозволить вимірювати напругу до 500 мВ з такою ж точністю, з якою сам АЦП оцифровує напругу до 5 В. При вхідному опорі 1 МОм попередній підсилювач з одиничним посиленням дозволить працювати на межі вимірювання «50 В» з використанням звичайного дільника 1/10.

З урахуванням простоти схем описуваних АЦП був розроблений найпростіший підсилювач. Дійсно, не потрібні ні широка смуга (вистачає декількох кілогерц), ні висока точність (достатньо 1%), ні зсув нуля на рівні мікрвольт, оскільки використовується 10-розрядний АЦП. Не треба розглядати схему, приведену на рис. 3.3, як зразок високоточного приладу – це не великий, але корисний пристрій, що надає чималі можливості.

Головний компонент схеми – зведений операційний підсилювач LM358, розроблений дуже давно і широко поширений. Він може працювати з напругою, дуже близькою до нуля, навіть без двополярного живлення.

З напруги живлення +5 В від USB-порту мікросхеми TLE2425 виробництва компанії Texas Instruments формується стабільна напруга 2,5 В. Звичайно ця мікросхема використовується в пристроях для отримання «віртуальної землі». Хоча напруга 2,5 В навіть при одиничному посиленні цілком достатня для зсуву нуля в середину робочого діапазону 0-5 В, за відсутності мікросхеми TLE2425 допустиме вживання інтегрального стабілізатора 78L05. При цьому параметри будуть менш стабільні, а споживання струму збільшиться. Розташування виводів у обох мікросхемах однакове:

Багатооборотний потенціометр R9 з опором 10 кОм служить для подачі частини вихідної напруги мікросхеми TLE2425 на резисторний суматор. На цей же суматор поступає і вхідна напруга, яка може подаватися безпосередньо (роз'єм DC – відкритий вхід) або через розділовий конденсатор C_{in} (роз'єм AG – закритий вхід), як і в звичайних осцилографіях.

Використання входу AC дозволяє прибрати з вхідного сигналу постійну складову (зазвичай присутню), а потім за допомогою потенціометра R9

					РП 05.08.003.00 ДП ПЗ	Арх.
						43
Змк.	Арх.	Наблюд.	Підпис	Дата		

підібрати таку її величину, яка максимально спростить проведення вимірювань. Ємкість неелектролітичного конденсатора C_{01} (бажано, щоб він був розрахований на напругу 400 В) залежатиме від умов вимірювань: величина 1 мкФ дозволє працювати на дуже низьких частотах, а величини 0,1 мкФ достатньо для роботи в звуковому діапазоні.

Номинали резисторів (з точністю 1%) були розраховані так, щоб перший каскад мав одиничне посилення, а вхідний опір складав 1 МОм. Резистор з опором 500 кОм можна отримати, з'єднавши паралельно два резистори по 1 МОм кожен.

Другий каскад має коефіцієнт посилення, що переміщується – 1, 10, 100, а багатооборотний потенціометр R10 22 кОм дозволє або точно відкалібрувати посилення при налаштуванні, або регулювати його плавно – так само, як це можна виконати за допомогою звичайного осцилографа, обертаючи ручку «Плавне посилення». Звичайно, ніщо не заважає використовувати інші номінали резисторів, щоб отримати інші коефіцієнти посилення: 2, 5 і так далі.

Для реалізації коефіцієнта посилення 100 в ланцюзі зворотного зв'язку операційного підсилювача використовується резистор з номіналом 1 МОм, для коефіцієнта посилення 10 – паралельне з'єднання резисторів 100 кОм і 1 МОм. Для реалізації одиничного посилення (схема повторювача) застосована звичайна перемінка. Щоб забезпечити високу точність пристрою, калібрування треба проводити при коефіцієнті посилення 10, оскільки режим «x100» використовується рідко і, крім того, в цьому режимі операційний підсилювач LM 358 схильний до впливу ефекту зсуву нуля (який навед чи компенсуватиметься поеніть).

Слід звернути увагу на те, що ненавмисна подача дуже великої напруги на вхід підсилювача або самого АЦП не може привести до його попадання на материнську плату ПК, тому що розроблена схема має оптронуву розв'язку. Але все одно, описаний пристрій не можна розглядати як захисний: він не призначений для цієї мети. Користувач при проведенні вимірювань сам повинен подбати про всі необхідні заходи обережності.

4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ USB-ЧАСТОТОМІРУ

У даному розділі буде описаний покроковий процес створення програмного забезпечення приладу USB-частотоміру. Розширення можливостей даного приладу передбачає створення окремих програмних модулів для інших типів вимірювань та відповідних пристроїв-приставок до розробленого в попередньому розділі модуля аналогового інтерфейсу. Далі наводиться опис процесу створення і роботи саме програмного забезпечення віртуального приладу USB-частотоміру, використання якого описане у Розділі 1. Особливістю створюваного програмного забезпечення є те, що для роботи із зовнішніми інтерфейсами ПК застосовуються виключно функції Windows API. Це означає, окрім іншого, що на відміну від подібних програм, описаних у [1] і [2] запропонована програма буде однаково добре працювати у різних операційних системах Microsoft сімейства Windows. Інтерфейс програми імітує шкалу цифрового приладу і по ходу розробки буде наведено повний його опис.

4.1 Склад технічних і програмних засобів для розробки

4.1.1 Технічні засоби розробки

При виборі технічних засобів для розробки віртуального вимірювального приладу – частотоміру найбільшу роль грає чинник швидкодії роботи ПК, оскільки саме від нього залежить час розробки ПЗ, а відповідно витрати на розробку і його собівартість.

Швидкість функціонування додатків на ПК в основному визначається наступними параметрами:

- Об'ємом оперативної пам'яті (ОП);
- Швидкодією процесора;
- Об'ємом відеопам'яті (ВП).

Виходячи з вимог, що пред'являються до використовуваних програмних засобів розробки (Embarcadero RAD Studio), мінімальні значення

					РП 05.08.004.00 ДП ПЗ	Арх.
Змк.	Арх.	Наблюд.	Голов.	Датум		45

випередильні параметри складають: ОП – 2 Гб, процесор – з частотою 2 ГГц, ВП – 128 Мб, ОС – Windows 7-10.

4.1.2 Середовище розробки ІЗ

Embarcadero RAD Studio – середовище швидкої розробки додатків (RAD) фірми Embarcadero Technologies, яке працює під ОС Windows. Поточна версія Embarcadero RAD Studio 10.3 Rio об'єднує Delphi і C++ Builder у єдине інтегроване середовище розробки:

- RAD Studio Professional – підходить для індивідуальних розробників і невеликих груп, що створюють настільні і мобільні додатки;
- RAD Studio Enterprise – підходить для груп розробників, що створюють клієнт-серверні або багаторізні ві додатки;
- RAD Studio Architect – підходить для корпоративних клієнтів, що створюють бази даних або веб-додатки.

Для виконання розробки програмного продукту віртуального частотоміру найбільш доцільним є використання саме версії RAD Studio Professional.

Embarcadero RAD Studio представляє собою набір засобів розробки додатків, який дозволяє створювати додатки з графічним призначенням для користувача інтерфейсом для Windows, Mac OS X, .NET, PHP та веб-рішень. До складу входять:

- Embarcadero Delphi – дає можливість створювати повнофункціональні додатки для Windows і Mac OS X;
- Embarcadero C++ Builder – це середовище C++, яке повністю відповідає концепції швидкої розробки додатків, об'єднує засоби ANSI C++ і багатофункціональну розширювану інфраструктуру візуальних компонентів;
- Embarcadero Prism – являє собою крос-платформне рішення для розробки і Delphi-подібну мову програмування для швидкої розробки додатків .NET, Mono, ASP.NET і додатків, створюваних у рамках парадигми Data Driven Design (орієнтованих на роботу з визначеними

наборами даних) для Windows, Linux і Mac OS X;

- Embarcadero RadPHP – спрощує створення веб-додатків на PHP завдяки наявності візуальних засобів проектування інтерфейсів, редактора, відладчика, засобів підключення до баз даних і інтегрованої бібліотеки повторно використовуваних класів компонентів. Компоненти RadPHP дозволяють робити веб-інтерфейси у стилі iOS і Android;
- ER / Studio Developer Edition – допомагає проектувальникам баз даних аналізувати, документувати і повторно використовувати дані і надає засоби зворотного проектування, аналізу та оптимізації баз даних;
- InterBase SMP Developer Edition – надає розробникам крос-платформну базу даних для створення і тестування для збудованих додатків і додатків для малих і середніх за розмірами підприємств.

RAD Studio включає в себе широкій набір додаткових програм:

- InstallAware Express – надає засоби, що дозволяють користувачам, які не мають навичок програмування і розробки сценаріїв, створювати складні настановні пакети;
- Rave Reports компанії Newton – набір рішень для створення звітів;
- FastReport;
- FireMonkey;
- App Wave;
- TeeChart Standard компанії Zeemla – компоненти для створення діаграм;
- VCL для веб-рішень (IntraWeb) компанії Atzed Software – платформа веб-додатків RAD;
- FinalBuilder Embarcadero Edition – служить для автоматизації процесу складання;
- Code Site Express – засоби ведення журналу для збірки програм;
- AQTime Standard компанії SmartBear – створення профілів продуктивності;
- Beyond Compare Text Compare – порівняння файлів вихідного коду;

					РП 05.08.004.00 ДП ПЗ	Арх.
						47
Зач.	Арх.	Наблюд.	Підпис	Дата		

- RemObjects Internet Tools і Obfuscator – додаткова функціональність для веб-розробки і обфускації коду у Delphi Prism.

Основні нові можливості Rad Studio 10 Seattle є такими.

Розробники на Delphi і C++ Builder можуть швидко відновити свої VCL- і FMX-додатки і в повній мірі скористатися засобами ОС Windows 10. Підтримуються компоненти Windows 10 і «рідні» API і компоненти WinRT / UWP, елементи інтерфейсу Windows 10 VCL (рис. 4.1). Також оновлена підтримка Windows 10 FMX.

