МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут автоматики і електротехніки
(повне найменування інституту, назва факультету)

Кафедра програмованої електроніки, електротехніки і телекомунікацій
(повна назва кафедри)

«Допущений до захисту»
Завідувач кафедри _____ В. М. Рябенький
д.т.н., професор
« 16 » червня 2025 р.

Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр«
(освітньо-кваліфікаційний рівень)

на тему: Мікропроцесорна система вимірювання температури з відображенням через Інтернет

Виконав: студент 4397ст3 групи
_____ В. І. Воскобойніков
(підпис)

Керівник роботи:
к.т.н., доцент, доц. каф. ПЕЕТ
(посада, науковий ступінь, вчене звання)
_____ Ш. М. Іхсанов
(підпис)

м. Миколаїв – 2025 року

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут автоматики та електротехніки

Кафедра програмованої електроніки, електротехніки і телекомунікацій
Спеціальність 172 «Телекомунікації та радіотехніка«
Освітня програма Телекомунікації

«ЗАТВЕРДЖУЮ»
Гарант освітньої програми
_____ О.К.Жук
(підпис)
« 14 » квітня 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття ступеня вищої освіти «бакалавр**«**
(освітньо-кваліфікаційний рівень)

Студенту Воскобойніков Владиславу Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Мікропроцесорна система вимірювання температури з відображенням через Інтернет
керівник роботи Іхсанов Шаміль Мухаметович, к.т.н., доцент, доц. каф. ПЕЕТ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ректора № УЧ-343 від «23» квітня 2025 року.

2. Термін подання роботи: червень 2025 р.

3. Вихідні дані по роботі:

4. Перелік питань, що належать до розробки (найменування розділів)
Огляд систем передачі даних по мережі, розробка апаратної частини системи моніторингу, розробка програмної частини мікропроцесорної системи моніторингу температури по мережі Internet, охорона праці

5. Перелік презентаційних матеріалів Структурна схема приладу, принципова схема, вікно перевірки зв'язку з пристроєм, вікно браузера, налаштування системи моніторингу в мережі, блок-схеми алгоритму

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Рябенький В. М.		
2	Рябенький В. М.		
3	Рябенький В. М.		
4	Тубальцев А.М.		

7. Дата видачі завдання « 24 » квітня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд систем передачі даних по мережі	квітень	Виконано
2	Розробка апаратної частини системи моніторингу	травень	Виконано
3	Розробка програмної частини мікропроцесорної системи моніторингу температури по мережі Internet	травень	Виконано
4	Охорона праці	червень	Виконано
5	Оформлення ПЗ	червень	Виконано

Студент _____ (підпис) В. І. Воскобойніков (прізвище та ініціали)

Керівник роботи _____ (підпис) Ш. М. Іхсанов (прізвище та ініціали)

Анотація

В роботі виконано аналіз технологій Інтернет і їх використання для передачі і відображення даних, які можуть створюватись і передаватись від вибраних об’єктів до необхідного комп’ютера. В конкретному випадку мається на увазі дистанційне вимірювання і передача даних про температуру, що може використовуватись для моніторингу температури небезпечних технологічних процесів, наприклад, в металургії. Пропонується вимірювання температури і підключення датчиків до Atmel-мікроконтролера по шині 1-WIRE, а через Ethernet передавати дані до приймача. Розроблені апаратні і програмні засоби передачі і відображення даних.

Ключові слова: Інтернет, 1-Wire, мікроконтролер, протоколи обміну, моніторинг інформації, Ethernet.

Abstract

The work analyzes Internet technologies and their use for data transmission and display, which can be created and transmitted from selected objects to the required computer. In a specific case, we mean remote measurement and transmission of temperature data, which can be used to monitor the temperature of hazardous technological processes, for example, in metallurgy. It is proposed to measure temperature and connect sensors to an Atmel microcontroller via the 1-WIRE bus, and transmit data to the receiver via Ethernet. Hardware and software tools for data transmission and display have been developed.

Keywords: Internet, 1-Wire, microcontroller, exchange protocols, information monitoring, Ethernet.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

HMI – Human-Machine Interface

OPC – OLE for Process Control

OLE – Object Linking and Embedding

ODBC – Open Database Connectivity

DDE – Dynamic Data Exchange

DLL – Dinamic Link Library

VBA – Visual Basic Application

VBS – Visual Basic Scripting

RFX – Record Field Exchange

ОС – операційна система

БД – база даних

АСК – автоматизована система керування

СКБД – система керування базами даних

ВСТУП

У наш час Інтернет став доступним не лише через комп'ютерні мережі, але й через супутники зв'язку, радіосигнали, кабельне телебачення, телефонні лінії, мережі стільникового зв'язку, спеціальні оптико-волоконні лінії і електропроводи. Всесвітня мережа стала невід'ємною часткою життя у розвинутих країнах, та країнах, котрі розвиваються.