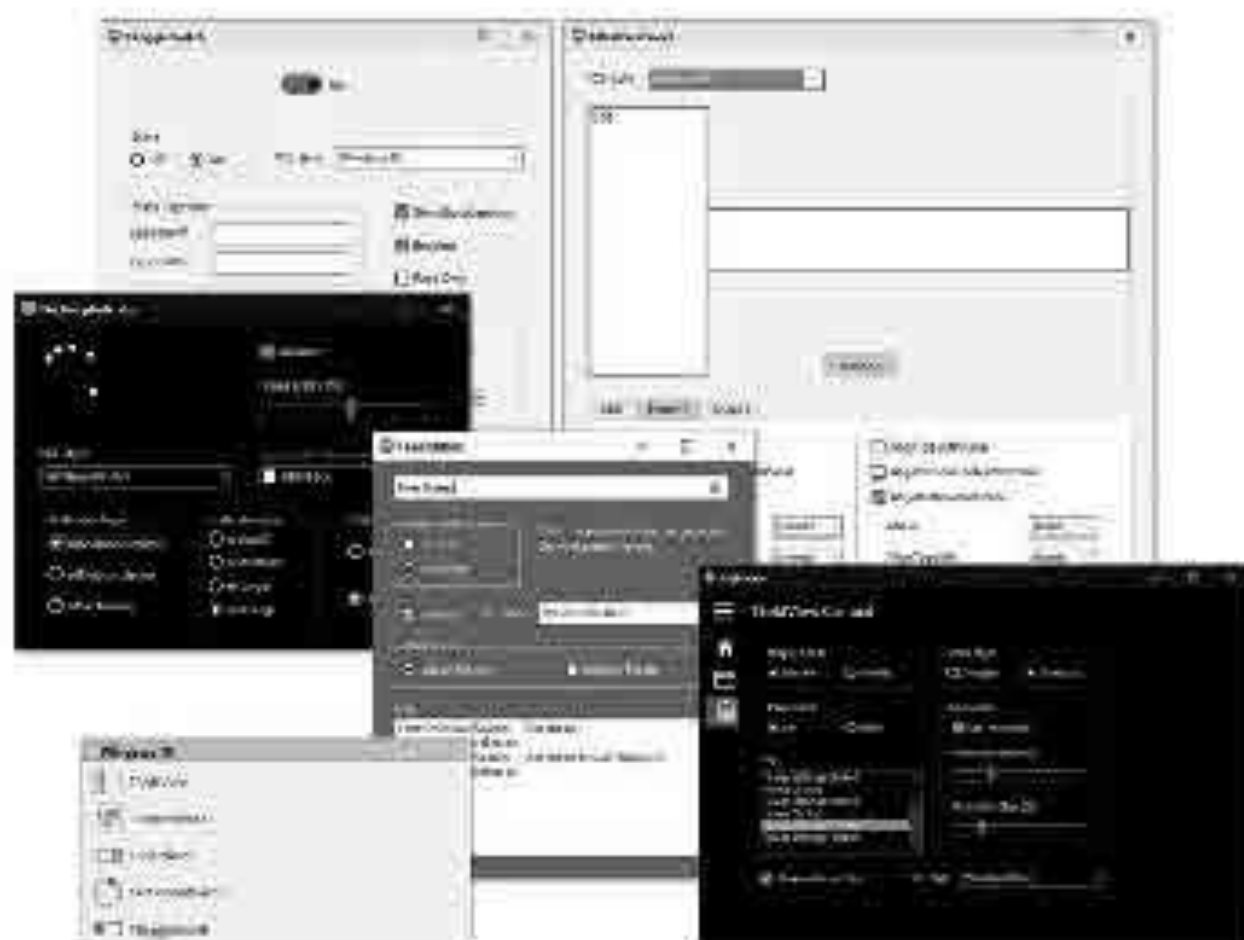


Рисунок 4.1. Елементи інтерфейсу Windows 10 VCL у RAD Studio

У C++ Builder 10 з'явився перший у світі компілятор C++ на основі CLANG для Windows і мобільних платформ з розширеннями RAD PME, що забезпечують швидку розробку для Windows і інших платформ. Він підтримує тісну інтеграцію з VCL для Windows і кросплатформними структурами FMX, мова C++ 11 і керування пам'яттю на основі ARC (автоматичного підрахунку

посилань) для C++, а також має зворотню сумісність. Новий компілятор C++ Builder (рис. 4.2) робить версію RAD Studio 10 обов'язковим оновленням для розробників на C++ і нових розробників, які переходять на C++ з інших мов і наборів інструментів – Java, Objective-C, C#, Xcode, Xamarin і Visual Studio.



Рисунок 4.2. Налаштування компілятора і проєкту C++Builder у RAD Studio



Рисунок 4.3. Інтеграція засобів розробки RAD Studio для різних платформ

Зміст	Арх.	На обліку	ГЛІС	Дата

РП 05.08.004.00 ДП ПЗ

Арх.

49

З непомітних для користувача змін можна відзначити перероблену у даній версії архітектуру системи керування продуктами та збірками, яка дозволила фактично подвоїти підтримуваний розмір проєктів і підняти стабільність і продуктивність при роботі з великими проєктами, особливо для декількох платформ (рис. 4.3). Оновлення до цієї версії стане в нагоді розробникам, які стикалися з обмеженнями ресурсів і іншими проблемами в великих проєктах.

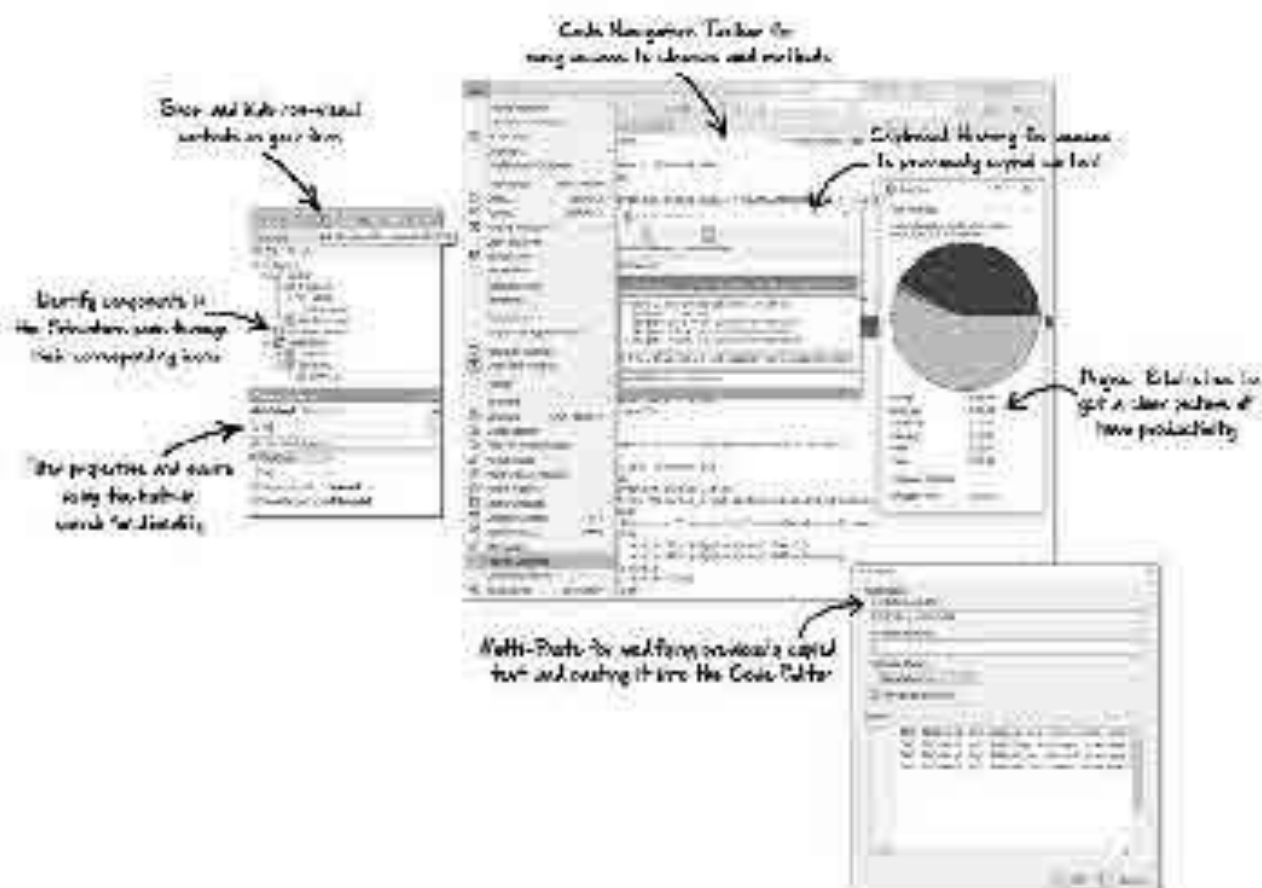


Рисунок 4.4. Ілюстрація можливостей системи документації у RAD Studio

Процедури розробки, тестування та складання документації були серйозно доопрацьовані (рис. 4.4). Тепер вони краще підходять для створення додатків на декількох мовах для платформ, підтримуваних новою версією RAD Studio.

Інші нові можливості Rad Studio (10.2 Tokyo та 10.3 Rio):

- підтримка паралельної компіляції C++;
- підтримка C++ 17 у C++ Builder для 64-бітної версії Windows;

Змк.	Арх.	Наблюд.	Підпис	Дата

РП 05.08.004.00 ДП ПЗ

- підтримка розробки додатків для iOS, Mac OS, Android, Linux;
- підтримка модульного тестування DUnitX для Android і iOS;
- підтримка DirectX 12;
- підтримка виклику API WinRT;
- підтримка FireDAC для бази даних NoSQL MongoDB;
- нова поведінка MultiView;
- виключена підтримка СУБД MariaDB.



Рисунок 4.5. Ілюстрація можливостей кросплатформної розробки у RAD Studio

4.2 Створення інтерфейсу користувача ПЗ USB-частотомір у

Після розміщення елементів інтерфейсу і налаштування всіх властивостей форма проекту RAD Studio Delphi буде виглядати так, як показано на рис. 4.6.

Збережемо проект і виконаємо його компіляцію натисненням клавіш Ctrl – F9. У папці D:\MySource буде створений виконуваний модуль програми з ім'ям ComPro.exe. Можна запустити його на виконання з середовища RAD Studio,

натиснувши клавішу F9. Спроби отримати реакцію на натиснення клавіш 'Старт', 'Закінчення' і 'Вихід' виявлялися безрезультативними, оскільки ми ще не додали програмі жодної функціональності. Закриємо вікно програми, натиснувши по квадрату у його правому верхньому куті.

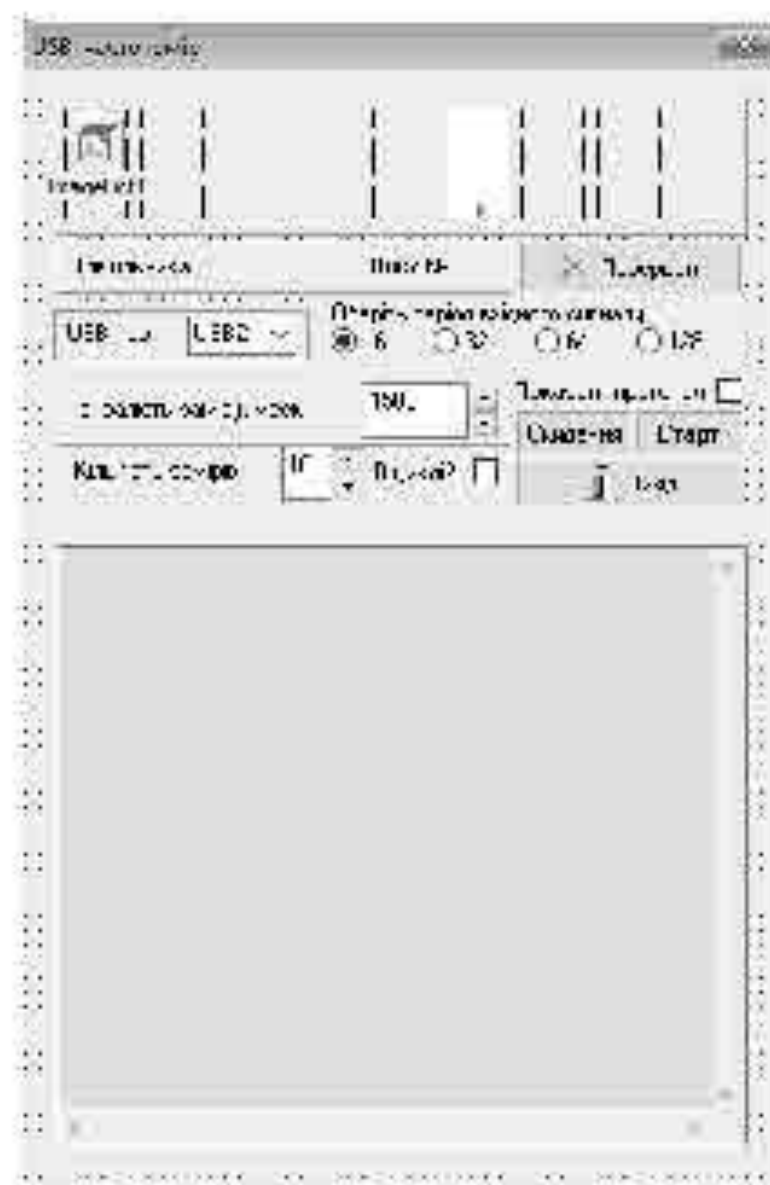


Рисунок 4.6. Зовнішній вигляд інтерфейсу застосунок
USB-частотоміру на етапі розробки

4.3 Розробка алгоритмів і створення обробників подій

Проектування призначеного для користувача інтерфейсу – це тільки частина роботи по створенню повноцінної програми. Друга частина полягає у тому, щоб змусити програму працювати по певному алгоритму (сценарію).

Розглянемо спочатку ж зробити так, щоб по клацанні на кнопці 'Вихід' закривалося вікно програми. Виділимо на формі кнопку BitBtn1 і натиснемо клавішу F11. У вікні інспектора об'єктів перейдемо на закладку Events. Двічі клацнемо лівою кнопкою миші по порожньому рядку правіше за напис onClick. Delphi автоматично створить текст процедури, що представляє собою реакцію на клацання мишею по кнопці 'Вихід', і виконає перехід у редактор програмного тексту. Там ми впишемо у тіло процедури єдину строчку: `Form1.Close;`. Тоді загальний текст даного обробника події буде таким:

```
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  Form1.Close;
end;
```

Повторно відкомпілюємо проєкт і запуснемо його на виконання. Переконаємося, що по клацанні на кнопці 'Вихід' вікно програми закривається і вона припиняє роботу. Аналогічним чином опишемо обробник події onClick для кнопки 'Очищення':

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  //очистити вікно протоколу:
  Memo1.Lines.Clear;
  //обнулити цифрову шкалу:
  OutDigitDisplay(0);
end;
```

Рядок `Memo1.Lines.Clear` – це процедура очищення протоколу. Описувати її немає необхідності, оскільки вона є методом стандартного об'єкту Delphi і вже описана в його бібліотеках. А ось процедура `OutDigitDisplay` – призначена для користувача. Це означає, що програміст створює її самостійно, починаючи від привласнення імені і закінчуючи описом тіла процедури. Ми зупинимося на цій процедурі докладніше, коли розглядатимемо програмування цифрової шкали. А зараз просто створимо для неї порожню заготовку, або 'заглушку'. У редакторі програмного тексту перейдемо в розділ `Published` об'єкту `TForm` і впишемо рядок:

```
procedure OutDigitDisplay(Value: Double);
```

Після цього перейдемо у розділ Implementation і опишемо "заглушку":

```
procedure TForm1.OutDigitDisplay(Value: Double);
begin
  {тут ми описуємо процедуру}
end;
```

Звернемо увагу, що процедура має один вхідний параметр; логічно передбачити, що цей параметр містить число, яке має відображатися на цифровій шкалі.

Збережемо проект і відкомпілюємо його. Так само створимо процедуру 'заглушку', що викликається по кліцанню на кнопці "Старт":

```
procedure TForm1.Button2Click(Sender: TObject);
begin
end;
```

Процедуру, пов'язану з кнопкою 'Перервати', опишемо повністю:

```
procedure TForm1.BitBtn2Click(Sender: TObject);
begin
  if MessageDlg('Перервати вимір?',
    mtConfirmation [mbNo, mbYes], 0) =
    mrYes then begin {перервати вимір}
    CheckBreak := True;
    DisableEnableVisual(True);
  end;
end;
```

Ця процедура спочатку видає запит на підтвердження переривання вимірювань. Якщо оператор підтверджує переривання, процедура призначає глобальній змінній CheckBreak значення 'істина', що викликає вихід з процедури вимірювання на будь-якому її кроці. Процедура DisableEnableVisual призначена для керування доступністю елементів інтерфейсу у відповідності з логікою програми. Ось її код:

```
procedure TForm1.DisableEnableVisual(B: boolean);
begin
  Button1.Enabled := B;
```

```

    Button2.Enabled := B;
    BitBtn1.Enabled := B;
    BitBtn2.Visible := not B;
    UpDown1.Enabled := B;
    UpDown2.Enabled := B;
    Edit1.Enabled := B;
    Edit2.Enabled := B;
    Memo1.Enabled := B;
    RadioGroup1.Enabled := B;
    ComboBox1.Enabled := B;
    CheckBox2.Enabled := B;
    Panel2.Visible := not B;
    Form1.Refresh;
end;

```

4.3.1 Реалізація процедури вимірювання вхідних сигналів

Тепер треба приступити до реалізації тієї частини програми, яку програмісти називають математичною. Вона виконує дії, направлені на реалізацію основного завдання – вимірювання частоти вхідного сигналу. По ходу програмування основного завдання програміст має забезпечити виведення розрахункових даних на пристрої відображення і (або) довготривалого зберігання. У нашому випадку пристроїми відображення є цифрова шкала і протокол вимірювання. Пристрій зберігання у даному вигляді відсутній. Проте, використовуючи можливість копіювання вмісту протоколу у буфер обміну, оператор може зберегти його у текстовому файлі, який, у свою чергу, і буде зв'язатися з пристроєм довготривалого зберігання.

Спрощений алгоритм отримання даних вимірюваних сигналів з USB-порту показаний на рис. 4.7.

Програма починає виконувати вимірювання частоти вхідного сигналу по кліцанню на кнопку 'Старт'. При цьому викликається процедура TForm1.Button2Click, 'заглушка' для якої описана вище. Зараз ми маємо вписати в тіло процедури конкретний код:

```

procedure TForm1.Button2Click(Sender: TObject);
begin
    Ізпро всіх випадок закриваємо порт

```



```

if FHandle <> invalid_handle_value then CloseHandle(FHandle);
// намагаємось відкрити порт
WorkPort := ComboBox1.Items.Strings[ComboBox1.ItemIndex];
// ім'я порту (USB1, USB2...)
StBaud := '1200'; // швидкість передачі
FHandle := CreateFile( // функція відкриття порту
    PChar(WorkPort), // ім'я порту
    generic_read, // відкриваємо порт тільки для читання
    0, // позначка нерозділюваного ресурсу
    NI, // режим безпеки - ігноруємо
    open_existing, // перевірка існування порту
    file_flag_overlapped, // дозволити режим накладання операцій
    0 // шаблон файлу, для USB-порту завжди 0
);
if FHandle = invalid_handle_value then begin // помилка відкриття порту
    MessageDlg('Порт ' + WorkPort + ' відкрити не вдалося!' + 'М' +
        'Або цей порт фізично відсутній' + 'М' +
        'або він зайнятий! Оберіть інший порт', mtError, [mbCancel], 0);
    Exit;
end;
// з цього місця ми при не вдалому відкритті порту:
// залежності від стану перемикача вхідних ліній
// визначимо потрібний біт:
if RadioGroup1.ItemIndex = 0 then LineByte := 32;
if RadioGroup1.ItemIndex = 1 then LineByte := 16;
if RadioGroup1.ItemIndex = 2 then LineByte := 64;
if RadioGroup1.ItemIndex = 3 then LineByte := 128;
// визиваємо функцію знаходження частоти:
FindFreq(FHandle, LineByte);
CloseHandle(FHandle); // закриваємо порт
end;

```

Текст процедури ретельно прокоментований, але розглянемо детальніше, які дії вона виконує. Спочатку аналізується дескриптор USB-порту на допустимі значення. Якщо умова виконується, то порт вже був відкритий і його задалегідь слід закрити. На наступному кроці змінній WorkPort призначається значення імені USB-порту з поля із списком ComboBox1. Далі викликається функція відкриття USB-порту, що повертає його дескриптор, і відразу ж виконується перевірка дескриптора на неприпустимі значення. Якщо умова виконується, видається повідомлення про помилку відкриття порту і проводиться вихід з процедури.

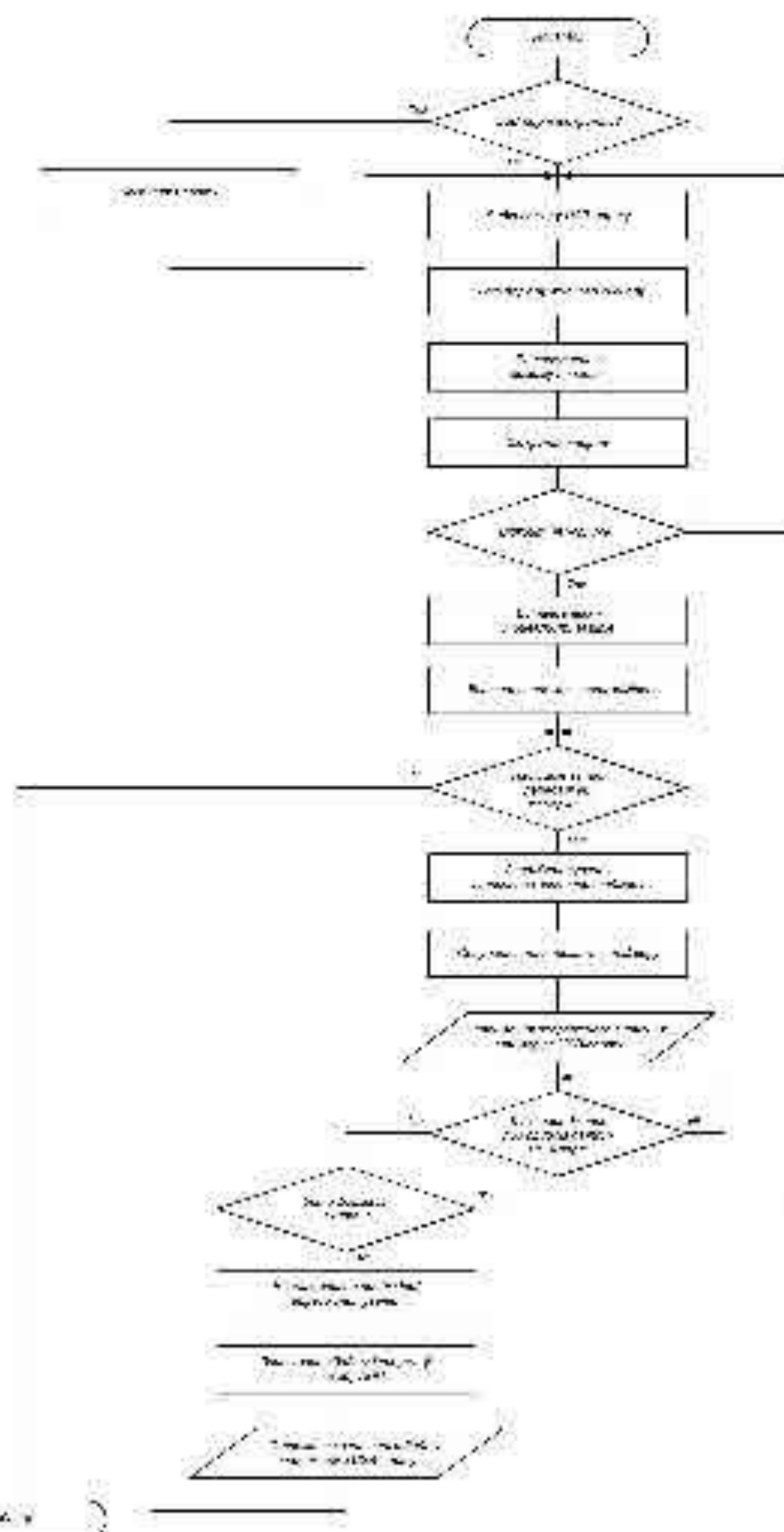


Рисунок 4.7. Алгоритм отримання даних вимірюваних сигналів з USB-порту

При вдалому відкритті USB-порту змінна LineByte ініціалізується значенням, відповідним обраній вихідній лінії. Потім викликається функція знаходження частоти FindFrqz, якій в якості параметрів передаються дескриптор USB-порту FHandle і змінна LineByte. Після того, як функція відпрацює, USB-порт закривається.

4.3.2 Реалізація функції знаходження частоти

Реалізація функції FindFrqz заснована на одному з наріжних каменів програмування – циклах. Далі наведено повний текст цієї функції з детальними коментарями.