Фактично Інтернет став 4-ю індустріальною революцією. Нині словосполучення «Індустрія 4.0» асоціюється з усім передовим, його часто вживають щоб показати все новітнє і про своє відношення до нього. Наприклад, замінюють словосполучення «новітні технології» на «технології Індустрії 4.0»

Але, не всі знають, що 4-та промислова революція, яка була об’явлена до того вона відбудеться. Можна казати про епоху 4-ї промислової, але про факт її настання очевидно що ще рано. По суті, саме словосполучення «Industrie 4.0» – є назвою Німецької програми розвитку цифрового виробництва. У інших передових країнах подібні програми розвитку називаються інакше. Наприклад в США це зветься промисловий Інтернет Речей (Industrial Internet of Things). В Україні національної програми наразі немає. Натомість є національний рух «Індустрія 4.0 в Україні» (https://industry4-0-ukraine.com.ua), який об’єднує велику кількість національних компаній для інтенсифікації розвитку та впровадження високих технологій на виробництві.

					172.4397ст3.КР.ПЗ.Р1			
Змн.	*Арк.*	*№ документу*	*Підпис*	*Дата*				
Розробник		Воскобойніков В. І.			ОГЛЯД СИСТЕМ ПЕРЕДАЧІ ДАНИХ ПО МЕРЕЖІ	*Літ.*	*Арк.*	*Аркушів*
Консультант		Рябенький В. М.					8	
Н. контр.		Добрінова Л. П.				НУК імені адмірала Макарова		
Керівник		Іхсанов Ш. М.						
Зав. каф.		Рябенький В. М.						

РОЗДІЛ 1. ОГЛЯД СИСТЕМ ПЕРЕДАЧІ ДАНИХ ПО МЕРЕЖІ

1.1. Історична довідка

У 1962 році Джозеф Ліклайдер (1915-1990), керівник Агентства передових оборонних дослідницьких проектів США (англ. Defense Advanced Research Projects Agency) висловив ідею Всесвітньої комп'ютерної мережі. У 1969 році Міністерство оборони США започаткувало розробку проекту, котрий мав на меті створення надійної системи передачі інформації на випадок війни. Агентство (англ. DARPA) запропонувало розробити для цього комп'ютерну мережу. Розробка була доручена Каліфорнійському університету Лос-Анджелеса, Стенфордському дослідному центрові, Університету штату Юта та Університету штату Каліфорнія в Санта-Барбарі. Ця мережа була названа ARPANET (англ. Advanced Research Projects Agency Network – Мережа Агентства передових досліджень). В рамках проекту мережа об'єднала названі заклади. Всі роботи фінансувались за рахунок Міністерства оборони. ARPANET почала активно рости й розвиватись; її дедалі ширше почали використовувати вчені із різних галузей науки.

Перший сервер ARPANET було встановлено 1 вересня 1969 року у Каліфорнійському університеті в Лос-Анжелесі. Комп'ютер «Honeywell 516» мав 12 кілобайт оперативної пам'яті.

До 1971 року була розроблена перша програма для відправки електронної пошти мережею, котра відразу стала дуже популярною.

У 1973 році до мережі через трансатлантичний кабель були підключені перші іноземні організації з Великобританії та Норвегії – мережа стала міжнародною.

У 1970-х роках мережа загалом використовувалась для пересилки електронної пошти, тоді ж появились перші списки поштових розсилок, групи новин та дошки оголошень. Але в ті часи мережа ще не могла легко взаємодіяти з іншими мережами, котрі були побудовані на інших технічних станда-

ртах. До кінця 1970-х років почали активно розвиватись протоколи передачі даних, що були стандартизовані у 1982–1983 роках.

1 січня 1983 року мережа ARPANET перейшла з протоколу NCP на протокол TCP/IP, який досі успішно використовується для об'єднання мереж. Саме у 1983 році за мережею ARPANET закріпився термін «Інтернет».

У 1984 році була розроблена система доменних назв (англ. Domain Name System, DNS). Тоді ж у мережі ARPANET з'явився серйозний суперник – Національний науковий фонд США (NSF) заснував міжуніверситетську мережу NSFNet (англ. National Science Foundation Network), котра була сформована з дрібніших мереж, включаючи відомі на той час Usenet та Bitnet і мала значно більшу пропускну здатність, аніж ARPANET. До цієї мережі за рік під'єдналось близько 10 тисяч комп'ютерів; звання «Інтернет» почало плавно переходити до NSFNet.

У 1988 році було винайдено протокол Internet Relay Chat (IRC), завдяки якому в Інтернеті стало можливим спілкування в реальному часі (чат).

У 1989 році в Європі, в стінах Європейського центру ядерних досліджень (франц. Conseil Européen pour la Recherche Nucléaire, CERN) народилась концепція тенет. Її запропонував знаменитий британський вчений Тім Бернерс-Лі, він же протягом двох років розробляв протокол HTTP, мову гіпертекстової розмітки HTML та ідентифікатори URI.

У 1990 році мережа ARPANET припинила своє існування, програвши конкуренцію NSFNet. Тоді ж було зафіксовано перше підключення до Інтернету телефонною лінією (так зване «дозво́нювання» англ. Dial-up access).

У 1991 році тенета стали доступні в Інтернеті, а в 1993 році з'явився знаменитий Веб-браузер (англ. web-browser) NCSA Mosaic. Всесвітня павутина ставала дедалі популярнішою.

У 1995 році NSFNet повернулась до ролі дослідницької мережі; маршрутизацією всього трафіку Інтернету тепер займались мережеві провайдери

(постачальники послуг), а не суперкомп'ютери Національного наукового фонду.

Протягом 1990-х років Інтернет об'єднав у собі більшість існуючих на той час мереж (хоча деякі, як, наприклад, Фідонет, залишились відособленими). Завдяки відсутності єдиного керуючого центру, а також завдяки відкритості технічних стандартів Інтернету, що автоматично робило мережі незалежними від бізнесу чи уряду, об'єднання виглядало неймовірно привабливим. До 1997 року в Інтернеті нараховувалось близько 10 мільйонів комп'ютерів і було зареєстровано більше мільйона доменних назв. Інтернет став дуже популярним засобом обміну інформацією.

У наш час Інтернет став доступним не лише через комп'ютерні мережі, але й через супутники зв'язку, радіосигнали, кабельне телебачення, телефонні лінії, мережі стільникового зв'язку, спеціальні оптико-волоконні лінії і електропроводи. Всесвітня мережа стала невід'ємною часткою життя у розвинутих країнах, та країнах, котрі розвиваються.

1.2. Фізична і логічна топології

Принцип роботи Ethernet заснований на принципах роботи комп'ютерних мереж і на розбитті задачі мережевого зв'язку на рівні.

В даний час термін Ethernet використовується для опису всіх локальних мереж, що використовують метод колективного доступу до середовища передачі даних з упізнанням несучої і виявленням невідповідностей. Щоб зрозуміти цей метод, розглянемо поняття фізичної і логічної топології мережі.

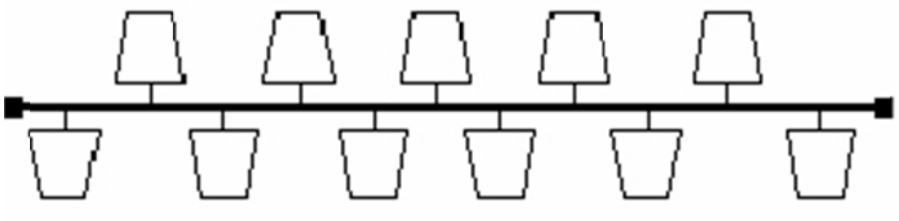

Рисунок 1.1 – Шинна топологія

Фізична топологія мережі – це реальне з'єднання її вузлів і ліній зв'язку.

Шинна топологія (топологія шини) – найпростіша і найбільш часто використовувана рис. 3.1. При цьому кожен вузол підключається до відрізка кабелю, який з'єднує всі станції послідовно. Кінці кабелю не з'єднуються один з одним, а закінчуються одним зі стандартизованих способів.

Недолік такої мережі полягає в тому, що обрив кабелю в якомусь місці виводить з ладу всю мережу. Хоча шину дуже легко прокласти, необхідно ретельно спланувати розташування вузлів, оскільки після прокладки шини не буде можливості переміщати робочі станції

Сучасні локальні мережі Ethernet і Fast Ethernet будуються на основі витої пари і концентраторів (комутаторів) за фізичною топологією «зірка» і «дерево».

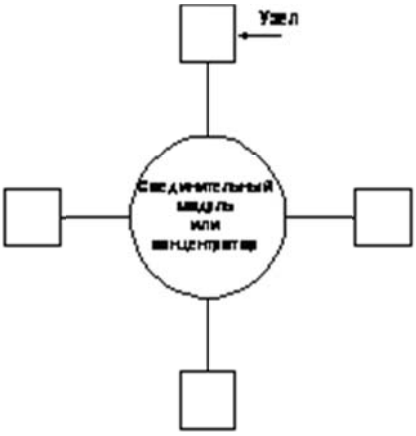

Рисунок 1.2 – Топологія зірка

В топології зірки (star topology) кожен вузол з'єднаний з центром з'єднувальним модулем (hub) або концентратором (concentrator); він діє як центральний вузол зв'язку всієї мережі. Деякі моделі з’єднувальних модулів і концентраторів мають вбудовані діагностичні засоби, керовані відповідним програмним забезпеченням.