```
function TForm1.FindFrqz(Th: THandle; Bit: Word): real;
(*
  Знайти період вхідного цифрового сигналу.
  Вхідний сигнал спеціфікується дескриптором порту (Th) і
  бітом вхідної лінії (Bit)
+-----+
+-----+
+-----+
+-----+
+-----+
+-----+
+-----+
+-----+
+-----+
*)
var
  Counter, MainCicle, Cicle: Integer;
  Time1, Time2, Count: int64;
  Average, Frqz, fMax, fMin: Extended;
begin
  Result := -1; (* значення за замовчуванням *)
  MainCicle := 1;
  repeat (* початок зовнішнього циклу *)
    if Memo1.Lines.Count > MemoCount then begin
      if MessageDlg('Протокол переповнено.' + ^M +
        'Продовжити(протокол буде очищено)?',
        mtConfirmation, [mbNo, mbYes], 0) <= mrYes then Break;
      Memo1.Lines.Clear;
    end;
    DisableEnableVisual(False); (* забороняємо видимі елементи *)
```

```

Average := 0; (*змінна для підрахунку середнього значення*)
Cicle := 1; (*номер заміру *)
Counter := StrToInt(Edit1.Text); (*тривалість заміру у мсек *)
//Timer1.Interval := StrToInt(Edit1.Text); (*тривалість заміру у мсек *)
Label8.Caption := IntToStr(MainCicle);
Panel2.Refresh;
Memo1.Lines.Add('Цикл № ' + IntToStr(MainCicle));
Memo1.Lines.Add('Старт : ' + TimeToStr(Time));
Memo1.Lines.Add('Тривалість заміру в мсек: ' + IntToStr(Counter));
Memo1.Lines.Add('Кількість замірів: ' + Edit2.Text);
if CheckBox2.Checked then
  Memo1.Lines.Add('*** Безперервний цикл ***');
Memo1.Lines.Add('-----');
FMin := 100000000000; (*для знаходження мінімальної частоти*)
FMax := 0; (*для знаходження максимальної частоти*)
CheckBreak := False; (*для відстежування виходу з циклу*)
repeat (*зовнішній цикл *)
  Count := 0; (*скидання лічильника вимірюваних імпульсів *)
  //перед початком підрахунку пропустити один такт,
  //щоб почати відлік точно з початку напівперіоду
  //нахростання сигналу:
  repeat (*зачекати високий рівень *)
    Application.ProcessMessages;
  until (GetModemStatus(Th) and Bit = Bit) or CheckBreak;
  repeat (*зачекати низький рівень *)
    Application.ProcessMessages;
  until (GetModemStatus(Th) and Bit = 0) or CheckBreak;
  QueryPerformanceCounter(Time1);
  (*початок відліку: запам'ятати стан лічильника *)
  //Timer1.Enabled := True;
  //початок відліку:
  Repeat
    repeat (*зачекати високий рівень *)
      Application.ProcessMessages;
    until (GetModemStatus(Th) and Bit = Bit) or CheckBreak;
    repeat (*зачекати низький рівень *)
      Application.ProcessMessages;
    until (GetModemStatus(Th) and Bit = 0) or CheckBreak;
    inc(Count); (*інкрементувати лічильник кількості періодів сигналу *)
    QueryPerformanceCounter(Time2);
    (*зчитати поточний стан лічильника *)
    Application.ProcessMessages;
  until Time2 - Time1 > Hertz * Counter * 0.001;
  //until not Timer1.Enabled;

```

```

    Frqz := Count/(Counter * 0.001);          (* обчислити частоту сигналу *)
    if fMin > Frqz then fMin := Frqz;          (* знайдено меншу частоту *)
    if fMax < Frqz then fMax := Frqz;
    Average := Average + Frqz;
                                         (* для підрахунку середньої частоти сигналу *)
    Memo1.Lines.Add(Format('%2d f, f = %2.3f, [Cicle, Frqz]));
    OutDigitDisplay(StrToFloat(Format('%6.3f, [Frqz]]));
                                         (* відобразити на дисплеї значення частоти *)

    inc(Cicle);
    Application.ProcessMessages;
    until (Cicle > StrToInt(Edit2.Text)) or CheckBreak;
    Memo1.Lines.Add('.....');
    // середнє значення частоти:
    Memo1.Lines.Add(Format('Середн f = %2.3f,
[Average/StrToInt(Edit2.Text)]));
    Memo1.Lines.Add('.....');
    Memo1.Lines.Add(Format('Max f = %2.3f, [fMax]));
    Memo1.Lines.Add(Format('Min f = %2.3f, [fMin]));
    Memo1.Lines.Add(Format('Розкид = %2.3f,
[(fMax-fMin)/fMin*100] + ' %8);
    // далі може бути код для вимірювання опору, індуктивності,
    // ємності і т.д. (введення контрольованого параметру, розрахованого
    // за апроксимованою функцією);
    {
    Memo1.Lines.Add(Format('Напруга, В = %2.3f,
[0.500 * Power(2.718281, 0.1415*Frqz*0.001)]));
    }
    {
    Memo1.Lines.Add(Format('Ємність, нФ = %2.3f,
[6012/(Average/StrToInt(Edit2.Text)]));
    }
    Memo1.Lines.Add('Фініш: ' + TimeToStr(Time));
    Memo1.Lines.Add('');
    Result := StrToFloat(Format('%2.3f [Average/StrToInt(Edit2.Text)]));
    // відобразити на дисплеї середнє значення частоти:
    OutDigitDisplay(StrToFloat(Format('%6.3f,
[Average/StrToInt(Edit2.Text)]));
    Application.ProcessMessages;
    if not CheckBox2.Checked then Break;
                                         (* Якщо прапорець "У циклі?" не встановлено *)

    Inc(MainCicle);
    until False or CheckBreak;          (* кінець зовнішнього циклу *)
    DisableEnableVisual(True);          (* дозволити видні елементи *)
end;

```


Функція містить зовнішній цикл {0}, у який вкладений головний цикл {1}, у який, у свою чергу, вкладені:

- цикл, що забезпечує задану тривалість вимірювань {2};
- два цикли очікування надходження високого рівня сигналу {High};
- два цикли очікування надходження низького рівня сигналу {Low}.

Призначення циклів зрозуміле з тексту коментарів. Слід пояснити призначення конструкції `Application.ProcessMessages`, присутньої у тілі кожного циклу. Цей метод вимушує програму обробляти повідомлення, що очікують. Іншими словами, якщо не викликати цей метод з середини довгого циклу, то користувач не зможе перервати виконання підзадачі клацанням по кнопці 'Перервати', що не є припустимим.

Функція Windows API `QueryPerformanceCounter` повертає поточне значення системного лічильника високої точності. Значення частоти лічильника, що визначається іншою функцією Windows API `QueryPerformanceFrequency`, легко визначити період вхідного сигналу. Ця функція викликається у самому початку роботи програми у обробнику створення форми `TForm1.Create`. Тут вона ініціалізує глобальну змінну `Hz`, використовувану надалі для відображення частоти лічильника у вікні програми, а також для перевірки умови виходу з циклу, що забезпечує задану тривалість вимірювання.

Читання стану вхідної лінії виконує користувальницька функція:

```
function TForm1.GetModemStatus(H: THandle): DWord;  
var  
    InByte: DWord; // для перевірки статусу регістра АЦП  
begin  
    GetCommModemStatus(FHandle, InByte); // Windows API  
    Result := InByte;  
end;
```

Вона містить одну функцію Windows API `GetCommModemStatus`, яка повертає у змінній `InByte` слово стан регістра модему (модуль АЦП). Згодом результат цієї функції використовується для аналізу надходження високого і низького рівнів сигналу на обрану вхідну лінію:

repeat //зачекати високий рівень

```

Application.ProcessMessages;
until (GetModemStatus(Th) and Bit = Bit)
.....
repeat //зачекати низький рівень
Application.ProcessMessages;
until (GetModemStatus(Th) and Bit = 0)

```

Подальший опис функції FindFrqz опускаємо, з огляду на те, що її текст супроводжується вичерпними коментарями. Повернемося до реалізації функції OutDigitDisplay, "запуск" якої описана на початку глави.

4.3.3 Розробка і опис цифрової шкали

Відображення інформації на шкалі, що нагадує дисплей цифрового приладу, є тривіальним завданням. Ось код функції, що виконує це завдання:

```

procedure TForm1.OutDigitDisplay(Value: Double);
var
  S: String;
begin
  if Value > 99999.999 then begin //перевірка
    ImageList1.GetBitmap(11, Image1.Picture.Bitmap);
    ImageList1.GetBitmap(11, Image2.Picture.Bitmap);
    ImageList1.GetBitmap(11, Image3.Picture.Bitmap);
    ImageList1.GetBitmap(11, Image4.Picture.Bitmap);
    ImageList1.GetBitmap(11, Image5.Picture.Bitmap);
    ImageList1.GetBitmap(11, Image7.Picture.Bitmap);
    ImageList1.GetBitmap(11, Image8.Picture.Bitmap);
    ImageList1.GetBitmap(11, Image9.Picture.Bitmap);
    Panel1.Refresh;
    Exit;
  end;
  S := Format('%09.3f', [Value]);
  //ShowMessage('>' + S + '<' + IntToStr(Length(S)));
  if S[1] <> #32 then
    ImageList1.GetBitmap(StrToInt(S[1]), Image1.Picture.Bitmap)
  else
    ImageList1.GetBitmap(10, Image1.Picture.Bitmap);
  if S[2] <> #32 then
    ImageList1.GetBitmap(StrToInt(S[2]), Image2.Picture.Bitmap)
  else
    ImageList1.GetBitmap(10, Image2.Picture.Bitmap);
  if S[3] <> #32 then

```

```

ImageList1.GetBitmap(StrToInt(S[3]), Image3.Picture.Bitmap)
else
ImageList1.GetBitmap(10, Image3.Picture.Bitmap);
if S[4] <> #32 then
ImageList1.GetBitmap(StrToInt(S[4]), Image4.Picture.Bitmap)
else
ImageList1.GetBitmap(10, Image4.Picture.Bitmap);
if S[5] <> #32 then
ImageList1.GetBitmap(StrToInt(S[5]), Image5.Picture.Bitmap)
else
ImageList1.GetBitmap(10, Image5.Picture.Bitmap);
if S[7] <> #32 then
ImageList1.GetBitmap(StrToInt(S[7]), Image7.Picture.Bitmap)
else
ImageList1.GetBitmap(10, Image7.Picture.Bitmap);
if S[8] <> #32 then
ImageList1.GetBitmap(StrToInt(S[8]), Image8.Picture.Bitmap)
else
ImageList1.GetBitmap(10, Image8.Picture.Bitmap);
if S[9] <> #32 then
ImageList1.GetBitmap(StrToInt(S[9]), Image9.Picture.Bitmap)
else
ImageList1.GetBitmap(10, Image9.Picture.Bitmap);
Panel1.Refresh;
end;

```

Ідея полягає у наступному. У будь-якому графічному редакторі для Windows заздалегідь створюються: десять малюнків, що відображують цифри від 0 до 9; один малюнок, що містить зображення десятикової коми; один малюнок, що містить зображення символу переповнення; один малюнок із зображенням порожнього знакомісця. Всі малюнки збережені у форматі "bmp" розмірами 57х33 пікселів. Після створення і збереження малюнків у файлах, звернемося до інтерфейсного елементу ImageList1 (набір картинок), де ми клікнувши по ньому "мишею". На екрані відобразуватиметься діалогове вікно (рис.4.8).

Використовуючи кнопку Add додамо в набір заздалегідь створені картинки в такій послідовності, щоб картинки із зображеннями цифр від нуля до дев'яти йшли першими, потім йшла картинка із зображенням порожнього знакомісця і на закінченні – картинка із зображенням символу переповнення.

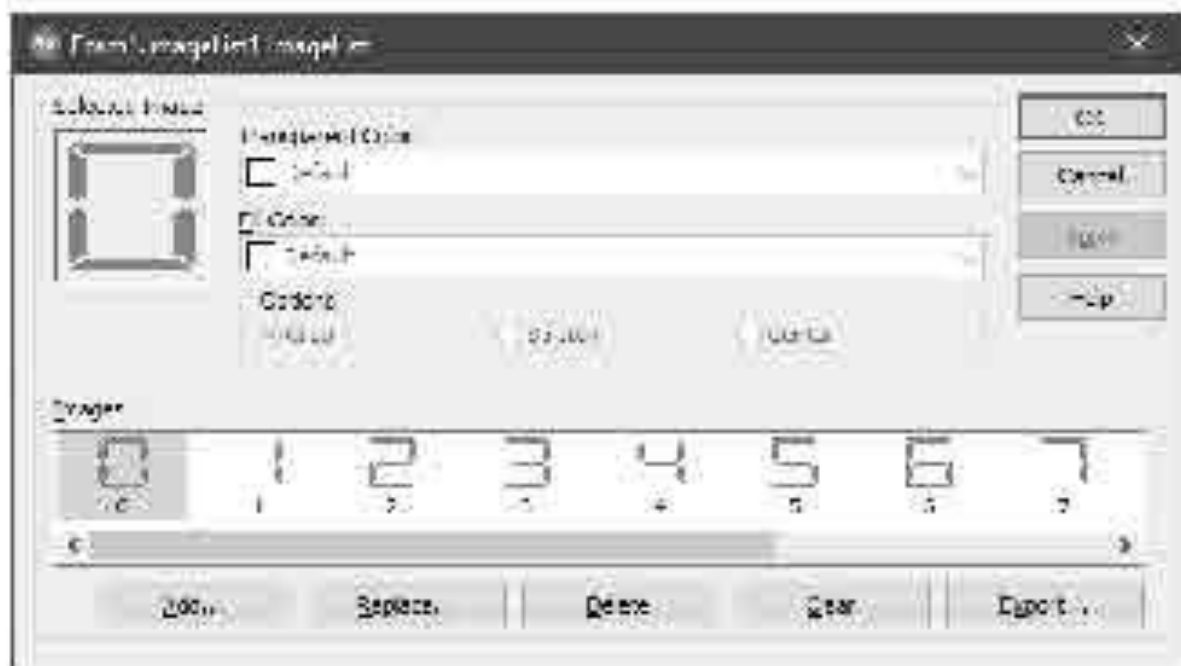


Рисунок 4.8. Зовнішній вигляд діалогового вікна елементу ImageList

У главі 4.2 показано розташування на формі у числі інших елемент Panel (панель), а також дев'ять елементів Image1...Image9 (зображення). Всі дев'ять елементів розташовані не на формі, а на панелі Panel. Це означає, що інтерфейсний елемент Panel виступає як контейнер, якому належать розташовані в ньому елементи.

Кожен з елементів Image має властивість Picture, що описує точковий малюнок або інший графічний ресурс. Залежно від параметра Value, що містить число для відображення на цифровій шкалі, функція OutDigitDisplay призначає властивостям Image1.Picture...Image9. Picture посилання на малюнок з набору ImageList1. Цю роботу виконує метод ImageList1.GetBitmap. Оператор Panel.Refresh викликає оновлення елементу Panel, що, у свою чергу, приводить до оновлення всіх пов'язаних на нього елементів Image1...Image9 і відображення на цифровій шкалі числа, що міститься у Value.

Для властивості Image6.Picture призначення не виконується, бо ця властивість містить малюнок із зображенням десятикової коми, положення якої фіксоване. Призначення значення властивості Image6.Picture виконується у інспекторі об'єктів.

4.3.4 Реалізація керування протоколом та запуск програми

Останнє, що залишається виконати, – створити обробник події по кліцянню на об'єкті `CheckBox1`. Цей обробник змушує форму розтягуватися донизу, відображуючи вміст протоколу, і повертатися до початкового розміру при знятті прапорця.