Перевага такої топології полягає в тому, що при обриві зв'язку між одним з вузлів і центром решта продовжує працювати. Тільки відмова з'єднувального модуля або концентратора впливає більш ніж на одну станцію.

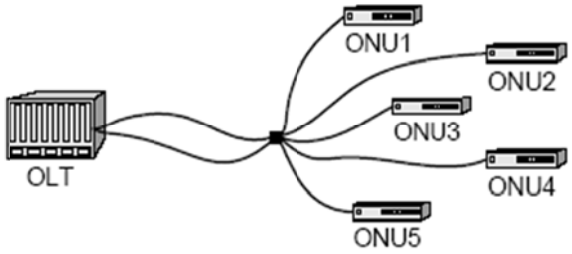

Рисунок 1.3 – Топологія дерево і дерево з надмірністю

Залишилася також зворотна сумісність з мережами Ethernet на коаксіалі, такі змішані мережі будуються за комбінованою або гібридною топологією.

Гібридна топологія – це комбінація кількох різних топологій. Гібридна топологія популярна в глобальних мережах та мережах підприємств, в яких є основне – «станове» – кільце, до якого підключаються інші мережі.

Логічна топологія – це схема з'єднання, пов'язана з методом доступу до передавального середовища. У мережі Ethernet всі комп'ютери локальної мережі мають можливість одночасного доступу до передавальноого середовища, тому логічна топологія є «шиною». Незважаючи на зміну фізичної топології у Fast Ethernet, при цьому не змінився метод доступу до середовища тому логічна топологія також не змінилася.

Ми вже знаємо, що в Ethernet всі комп'ютери мережі мають можливість одночасно отримувати дані, які будь-який з комп'ютерів почав передавати на загальну шину. Кабель, до якого підключені всі комп'ютери, працює в режимі колективного доступу. У конкретний момент часу передавати дані на загальну шину може тільки один комп'ютер в мережі. При цьому всі комп'ютери мережі мають рівні права доступу до середовища. Щоб впорядкувати доступ комп'ютерів до загальної шини, використовується метод колективного доступу з упізнаванням несучої і виявленням колізій (CSMA / CD).

1.3. Протоколи

Важливим елементом організації зв'язку між комп'ютерами, що дозволяє їм «спілкуватися» один з одним, є протоколи. Протоколи можна порівняти з правилами етикету для комп'ютерів.

Завдяки протоколам мережеві пристрої мають можливість взаємодіяти. Протоколи надають загальновизнані мову та правила такої взаємодії.

Наприклад, IP (Internet Protocol), який є протоколом міжмережевого обміну в стеку протоколів TCP/IP, забезпечує проходження блоків інформації, які називаються дейтаграммами, за різними маршрутами в мережі Інтернет. У кінцевому пункті маршруту ці дейтаграми збираються в одне повідомлення.

Інші протоколи, зокрема Ethernet, як уже було сказано вище, визначають взаємодію персональних комп'ютерів, що знаходяться в одній будівлі або офісі.

В Інтернеті також використовується протокол передачі гіпертексту (HyperText Transfer Protocol, HTTP), за допомогою якого користувачі отримують доступ до веб-сторінок і документів.

Комп'ютери Macintosh фірми Apple, можуть спілкуватися один з одним завдяки протоколу AppleTalk.

Протоколи забезпечують рішення, наприклад, таких питань:

- Хто першим починає передачу даних?

- Як визначається черговість обміну інформацією в мережі з безліччю пристроїв?

- Яка схема адресації таких мережевих пристроїв, як комп'ютери?

- Як визначити, чи була допущена помилка при передачі?

- Яким способом виправляти помилки?

- Якщо ніхто з учасників обміну не посилає інформацію в мережу, – як довго вони повинні знаходитися в режимі очікування, перш ніж з'єднання буде розірвано?

- Чи слід у разі виявлення помилки ініціювати повторну передачу всього повідомлення або ж тільки того сегмента, в якому помилка виявлена?

- Якими блоками повинні передаватися дані: по одному біту або групами бітів? Скільки бітів повинно бути в блоці? Чи слід упаковувати дані в так звані пакети?

Протоколи можуть відрізнятися по швидкості та ефективності виконання своїх функцій. Наведемо два приклади.

- SLIP (serial line internet protocol – протокол послідовного міжмережевого зв'язку) – дозволяє комп'ютерам використовувати протокол IP для отримання віддаленого доступу в Інтернет.

- PPP (point-to-point protocol – протокол двоточкового зв'язку, або зв'язку точка-точка). В даний час майже повністю витіснив SLIP. Може працювати не тільки з TCP/IP, але і з іншими протоколами передачі даних. Має поліпшені в порівнянні з SLIP засоби захисту. Як і SLIP, дозволяє з'єднувати віддалені локальні мережі.

Також вони виконують функції, такі як виявлення та виправлення помилок чи пересилання файлів, однаково у всіх комп'ютерах, щоб ті могли взаємодіяти. Один комп'ютер може передавати дані в інший, використовуючи, наприклад, протокол IPX-протокол мережевого рівня операційної системи (ОС) NetWare, розроблений фірмою Novell для передачі даних локальними мережами ((local area networks , LAN).

Комп'ютери та інші пристрої від різних виробників повинні вміти обмінюватися інформацією через мережу. Задачу організації їх взаємодії виконують різні мережеві моделі та набори (стеки) протоколів. Мережева модель з'єднує комп'ютери та периферійні пристрої в узгоджену структуру, в якій кожній мережевій функції відповідає певний рівень. Рівні всередині моделі реалізуються протоколами, що визначають метод виконання цих функцій, наприклад маршрутизації, контролю помилок, адресації і т.д. Мережева мо-

дель або набори протоколів грають роль своєрідного захисного екрану, під яким різні компоненти мережі «спілкуються» один з одним.

Передача голосу по IP була реалізована різними способами з використанням як власних і відкритих протоколів і стандартів. Приклади технологій, що використовуються для реалізації передачі голосу по IP включають:

- H.323;
- IP Multimedia Subsystem (IMS);
- Media Gateway Control Protocol (MGCP);
- Session Initiation Protocol (SIP);
- У режимі реального часу Transport Protocol (RTP);
- Сесія Опис протоколу (СДП);
- Inter-Asterisk Біржа (IAX).

Протокол H.323 був одним з перших протоколів VoIP, який знайшов широке впровадження для дальніх передач, а також задовольняв послугам місцевої мережі. А Session Initiation Protocol (SIP) получив широке розповсюдження на ринку VoIP проникнення.

Помітним кроком вперед стала реалізація протоколу Skype, який частково заснований на принципах рівноправних вузлів ЛВС (P2P) мереж.

1.4. «Реальний» час

Одна з основних проблем, в галузі управління часу, полягає в забезпеченні адекватних механізмів для вимірювання моментів, в які повинен відбутися обмін сигналами, і тривалості інтервалів часу між подіями.

Справитися з цією проблемою можна забезпечуючи процеси посиланням на час зазначеної точності. Посилання може бути побудовано, синхронізацією точності місцевого часу в реальному масштабі часу, включеному в кожному процесорі системи, щоб отримати глобальний час в межах системи.

Алгоритм дій повинен відповідати наступним чотирьом вимогам:

1) алгоритм синхронізації годинників повинен бути здатний до обмеження максимальної відмінності значень часу між спостереженням того ж самого випадку від будь-яких двох різних вузлів системи (виміряних згідно значенням місцевого часу кожного з цих двох вузлів);

2) поняття глобального часу, побудованого алгоритму синхронізації повинно бути достатньо точним щоб дозволити тому вимірювати маленькі інтервали;

3) алгоритм синхронізації годинників повинен бути здатний обробляти можливі помилки місцевих RT годинників, або втрату повідомлення про синхронізацію часу;

4) робота системи не має сповільнюватися виконанням алгоритму синхронізації годин.

Для виконання цих вимог є два підходи.

Централізований підхід може бути здійснений за допомогою центральної одиниці синхронізації, наприклад «сервера часу» вузла, відповідального за розподіл повідомлень про синхронізацію часу іншим вузлам в системі, такий підхід дуже вразливий до відмов одиниці синхронізації безпосередньо.

Децентралізований підхід, внаслідок надмірності в розподіленій інфраструктурі, яка може використовуватися для виконання, цей підхід може пропонувати кращі гарантії щодо захисту від помилок. Як уже згадано, алгоритми синхронізації годин в розподілених RT системах можуть бути здійснені обмінами повідомлення. Однак такі алгоритми можуть зачіпати продуктивність системи, таким чином порушуючи вимогу 4 вище.

Для цієї проблеми, практичне та ефективне рішення було запропоновано і розвинене в межах контексту проекту MARCA.

Щоб забезпечити передбачуваність і надійність, приведення до відповідності часу, перш за все необхідна підтримка зв'язку, який вони використовують, і забезпечення їх детермінованою поведінкою інфраструктури зв'язку.