```
procedure TForm1.CheckBox1Click(Sender: TObject);  
begin  
  if CheckBox1.Checked then  
    Form1.Height:=610; //показати  
  else  
    Form1.Height:=270; //сховати  
end;
```

Зберігаємо проект, виконуємо компіляцію і запускаємо програму на виконання. Клацаємо по кнопці "Старт" і бачимо, що на цифровій шкалі нулі, програмний протокол не оновлюється. Програма виконує нескінченний цикл очікування надходження сигналу на вхідну лінію. Але оскільки до лінії нічого не підключено, цикл очікування повторюється знову і знову. Клацаємо по кнопці "Перервати" і відповімо ствердно на запит "Перервати вимірювання?". Вимірювання буде перервано, на цифровому табло і у протоколі відображатиметься випадкове значення частоти, яке не варто брати до уваги.

4.4 Опис інтерфейсу і роботи ПЗ USB-частотоміру

Зовнішній вигляд головного вікна програми показаний на рис. 4.9. Верхня частина вікна містить панель із стилізованою шкалою цифрового приладу. У правому нижньому кутку вікна розташовано три кнопки. Кнопка "Вихід" призначена для завершення роботи із програмою. Кнопка "Старт" ініціює початок вимірювань. Кнопка "Скидання" викликає негайне обнулення індикатора і очищення протоколу вимірювань.

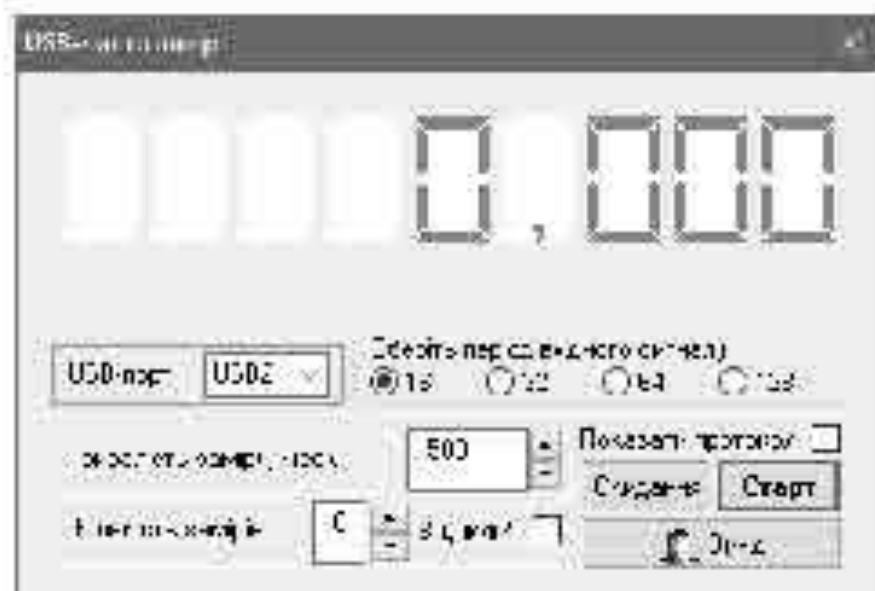


Рисунок 4.9. Зовнішній вигляд головного вікна USB-частотоміру

Протокол є програмним списком, що формується у ході вимірювань і містить початкові установки для вимірювань, їх результати, мінімальне, максимальне, середнє значення і інше. Протокол стає доступним після установки прапорця 'Показати протокол'. При цьому вікно програми розширюється донизу і відкриває текст протоколу. Цей текст доступний для копіювання через буфер обміну Windows. Зовнішній вигляд вікна програми з протоколом і реальними результатами вимірювань показаний на рис. 4.10. Решта елементів інтерфейсу призначена для аналізування ліній і встановлення параметрів вимірювань. Список вибору 'USB-порт' дозволяє вибрати USB-порт, до якого підключається розроблена схема аналогового інтерфейсу (див. Розділ 3), тобто джерело сигналу, частоту якого належить визначити. Група взаємозв'язаних кнопок під спільною назвою 'Оберіть період вхідного сигналу' надає можливість вибору одного з чотирьох доступних значень періоду вхідного сигналу послідовного порту. Вхідний сигнал підключається між обраною лінією і схемною 'землею' (SG). Елемент 'Тривалість заміру, мс' призначений для встановлення часу, протягом якого USB-частотомір 'опитує' джерело сигналу і підраховує його частоту. Чим більше час вимірювання, тим точніше отриманий результат. Елемент 'Кількість замірів' дозволяє задати де яку кількість послідовних вимірювань встановленої тривалості з отриманням

Змк.	Арх.	Наблюд.	Підпис	Дата

РП 05.08.004.00 ДП ПЗ

Арх.

66

усередненого результату, що відображується у протоколі і на цифровій шкалі. Чем більше кількість вимірювань, тим достовірніше отриманий результат.



Рисунок 4.10. Зовнішній вигляд вікна програми USB-частотоміру з протоколом і результатами вимірювань

Прапорець ‘У шкалі?’, якщо він встановлений, дозволяє перевести програму у стан безперерного вимірювання. Цей режим зручний при проведенні серії вимірювань з накопиченням результату у програмному протоколі. Слід мати на увазі, що розмір протоколу обмежений певною кількістю рядків (задається у програмі), при перевищенні якого вимірювання не буде продовжене, поки користувач не обере варіант: очистити протокол чи вийти з режиму вимірювань.

4.5 Тестування розробленого програмного забезпечення

Для перевірки роботи створеного віртуального частотомиру у режимі вимірювання частоти аналогового сигналу достатньо підключити до вхідної лінії аналогового інтерфейсу джерело вимірюваного сигналу і знову клацнути по кнопці "Старт". Переконавшись, що на цифровій шкалі і у протоколі вимірювань вірно відображується вимірювана частота, можна за допомогою зовнішнього частотомира або осцилографа. Збільшуючи тривалість вимірювання можна переконатися у підвищенні точності отримуваних результатів.

У режимі вимірювань (натиснута кнопка 'Старт') стають видимі ще два елементи інтерфейсу: панель з відображенням частоти точного таймеру і номера поточного циклу, а також кнопка 'Перервати' (рис.4.11); решта всіх елементів інтерфейсу, за винятком прапорця 'Показати протокол', стають недоступними.

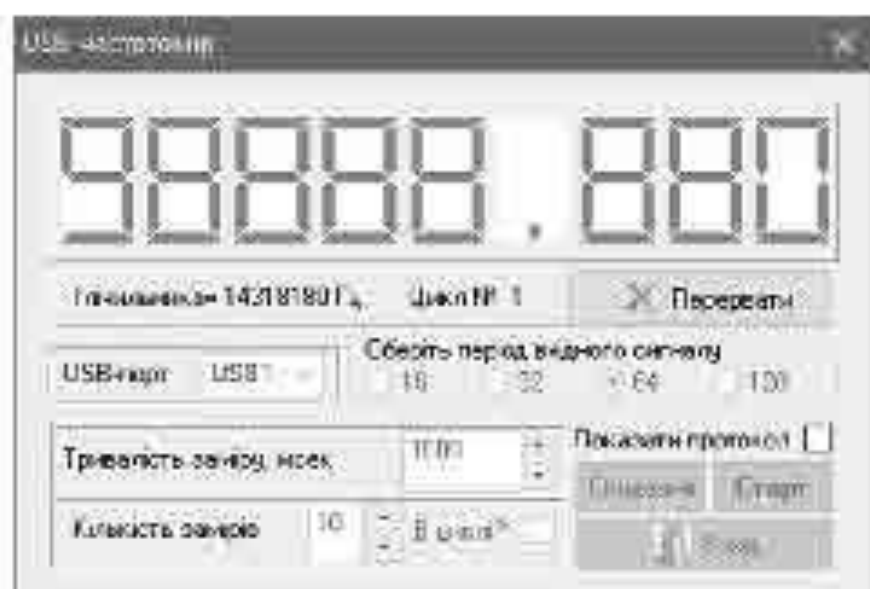


Рисунок 4.11. Режим визначення частоти аналогового сигналу

При натисненні кнопки 'Перервати' буде припинене обчислення частоти джерела та видане відповідне повідомлення у діалоговому вікні MessageDlg (рис. 4.12).



Рисунок 4.12. Переривання вимірювання частоти

Після переривання вимірювання у головне вікно додатку буде виведене значення частоти аналогового сигналу у герцах (4.13) з можливістю відображення протоколу з результатами усіх замірів (4.10).

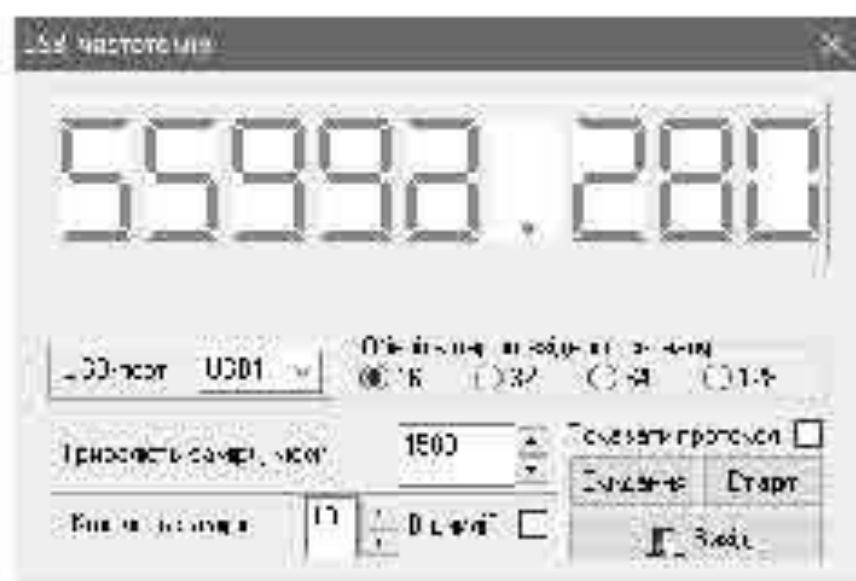


Рисунок 4.13. Відображення результату вимірювання частоти

Після натиснення кнопки 'Вихід' буде виведено попередження (рис. 4.14) про втрату результатів вимірювання з протоколу (поле Мемо), тому вміст протоколу можна скопіювати у буфер.



Рисунок 4.14. Попередження про втрату результатів вимірювань

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Резюме

В даному дипломному проекті розроблене програмне забезпечення USB-частотоміру для технічних вимірювань. Спеціалізоване програмне забезпечення дає можливість використовувати комп'ютер у якості однієї з основних ланок вимірювального комплексу, дозволяючи позбавитися від зовнішніх громіздких і складних приладів взагалі.

Ефективність кожного програмного продукту визначається його якістю та ефективністю процесу розробки. Якість ПП визначається наступними складовими: з точки зору користувача; з позиції використання ресурсів; виконання вимог до програмного забезпечення.

5.2 Визначення трудомісткості розробки програмного забезпечення

Тривалість розробки програмного продукту залежить від його обсягу, трудомісткості розробки, кваліфікації виконавців, а також планованих термінів, визначених умовами ринку. Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, у тих же умовах машинних команд програми аналога.

Таблиця 5.1 Каталог аналогів

Найменування ПП	Обсяг функцій ПП – V_a , усл. машинних команд
1. ПП автоматизації засобів по каталогу	680 – 7000
2. ПП автоматизованих розрахунків	1300 – 8600
3. ПП загальної математики і ПП імітаційного моделювання	7800 – 8800

У таблиці 5.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт. Для нашого варіанта виділено сірим кольором.

Вибраний аналог ПП, що містить V_0 в умовних машинних командах, трудомісткості визначати на основі табл.5.2

Таблиця 5.2

Обсяг ПП, тис умов машинних команд	Норма часу, людгод
1.00	229
2.00	244
3.00	262

На підставі отриманого значення, по довіднику, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера, $K_k=0,7 \div 0,8$): $T_{ар} = 229 \times 0,7 = 160,3$ (люддодин).

Трудомісткість програмного продукту визначається по кожному етапу розробки окремо на підставі трудомісткості аналога з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

$$T_{pi} = T^* p \times L_i \times K_n \quad (5.1)$$

$$T_{pi} = T^* p \times L_i \times K_n \quad (5.2)$$

$$T_{pi} = T^* p \times L_i \times K_n \times K_r \quad (5.3)$$

Для розрахунку необхідні наступні коефіцієнти:

L_i – питома вага i -го етапу розробки (див. табл. 5.2.);

K_n – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 5.3.);

K_r – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 5.4.).

Таблиця 5.2 Значення питомих коефіцієнтів трудомісткості стадії в загальній трудомісткості розробки ПП

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ (L_1)	0,15	0,12	0,12
ТП (L_2)	0,16	0,15	0,11
РП (L_3)	0,55	0,58	0,61

Для нашого варіанта виділено сірим кольором.

Таблиця 5.3. Значення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення K_n
A	Принципово нові ПО	1,25 – 1,2
B	ПО – розвиток визначеного параметричного ряду	1,0 – 0,8
B	ПО мавчий аналог	0,7

Для нашого варіанта виділено сірим кольором.

Таблиця 5.4 Значення коефіцієнта ступеня використання в розробці типів програм

Ступінь охоплення реалізованих функцій розроблювального ПО типовими програмами, %	Значення K_r
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Для нашого варіанта виділено сірим кольором.

Тепер розраховуємо трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{\text{тз}} = T_a * L_1 * K_n = 160,3 * 0,12 * 0,8 = 15,38 \text{ (люд.д/год)} \quad (5.1)$$

Трудомісткість розробки технічного проекту

$$T_{\text{тп}} = T_a * L_2 * K_n = 160,3 * 0,11 * 0,8 = 14,11 \text{ (люд.д/год)} \quad (5.2)$$

Трудомісткість розробки робочого проекту

$$T_{\text{рп}} = T_a * L_3 * K_n * K_r = 160,3 * 0,61 * 0,8 * 0,8 = 62,58 \text{ (люд.д/год)} \quad (5.3)$$

Для подальших розрахунків визначили кількість папера, витраченого на кожен етап: технічне завдання $N_{\text{тз}}=3$ (стр), розробка ТП $N_{\text{тп}}=15$ (стр), розробка робочого проекту $N_{\text{рп}}=20$ (стр), пояснювальна записка відповідно $N_{\text{пз}}=30$ (стр)

Таблиця 5.5 Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, год.ч.		
1	2	3	4
1.ТЗ	$T_{pm}=15,38$	$T_{aa}=0,7 \cdot N_m=0,7 \cdot 3=2,1$	$T_{aa}=0,15 \cdot N_m=0,15 \cdot 3=0,45$
2 Розробка ТП	$T_{pm}=14,11$	$T_{aa}=0,7 \cdot N_m=0,7 \cdot 15=10,5$	$T_{aa}=0,15 \cdot N_m=0,15 \cdot 15=2,25$
3 Розробка РП	$T_{pm}=62,58$	$T_{aa}=0,7 \cdot N_{pa}=0,7 \cdot 20=14,0$	$T_{aa}=0,15 \cdot N_{pa}=0,15 \cdot 20=3,0$
4 Розробка ПЗ	$T_{aa}=1,5, N_{aa}=1,5 \cdot 30=45$	$T_{aa}=0,7 \cdot N_m=0,7 \cdot 30=21,0$	$T_{aa}=0,15 \cdot N_m=0,15 \cdot 30=4,5$
Усього, в т.ч.:	183,87		
- на розробку	$\Sigma T_p=126,07$		
- контроль керівника		$\Sigma T_{aa}=47,6$	
- нормоконтроль			$\Sigma T_{aa}=10,2$

5.3 Розрахунок ціни програмного продукту

У цьому розділі для визначення ціни розраховуємо основну заробітну плату виконавців, матеріальні витрати, вартість машинно – годин і витрати на розробку ПО. Розрахунок основної заробітної плати виконавців приведений у таблиці 6.7. Відповідно до статті 8 «Закону про Державний бюджет України на 2021» встановлено мінімальну заробітну плату у місячному розмірі з 1 січня 2022 року - 6500 гривень; мінімальну погодинну тарифну ставку – 39.26 грн.

Таблиця 5.6 Розрахунок основної заробітної плати виконавців

Найменування робіт	Трудомісткість робіт, години	Погодинна тарифна ставка, грн.	Розрахунок, грн.
1 Розробка ПП	126,07	39,26	4949,60
2 Контроль керівника	47,6	60,10	2860,76
3 Нормоконт-роль	10,2	60,10	613,02
Усього	-	-	$\Sigma 30= 8423,38$

Зробимо розрахунок матеріальних витрат на розробку ПП. Розрахунок зведемо в таблицю 5.7

Таблиця 5.7 Розрахунок матеріальних витрат на розробку ПО

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	66	2,5	165,0
Разом	-	-	-	$B_{м1}=165,0$
Транспортно – заготівельні Витрати (10%)				$B_{тп1} = 0,1 \times B_{м1} = 0,1 \times 165 = 16,50$
Усього				$B_{м1} = B_{м1} + B_{тп1} = 181,50$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формою, приведеною в таблиці 5.8.

Таблиця 5.8 Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	181,50	$B_{м1}$ (див. табл. 5.7)
2. Основна заробітна плата	8423,38	$З_о$ (див. табл. 5.6)
3. Додаткова заробітна плата	842,34	$З_д = 0,1 \times З_о = 8423,38 \times 0,1$
4. Відрахування до єдиного фонду соціального внеску	2038,45	$Вс.с.в. = 0,22 \times (З_о + З_д) = 0,22 \times (8423,38 + 842,38)$
5. Накладні витрати	3369,35	$В_{накл.} = 0,4 \times З_о = 0,4 \times 8423,38$
6. Певна собівартість	14855,02	$C_{п1} = B_{м1} + З_о + З_д + Вс.с.в. + В_{накл.} = 181,50 + 8423,38 + 842,34 + 2038,45 + 3369,35$

Розмір прибутку, що включається в ціну, визначаємо по наступній формулі:

$$П = (C_{п1} \times P) / 100 = (14855,02 \times 10) / 100 = 1485,50 \text{ грн} \quad (5.4)$$

Де P – плановий рівень рентабельності (10-15%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$$Ц_о = C_{п1} + П = 14855,02 + 1485,50 = 16340,52 \text{ грн}; \quad (5.5)$$

Виходячи з отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме:

$$Ц_р = Ц_о + П_{ДВ} = 16340,52 + 16340,52 \times 1,2 = 19608,52 \text{ грн}; \quad (5.6)$$

6 ОХОРОНА ПРАЦІ

6.1 Вступ