Ця поведінка може бути досягнута Архітектурою протоколу зв'язку, характеризувався такими детермінованими властивостями як:

- обмежена затримка доступу каналу – інтервал часу між моментом, в якому завдання робить запит про посилку повідомлення, і моментом, в якому в локальному вузлі, де ця завдання виконується, фактично передається, що повідомлення на каналі зв'язку;

- і обмежена затримка повідомлення – інтервал часу між моментом, в якому завдання запитує передачу повідомлення, і моментом, в якому то повідомлення успішно доставлено.

Всі ці дії обумовлені протоколом.

Різноманітність систем, показують нам широкий спектр технологічних рішень, які можуть бути обрані при проектуванні систем реального часу.

Наприклад, SPRING, призначена для передбачуваних інфраструктур реального часу, ігнорує проблеми виправлення помилок, але її автори розробили гнучку архітектуру, яка може бути пристосована до завдань з різними вимогами (критичним, істотним і несуттєвим завданням).

У розробці HARTS велику увагу приділено низькорівневій архітектурі і введена мережу з особливою топологією.

MARS (передбачувана, синхронізована архітектура, яка використовує переваги синхронної системи, щоб доставити користувачеві інфраструктуру для роботи додатків жорсткого реального часу.

Нарешті, в системах MARUTI і CHAOS досліджено застосування об'єктно-орієнтованого підходу до розробки високонадійних операційних систем реального часу. MARUTI надає можливість інтегрованого виконання як звичайних завдань, так і завдань реального часу. CHAOS підтримує можливість виправлення помилок та надає можливість самим додаткам виправити зроблені ними помилки.

«Залежність від часу» є неодмінний атрибут системи реального часу. Але виконання цієї вимоги ще недостатньо для ефективної роботи системи, тому що в першу чергу система реального часу повинна бути «передбачуваною».

Я знаю дві архітектури, що забезпечують передбачуваність системи:

- це або всі дії в системі синхронізовані за часом,

- або коли кожній дії передує певна подія.

Мета цих двох типів архітектур (задовольнити вимогу передбачуваності, застосовуючи статичну або динамічну стратегії відповідно, щоб підтримати вимоги до ресурсів і тимчасові вимоги, що пред'являються до завдань.

Наступний важливий момент – це синхронізація годинників в застосуванні до керуючих систем реального часу. Обмін повідомленнями синхронізації часу дозволяє досягти ефективності алгоритмів синхронізації, які можуть бути використані в цих системах.

Не менш важливий факт взаємодії між процесами в системах реального часу.

Важливими характеристиками комунікаційного механізму також є:

- відсоток втрати в звуковому повідомленні,

- швидкість передачі його,

- ефективність використання каналу передачі даних і

- масштабованість механізму.

До недавнього часу розповсюдження звуку передбачало наявність великої кількості джерел симетричного звукового сигналу, підключених до центральної системи обробки та комутації сигналу. Кількість кабельних з'єднань при цьому могла бути просто величезна. Для зменшення наведених перешкод і шуму необхідно було прокладати звукові кабелі осторонь від кабелів, що несуть інші сигнали.

Кабель, який використовується для передачі звукового сигналу, звичайно являє собою дорогий варіант витої пари, захищеної екраном (shielded twisted pair – STP). З появою стандарту передачі цифрового звуку AES3 з'явилася можливість передавати звуковий сигнал в цифровій формі. Цей варіант має ряд переваг в порівнянні з аналоговим.

По-перше, лінії передачі аналогового сигналу не захищені від перешкод. Ці перешкоди, незалежно від їх рівня, складаються з корисним сигналом

і мають властивість наростати по всій довжині кабелю. Якщо ж передача сигналу здійснюється в цифровій формі, то вплив перешкод проявляється тільки в тому випадку, коли вони перевищують певний поріг. Більш того, цифровий звуковий сигнал можна передавати по волоконно-оптичної лінії, де забезпечується повна електрична ізоляція і захист від перешкод. Захист від перешкод при цифровій передачі звуку є безперечною перевагою, що дозволяє знизити вартість системи при збереженні і навіть підвищенні якості звуку. У більшості випадків передачу звукового сигналу в цифровій формі можна здійснити за допомогою недорогої неекранованої крученої пари (unshielded twisted pair – UTP).

По-друге, цифрові дані можуть бути мультиплексовані набагато ефективніше (збільшення економічного ефекту), ніж дані в аналоговій формі. Можна використовувати для мультиплексування кілька каналів цифрового звуку, передаються по одному кабелю. У цьому випадку економія на кабелях, обслуговуванні, технічній підтримці і т. д. може бути воістину величезною.

По-третє, системи цифрової передачі даних працюють в режимі замкнутої петлі. У цифровій системі, завдяки застосуванню алгоритмів виявлення і корекції помилок, можна гарантувати або верифікувати цілісність прийнятих даних перед їх використанням. Аналогові замкнуті системи передачі мають обмежені функціональність і сферу застосування. Якщо в аналоговій системі виникають перешкоди або порушується з'єднання, то виникають в результаті цих несправностей шум або тиша все одно сприймаються обладнанням приймаючої сторони як нормальний сигнал.

1.5. «Пересилка пакетів і комутація всередині маршрутизатора

IP протокол (англ. IP – Internet protocol) – найбільш широко розповсюджена реалізація ієрархічної схеми мережної адресації. Використовуваний в мережі Інтернет, протокол відповідає за адресацію пакетів, але не відповідає за встановлення з'єднань, не є надійним і дозволяє реалізувати тільки негарантовану доставку даних. Термін «протокол без встановлення з'єднань» (англ.

connectionless) означає, що протокол для взаємодії не потребує виділеного каналу, як це відбувається під час телефонної розмови і не існує процедури виклику перед початком передачі даних між мережними вузлами. Протокол IP вибирає найбільш ефективний шлях з числа доступних на основі рішень прийнятих протоколом маршрутизації. Відсутність надійності і негарантована доставка не означає, що система працює погано або ненадійно, а вказує лиш на те, що протокол IP не докладає ніяких зусиль, щоб перевірити чи був пакет доставлений за призначенням. Ці функції делеговані протоколам транспортного та вищих рівнів. Транспортний рівень також відповідає за збірку пакетів у повідомлення в потрібній послідовності.

Інформація, проходячи вниз по рівням моделі OSI, на кожному рівні певним чином обробляється протоколами цього рівня. На малюнку виможете бачити, як на мережному рівні дані інкапсулюються всередину пакетів, іноді названих дейтаграммами (датаграмами). Протокол IP розпізнає формат заголовку пакету (адресну частину та іншу службову інформацію включно), але ніяким чином не аналізує і не піклується про фактичні дані. Він приймає і передає будь-які дані, передані протоколами верхніх рівнів.

Заголовок і хвіст фрейму відкидаються і замінюються новими щоразу при проходженні пакетом маршрутизатора. Причина в тому, що блоки інформації другого рівня (фрейми) використовуються для локальної доставки пакетів, у той час як блоки третього рівня (пакети) призначені для скрізної передачі даних згідно з схемою маршрутизації.

Ethernet-фрейми другого рівня призначені для роботи всередині широкомовних доменів з призначеними кожному мережному пристрою MAC-адресами. Фрейми другого рівня інших типів, як наприклад послідовні двохточкові з'єднання або Frame relay розподілених мереж (мереж WAN), використовують свою власну схему адресації другого рівня. Принциповим є те, що незалежно від використовуваної схеми адресації другого рівня, всі вони розроблені для використання всередині одного широкомовного домену другого

рівня. При проходженні крізь пристрій третього рівня інформація другого рівня змінюється.

Із фрейму, що приходить на інтерфейс роутера, витягається MAC-адреса і перевіряється, адресований цей пакет безпосередньо якомусь вузлу чи інтерфейсу, чи він є широкомовним (ця процедура виконується всіма пристроями всередині домену колізій). В будь-якому з цих випадків пакет буде оброблено, в іншому – відкинуто, оскільки він адресований іншому вузлу в домені колізій. Таким чином, домен колізій – розподілене середовище передачі даних, в якому пристрої працюють в режимі конкуренції. На основі значення, що зберігається в полі контрольної суми, за допомогою циклічного збиткового коду (Cyclical Redundancy Check – CRC), що був вилучений з хвоста отриманого фрейму, перевіряється чи були дані пошкоджені. Якщо перевірка дає позитивний результат – фрейм відкидається. У випадку негативного результату, заголовок і хвіст фрейму відкидаються і пакет передається на третій рівень. Далі виконується перевірка, чи було пакет адресовано маршрутизатору, чи потрібна подальша маршрутизація на шляху до місця призначення. Пакети, адресовані роутеру в якості IP-адреси отримувача, мають адресу одного з його інтерфейсів. У таких пакетів видаляється заголовок і вони передаються на четвертий, транспортний рівень. Якщо пакет потребує маршрутизації, IP-адреса пункту призначення пакету порівнюється з записами в таблиці маршрутизації. Якщо знайдено точну відповідність або існує стандартний маршрут, – пакет відправляється на інтерфейс, що вказано в таблиці маршрутизації. Коли пакет комутується на вихідний інтерфейс, нове значення CRC додається у хвіст фрейму і, в залежності від типу інтерфейсу (Ethernet, Frame relay або послідовний), пакету додається відповідний заголовок. Після чого фрейм пересилається в інший широкомовний домен на шляху до кінцевого пункту призначення.