Відповідно до Конституції України, громадянам забезпечується рівноправність у області праці, незалежно від національності і раси.

Умови праці впливають на здоров'я, працездатність і всебічний розвиток особи трудящого. Узагальнюючи приведені вище положення, можна зробити висновок, що чим вища культура виробництва, тим краще умови праці, а отже, забезпечується здоров'я і безпека працівників.

Робоче місце користувача послуг апаратно-програмного комплексу комп'ютерного зору складається з персонального комп'ютеру з програмним забезпеченням та може бути організовано в спеціалізованих закладах нагляду за станом дорожнього руху.

Тому для нього застосовуються зокрема вимоги до організації робочого місця користувача персонального комп'ютеру.

6.2 Коротка характеристика і основні вимоги безпеки до мікроклімату виробничих приміщень, освітлення, шуму, вібрації, ультразвуку, інфразвуку, виробничих випромінювань, небезпека ураження електричним струмом

Під час роботи на робочому місці користувача ПК виникають небезпечні та шкідливі фактори:

- підвищений рівень шуму;
- несприятливі мікрокліматичні умови;
- недостатній рівень освітленості приміщення та робочого місця;
- підвищений рівень електромагнітних випромінювань радіочастот;
- висока напруга електричної мережі;
- статична електрика та інші.

					РП 05.08.006.00 ДП ПЗ	Арх.
Змк.	Арх.	Наблюд.	Підпис	Дата		75

Робота з ПК супроводжується також підвищенням ступенем напруженості трудового процесу. При систематичному впливі виробничих факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності працівників та можуть виникнути професійні захворювання органів зору, руку, нервової системи. Таким чином, покращення умов праці на робочому місці користувача ПК є необхідною умовою запобігання негативних наслідків впливу небезпечних та шкідливих факторів.

Головним завданням будь-якої галузі промисловості є збільшення продуктивності праці. Разом з тим, людина, що працює, проводить на виробництві значну частину свого життя. Тому для її нормальної життєдіяльності в умовах виробництва треба створити санітарні умови, які б дали змогу їй спокійно працювати не перевтомлюючись та зберігати своє здоров'я. Для цього треба, щоб енергетичні витрати при праці компенсувалися відпочинком та умовами оточуючого середовища. Ці умови створюються забезпеченням для працюючого:

- 1) зручного робочого місця;
- 2) чистого повітря;
- 3) нормованої освітленості;
- 4) захисту від шуму та вібрацій;
- 5) захисту від дії шкідливих речовин та випромінювань;
- 6) робочим одягом та різними засобами індивідуального захисту;
- 7) побутовими приміщеннями та спеціальними службами, що призначені
- 8) створювати безпечні та нормальні умови праці.

6.2.1 Мікроклімат виробничих приміщень

Основні нормативні документи, де наводяться норми мікроклімату – це санітарні норми та стандарти безпеки праці. Виробничий мікроклімат характеризується температурою й вологістю повітря, швидкістю його руху, а також інтенсивністю радіації і повинен відповідати ГОСТ 12.1.005-88 і СНиП 2.04.05-86.

					РП 05.08.006.00 ДП ПЗ	Арх.
Змк.	Арх.	Наблюд.	Голов.	Діагн.		76

Оптимальні значення параметру мікроклімату повинно становити: температуру повітря від 18-22 градусів Цельсія, вологість повітря від 40%-60%, та швидкість повітря від 0,1-0,2 м/с.

Для підтримки в приміщеннях нормального, що відповідає гігієнічним вимогам складу повітря, видалення з нього шкідливих газів, пару і пилу використовують вентиляцію. В приміщеннях з ПК це змішана вентиляція – природна та механічна.

У приміщеннях, де відбувається робота програміста вимоги до параметрів мікроклімату в цілому виконанні.

6.2.2 Освітлення виробничих приміщень

Для освітлення приміщення, у якому працює користувач ПК, використовується змішане освітлення, тобто сполучення природного й штучного освітлення. Природне освітлення – здійснюється через вікна в зовнішніх стінах будинку. Штучне освітлення – використовується при недостатньому природному освітленні й здійснюється за допомогою двох систем: загального й місцевого освітлення. Для загального освітлення приміщення, використовуються газорозрядні лампи типу ЛД.

Значення освітленості на поверхні робочого столу в зоні розміщення документів має становити 300-500 лк.

6.2.3 Заходи щодо захисту від дій шуму та вібрації

У робочих приміщеннях основними джерелами акустичних шумів є шуми ПЕОМ. ЕОМ є також джерелами шумів електромагнітного походження (коливання елементів електро механічних пристроїв під впливом змінних магнітних полів). Систематичний шум може викликати стомлення слуху й ослаблення звукового сприйняття, а також значне стомлення всього організму.

Зниження рівня шуму у приміщенні можна здійснити наступними методами:

- використанням блоків живлення ПК з вентиляторами на гумових підвісках;
- облицювання стелі і стін звукопоглинаючим матеріалом;

					РП 05.08.006.00 ДП ПЗ	Арх.
Зав.	Арх.	Наблюд.	Підпис	Дата		11

- екранування робочого місця (постановкою перегородок, діафрагм);
- раціональне планування приміщення.

6.2.4 Безпека праці

Обладнання і організація робочого місця з ВДТ мають забезпечувати відповідність конструкцій всіх елементів робочого місця та їх взаємного розташування ергономічним вимогам з урахуванням характеру і особливостей трудової діяльності (ГОСТ 12.2032-78, ГОСТ 22.269-76, ГОСТ 21.889-76). Конструкція робочого місця й взаємне розташування всіх його елементів (сидіння, органи керування, засоби відображення інформації) відповідають антропометричним, фізіологічним і психологічним вимогам, а також характеру роботи. Конструкція робочих меблів повинна забезпечувати можливість індивідуального регулювання відповідно росту працівників для підтримки зручної пози. Робочий стіл повинен бути пофарбований матовою фарбою. Дисплей розташований так, що його верхній край перебуває на рівні очей на відстані близько 70 см, що укладається в у припустимі рамки від 60 до 90 см. Частота мерехтіння екрана $f_{\text{мр}} = 100 \text{ Гц}$, що відповідає умові $f_{\text{мр}} > 70 \text{ Гц}$.

Робоче місце розташоване перпендикулярно віконним прорізам, це зроблено з тією метою, щоб виключити пряму й відбиту мерехтливість екрана від вікон і приладів штучного освітлення, якими є лампи накаливання. Обладнання і організація робочого місця з ВДТ мають забезпечувати відповідність конструкцій всіх елементів робочого місця та їх взаємного розташування ергономічним вимогам з урахуванням характеру і особливостей трудової діяльності згідно правил охорони праці під час експлуатації ЕОМ.

6.2.5 Електробезпека

Приміщення, де використовуються імпульсні джерела живлення відповідно до ОНПІ24-86 і ПУЕ-87 відноситься до класу приміщень без підвищеної небезпеки поразки персоналу електричним струмом, оскільки відносна вологість повітря не перевищує 75%, температура не більш 35°C, відсутні хімічно агресивні середовища. Живлення електроприладів усередині приміщення здійснюється від двозфазної мережі з заземленою нейтраллю

					РП 05.08.006.00 ДП ПЗ	Арх.
						78
Зам.	Арх.	Наблюд.	Підпис	Дата		

напругою 220 В і частотою 50 Гц із використанням автоматів токового захисту.

У приміщенні повинна бути застосована схема заземлення.

Ураження людини електричним струмом може відбутися у випадку:

1. дотику до відкритих струмоведучих частин;
2. у результаті дотику до струмопровідних не струмоведучих елементів устаткування, що опинилися під напругою в результаті порушення ізоляції або з інших причин.

6.3 Пожежна безпека

Пожежна безпека – стан об'єкта, при якому з регламентованою ймовірністю виключається можливість виникнення та розвитку пожежі і шквду на людей її небезпечних факторів, а також забезпечується захист матеріальних цінностей.

За стан пожежної безпеки на підприємстві відповідають її керівники, начальники цехів, майстри та інші керівники. Можливими причинами виникнення пожежі в приміщенні є:

- 1) коротке замикання проводки;
- 2) користування побутовими електрорадіоприладами;
- 3) не дотримання умов протипожежної безпеки.

Для гасіння пожеж на робочому місці використовують вуглекислотні та порошкові вогнегасники.

Наявність переносних засобів пожежегасіння і вогнегасників, їхня кількість і зміст відповідає вимогам ГОСТ 12.4.009-75 і ISO 3941-77.

У приміщенні виконуються усі вимоги по пожежній безпеці відповідно до вимог НАПБ А.0.001-95 "Травила пожежної безпеки в Україні". У приміщенні також маєтс план евакуації на випадок виникнення пожежі.

					РП 05.08.006.00 ДП ПЗ	Арх.
						79
Змк.	Арх.	Наблюд.	Підпис	Дата		

ВИСНОВКИ

При діагностиці роботи або при налаштуванні будь-якого електронного пристрою важливу роль грають вимірвальні прилади. Спеціалізоване програмне забезпечення дає можливість використовувати комп'ютер у якості однієї з основних ланок вимірвального комплексу, дозволяючи позбавитися від зовнішніх громіздких і складних приладів взагалі. Для цього необхідні одна або декілька схем збору даних, які візьмуть на себе всі дії по введенню у комп'ютер аналогової і цифрової інформації, і спеціальне програмне забезпечення, яке реалізує функції віртуального вимірвального приладу. Таким чином, можна реалізувати всі функції осцилографів, частотомірів, вольтметрів і інших приладів безпосередньо на ПК. На базі однієї схеми збору даних, що містить аналого-цифровий перетворювач, можна реалізувати декілька приладів, оскільки плата АЦП є універсальним аналоговим інтерфейсом. Саме таке програмне забезпечення та схема аналогового інтерфейсу розроблені у даному дипломному проекті. Використання оптрону в розв'язку у ланцюгах зв'язку інтерфейсу USB дозволило знизити рівень цифрових перешкод і добитися максимальної безпеки при проведенні вимірювань.

Характеристики віртуальних вимірвальних приладів, зокрема розробленого у проекті USB-частотоміру для технічних вимірювань, визначаються лише технічними параметрами розробленої схеми аналогового інтерфейсу. При цьому співвідношення ціна-якість оптимальне.

Окрім персональних стаціонарних комп'ютерів для створення віртуального приладу для вимірювання частоти електричного струму можуть застосовуватися ноутбуки і переносні комп'ютери.

Поєднання у ПК можливостей швидкої цифрової обробки, отриманої при введенні через аналоговий інтерфейс інформації, з якісним одночасним відображенням результатів обробки робить віртуальні засоби на базі ПК популярними інструментами динамічних вимірювань.

					РП 05.08.000.00 ДП ПЗ	Арх.
Зм.	Арх.	Наб.ком.	Підпис	Дата		

ПЕРЕЛК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Зайцев С. Цифровые методы времячастотных измерений / Сергей Зайцев // Современная электроника. - 2009. - №6. - С.20-23.
2. Муханин Л.Г. Схемотехника измерительных устройств: Учебное пособие / Л.Г. Муханин. - СПб.: Издательство «Лань», 2009. - 288 с.
3. Гелль П. Как превратить компьютер в измерительный комплекс/ Пер. с франц. - 2 изд., испр. - М.: ДМК, 2001. - 144 с.:ил.
4. Один компьютер – вся измерительная лаборатория // Электроника: Наука, Технология, Бизнес. - 2000. - №2. - С.48-50
5. Хрусталева З.А. Электротехнические измерения. -М.: «КноРус», 2009 г.
6. Ан П. Сопряжение ПК с внешними устройствами/ Пей Ан; Пер. с англ. - 2-е изд., стер. - М.: ДМК Пресс; СПб.: Питер, 2004. - 320 с.:ил.
7. Ратхор Т.С. Цифровые измерения. Методы и схемотехника. - М.: Техносфера, 2004. - 376с.
8. Информационно-измерительная техника: Учебник для студ. ВУЗ / Г.Г. Раннев, В.А. Сурогина, В.И. Калашников и др.; под ред. Г.Г. Раннева. - М.: «Академия», 2006. - 512 с.
9. Озеров В. Delphi. Советы программистов. - СПб.: Символ - Плюс. 2002. - 912 с.:ил.
10. Практическое руководство по программированию / Пер. с англ. Б. Микс, П. Хит, Н. Рашби и др.; под ред. Б. Микса, П. Хит, Н. Рашби. - М.: Радио и связь, 2004. - 168 с.,ил.
11. Осипов Д. Л. Delphi. Программирование для Windows, OS X, iOS и Android. - СПб.: БХВ-Петербург, 2014. - 464 с.:ил.
12. Осипов Д. Л. Delphi XE2. - СПб.: БХВ-Петербург, 2012. - 912 с.:ил.
13. Кульпин Н. Б. Основы программирования в Delphi XE. — СПб.: БХВ-Петербург, 2011. - 416 с.:ил. + CD-ROM.
14. Бударин А. Концепция построения виртуальной измерительной лаборатории <http://www.asiup.ru/?p=600001>.

					РП 05.08.000.00 ДЛ ПЗ	Арх.
Зам.	Арх.	На докум.	Год	Дата		

Текст головного модулю програми
на мові Delphi для ICP RAD Studio

```

unit ComUnit;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, ComCtrls, ImageList, Buttons, Math, UFrame,
  System.ImageList;
type
  TForm1 = class(TForm)
    Memo1: TMemo;
    BitBtn1: TBitBtn;
    Button1: TButton;
    Button2: TButton;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    UpDown1: TUpDown;
    UpDown2: TUpDown;
    Bevel1: TBevel;
    Bevel2: TBevel;
    Panel1: TPanel;
    Image9: TImage;
    Image8: TImage;
    Image7: TImage;
    Image6: TImage;
    Image5: TImage;
    Image4: TImage;
    Image3: TImage;
    Image2: TImage;
    Image1: TImage;
    ImageList1: TImageList;
    ComboBox1: TComboBox;
    RadioGroup1: TRadioGroup;
    CheckBox1: TCheckBox;
    BitBtn2: TBitBtn;
    CheckBox2: TCheckBox;
    Bevel3: TBevel;
    Panel2: TPanel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
  procedure FormCreate(Sender: TObject);
  function FindFigs(Th: THandle; BitWord: TWord): TWord;
  function GetModemStatus(H: THandle): TWord;
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure DisableEnableVisual(B: Boolean);
  procedure ConfigDisplay(Value: TDouble);
  end;

```

```

procedure BiffBtnClick(Sender: TObject);
procedure CheckBoxClick(Sender: TObject);
procedure BiffBtn2Click(Sender: TObject);
procedure FormClose(Sender: TObject var Action: TCloseAction);
//procedure ButtonClick(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form1: TForm1;      //оформление об'екту TForm1
FHandle: THandle;    //дескриптор порта
WidthPort: string;  //им'я порта
LineByte: Byte;     //64х для асинхронного ввода/вывода
MemoCount: Integer; //записываемые пропускные числа переданы протоколу
CheckBox: Boolean;   //должно ли использоваться
Hertz: Integer;      //частота передачи по интерфейсу
implementation
{$R *.dfm}
(* ***** *)
(* Оформление *)
(* ***** *)
procedure TForm1.ButtonClick(Sender: TObject);
begin
Memo1.Lines.Clear; //очищаем протокол
OutDigitDisplay(0) //64х передается по интерфейсу
end;
(* ***** *)
(* Подпись параметров *)
(* ***** *)
procedure TForm1.FormClose(Sender: TObject var Action: TCloseAction);
begin
if Memo1.Lines.Count < 0 then begin //параметры не записаны протоколу
if MessageDlg('Протокол не имеет результатов работы' + 'M' +
'Писать данные в протокол необходимо вручную' + 'M' +
'Вы уверены?', mtConfirmation, [mbNo, mbYes], 0) = mrYes then begin
CheckBox := True;
Action := caFree;
and also begin
if not CheckBox.Checked then CheckBox.Checked := True;
Action := caNone;
end;
end;
end;
end;
(* ***** *)
(* Кнопка 'Справка' *)
(* ***** *)
procedure TForm1.Button2Click(Sender: TObject);
begin
(* про обработку параметров *)
if FHandle <> invalid_handle_value then CloseHandle(FHandle);
//создаем дескриптор порта
WidthPort := ComboBox1.Items.Strings[ComboBox1.ItemIndex]; //им'я порта (USB1, USB2...)
//$Baud := '1200'; //интерфейс порта
FHandle := CreateFile( //физический дескриптор порта
PChar(WidthPort) //им'я порта

```

```

global _ad, //адреси портів і шин для чипів
0, //адреса портів шинного ресурсу
Nil, //адреси безпечних портів
open_existing, //адреси портів шинного ресурсу
file_flag_overlapped, //адреси портів шинного ресурсу
0 //адреси портів шинного ресурсу
}
if FHandle = invalid_handle_value then begin //адреси портів шинного ресурсу
  MessageDlg('Порт ' + WordPort + 'адреси портів шинного ресурсу' + 'М +
  'Адреси портів шинного ресурсу' + 'М +
  'Адреси портів шинного ресурсу', mtInformation, [mbCancel], 0);
  Exit;
end;
//адреси портів шинного ресурсу
//адреси портів шинного ресурсу
//адреси портів шинного ресурсу
if RadioGroup1.ItemIndex = 0 then LineByte := 32;
if RadioGroup1.ItemIndex = 1 then LineByte := 16;
if RadioGroup1.ItemIndex = 2 then LineByte := 64;
if RadioGroup1.ItemIndex = 3 then LineByte := 128;
//адреси портів шинного ресурсу
FindFreq(FHandle, LineByte);
CloseHandle(FHandle); //адреси портів шинного ресурсу
end;
(*
  Адреси портів шинного ресурсу
*)
function TForm1.FindFreq(FHandle: THandle; Bit: Word): real;
(*
  Адреси портів шинного ресурсу
  Виводить частоту шинного ресурсу (FHandle) і
  біт шинного ресурсу (Bit). Див. табл.
  +-----+
  | I | 16 | I |
  | I | 32 | I |
  | I | 64 | I |
  | I | 128 | I |
  +-----+
  *)
var
  Counter, MainCick, Cick: Integer;
  Time1, Time2, Count: real;
  Average, Freq, fMax, fMin: Extended;
begin
  Result := -1;
  MainCick := 1;
  repeat
    if Memo1.Lines.Count > MemoCount then begin
      if MessageDlg('Протокол шинного ресурсу' + 'М +
      'Протокол шинного ресурсу' + 'М +
      'Протокол шинного ресурсу', mtInformation, [mbNo, mbYes], 0) <= mtYes then Break;
      Memo1.Lines.Clear;
    end;
    DisableEnableVisual(Fake);
    Average := 0;
    Cick := 1;
    Counter := StrToInt(Edit1.Text);
  end;

```

```

//Timer1.Interval := StrToInt(Edit1.Text);      (* транзитная задержка минимизировать *)
Label5.Caption := IntToStr(MainClock);
Panel.Refresh;
Memo1.Lines.Add('TimeN: ' + IntToStr(MainClock));
Memo1.Lines.Add('Cycle: ' + IntToStr(Time));
Memo1.Lines.Add('Транзитная задержка макс.: ' + IntToStr(Counter));
//Memo1.Lines.Add('Транзитная задержка мин.: ' + IntToStr(Timed.Interval));
Memo1.Lines.Add('Кинематическая задержка: ' + Edit1.Text);
if CheckBox1.Checked then
  Memo1.Lines.Add('*** Без учета перемещения ***')
  Memo1.Lines.Add('-----');
fMin := 1000000000000;      (* для минимизации минимального значения *)
fMax := 0;                  (* для минимизации максимального значения *)
CheckBox2.Checked := False; (* для минимизации минимизации *)
repeat                      (* до тех пор пока *)
  Count := 0;              (* количество измерений *)
  //перед тем как измерить, нужно измерить длину
  //и в то время как измерять, нужно измерить длину
  //и в то время как измерять, нужно измерить длину
  repeat                  (* до тех пор пока *)
    Application.ProcessMessages;
    until(GetModemStatus(1h) and Bit = Bit) or CheckBox2.Checked;
  repeat                  (* до тех пор пока *)
    Application.ProcessMessages;
    until(GetModemStatus(1h) and Bit = 0) or CheckBox2.Checked;
    QueryPerformanceCounter(fTime1);      (* до тех пор пока *)
    //Timed.Enabled := True;
    //и в то время как
  Repeat
    repeat                  (* до тех пор пока *)
      Application.ProcessMessages;
      until(GetModemStatus(1h) and Bit = Bit) or CheckBox2.Checked;
    repeat                  (* до тех пор пока *)
      Application.ProcessMessages;
      until(GetModemStatus(1h) and Bit = 0) or CheckBox2.Checked;
      inc(Count);          (* количество измерений *)
      QueryPerformanceCounter(fTime2);      (* до тех пор пока *)
      Application.ProcessMessages;
      until fTime2 - fTime1 > Hertz * Counter * 0.001;
    //until no + Timed.Enabled;
    Freq := Count / (Counter * 0.001);      (* до тех пор пока *)
    if fMin > Freq then fMin := Freq;      (* до тех пор пока *)
    if fMax < Freq then fMax := Freq;      (* до тех пор пока *)
    Average := Average + Freq;      (* до тех пор пока *)
    Memo1.Lines.Add(Format('%2d. fHz = %12.3f, [Clock, Freq]));
    OutDigitDisplay(StrToFloat(Format('%9.3f, [Freq]));) (* до тех пор пока *)
    inc(Clock);
    Application.ProcessMessages;
    until (Clock > StrToInt(Edit1.Text)) or CheckBox2.Checked;
  Memo1.Lines.Add('-----');
  //и в то время как
  Memo1.Lines.Add(Format('Средняя f = %12.3f, [Average/StrToInt(Edit1.Text)]));
  Memo1.Lines.Add('-----');
  Memo1.Lines.Add(Format('Макс. f = %12.3f, [fMax]'));
  Memo1.Lines.Add(Format('Мин. f = %12.3f, [fMin]'));
  Memo1.Lines.Add(Format('Погрешность = %12.3f, [(fMax-fMin)/fMin*100] + %');
  //и в то время как

```



```

//используют (используют) конформного параметру, розраховано
//запрошеном в функции
{
  Memo1.Lines.Add(Format('Нагрузка B = %12.3f,
    [0.500 * Power(2.718281, 0.1415 * Freq * 0.001)])');
}
{
  Memo1.Lines.Add(Format('Средняя мощность = %12.3f,
    [0.128(Average/StrToFloat(Edit1.Text))]);
}
Memo1.Lines.Add('Финиш: ' + TimeToStr(Time));
Memo1.Lines.Add('');
Result := StrToFloat(Format('%12.3f, [Average/StrToFloat(Edit1.Text)]));
//вспомогательная процедура для вывода значений
OutDigitDisplay(StrToFloat(Format('%9.3f, [Average/StrToFloat(Edit1.Text)])));
Application.ProcessMessages;
if not CheckBox1.Checked then Break;      (* Кнопка "Выход" не нажата *)
Inc(MainClick);
until False or CheckBox1.Checked;        (* кнопка по окончании работы *)
DisableEnableVisualForm;                (* по этому состоянию изменит *)
end;
(*****
(* Заголовок окна по этому состоянию изменит *)
*****
procedure TForm1.DisableEnableVisual(B: Boolean)
begin
  Button1.Enabled := B;
  Button2.Enabled := B;
  BitBtn1.Enabled := B;
  BitBtn2.Visible := not B;
  UpDown1.Enabled := B;
  UpDown2.Enabled := B;
  Edit1.Enabled := B;
  Edit2.Enabled := B;
  Memo1.Enabled := B;
  RadioGroup1.Enabled := B;
  ComboBox1.Enabled := B;
  CheckBox1.Enabled := B;
  Panel1.Visible := not B;
  Form1.Refresh;
end;
(*****
(* Вывод значений на экран изменит *)
(* Максимальное значение, по которому = 99999.999 *)
(* При этом переключатель на экран будет ===== *)
*****
procedure TForm1.OutDigitDisplay(Value: Double);
var
  S: String;
begin
  if Value > 99999.999 then begin //не переключать
    ImageList1.GetBitmap(11, Image1.Picture.Bitmap);
    ImageList1.GetBitmap(11, Image2.Picture.Bitmap);
    ImageList1.GetBitmap(11, Image3.Picture.Bitmap);
    ImageList1.GetBitmap(11, Image4.Picture.Bitmap);
    ImageList1.GetBitmap(11, Image5.Picture.Bitmap);
    ImageList1.GetBitmap(11, Image7.Picture.Bitmap);

```

```

ImageList1.GetBitmap(11, Image8.Picture.Bitmap);
ImageList1.GetBitmap(11, Image9.Picture.Bitmap);
Parallel Refresh;
Exit;
end;
S := Format('%03d', [Value]);
//ShowMessage(''+S+''+ IntToStr(Length(S)))
if S[1] <> #32 then
    ImageList1.GetBitmap(StrToInt(S[1]), Image1.Picture.Bitmap)
else
    ImageList1.GetBitmap(10, Image1.Picture.Bitmap);
if S[2] <> #32 then
    ImageList1.GetBitmap(StrToInt(S[2]), Image2.Picture.Bitmap)
else
    ImageList1.GetBitmap(10, Image2.Picture.Bitmap);
if S[3] <> #32 then
    ImageList1.GetBitmap(StrToInt(S[3]), Image3.Picture.Bitmap)
else
    ImageList1.GetBitmap(10, Image3.Picture.Bitmap);
if S[4] <> #32 then
    ImageList1.GetBitmap(StrToInt(S[4]), Image4.Picture.Bitmap)
else
    ImageList1.GetBitmap(10, Image4.Picture.Bitmap);
if S[5] <> #32 then
    ImageList1.GetBitmap(StrToInt(S[5]), Image5.Picture.Bitmap)
else
    ImageList1.GetBitmap(10, Image5.Picture.Bitmap);
if S[7] <> #32 then
    ImageList1.GetBitmap(StrToInt(S[7]), Image7.Picture.Bitmap)
else
    ImageList1.GetBitmap(10, Image7.Picture.Bitmap);
if S[8] <> #32 then
    ImageList1.GetBitmap(StrToInt(S[8]), Image8.Picture.Bitmap)
else
    ImageList1.GetBitmap(10, Image8.Picture.Bitmap);
if S[9] <> #32 then
    ImageList1.GetBitmap(StrToInt(S[9]), Image9.Picture.Bitmap)
else
    ImageList1.GetBitmap(10, Image9.Picture.Bitmap);
Parallel Refresh;
end;
(*****
(*      Программируем форму      *)
*****)
procedure TForm1.FormCreate(Sender: TObject);
begin
    //Оформляем заголовок программы по имени файла и значению
    //на панели инструментов по имени файла:
    if not QueryPerformanceFrequency(Header) then begin
        MessageDlg('Процессор Вашего ПК' + 'M' +
            'не поддерживает инструкции' + 'M' +
            'робота с преобразованием чисел в текст' + 'M' +
            'Далее будет закрыто.', mtError, [mbCancel], 0);
        Application.Terminate; //закрываем программу
    end;
    //заполняем форму значениями по умолчанию:

```

```

LabelOption := ' + InfoStr(Hex) + ' Int
MemoCount := 500; //максимальное количество чисел в строке протокола
//объявление процедуры для инициализации
OutDigitDisplay(0)
//процедура для проверки протокола
CheckBoxClick(Nil)
and;
{*****}
(*      Кнопка закрытия программы      *)
{*****}
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  Form1.Close;
end;
{*****}
(*      Проверка и изменение протокола      *)
{*****}
procedure TForm1.CheckBox1Click(Sender: TObject)
begin
  if CheckBox1.Checked then
    Form1.Height := 610; //проверка протокола
  else
    Form1.Height := 270; //изменение протокола
end;
{*****}
(*      Передача данных      *)
{*****}
procedure TForm1.BitBtn2Click(Sender: TObject);
begin
  if MessageDlg('Передача данных?',
    mtConfirmation, [mbNo, mbYes], 0) = mrYes then begin //передача данных
    CheckButton := True;
    DisableEnableVisual(True);
  end;
end;
{*****}
(*      Узнать параметры модема      *)
{*****}
function TForm1.GetModemState(H: THandle): DWord;
var
  InByte: DWord; //для передачи параметров модема
begin
  GetCommModemState(FHandle, InByte); //Windows API
  Result := InByte;
end;
end.

```