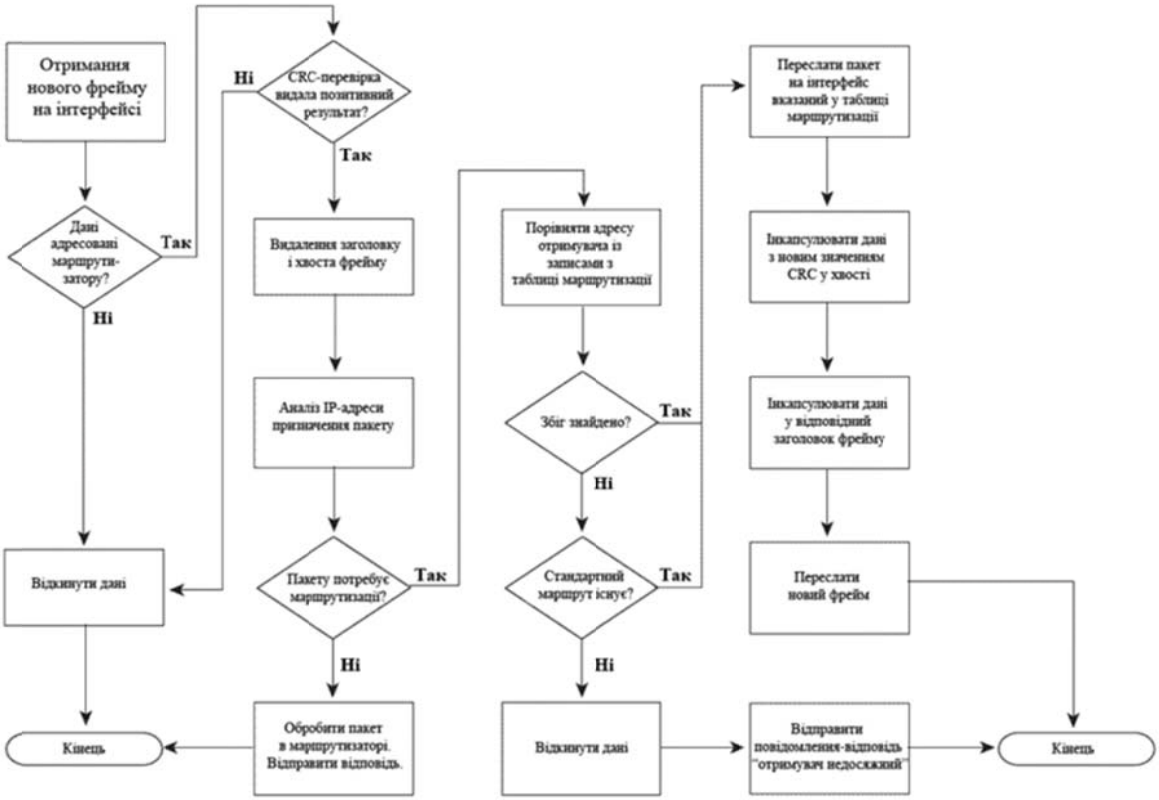

Рисунок 1.4 – Процеси, виконувані пристроями третього рівня

					172.4397ст3.КР.ПЗ.Р2	
Змн.	*Арк.*	*№ документу*	*Підпис*	*Дата*		
Розробник		Воскобойніков В. І.			РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ СИСТЕМИ МОНІТОРИНГУ	*Літ.* / *Арк.* 24 / *Аркушів*
Консультант		Рябенький В. М.				
Н. контр.		Добрінова Л. П.				НУК імені адмірала Макарова
Керівник		Іхсанов Ш. М.				
Зав. каф.		Рябенький В. М.				

РОЗДІЛ 2. РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ СИСТЕМИ МОНІТОРИНГУ

В останні роки для побудови систем збору та обробки інформації використовуються мікроконтролери, оскільки вони мають досить розвинену периферію та дозволяють використовувати прості схемотехнічні рішення. На основі проведеного аналізу були визначені вимоги до апаратної частини, що дозволить виконувати дистанційний контроль температури. В розділі виконується розробка структурної та принципової схеми системи дистанційного контролю температури, виконується розробка друкованої плати.

2.1. Структурна схема

Основним елементом системи моніторингу температури є пристрій збору та обробки інформації. Даний пристрій розташовується безпосередньо на об’єкті, температуру якого потрібно контролювати. Система моніторингу температури повинна мати блоки вводу/виводу, блоки індикації, обміну з зовнішніми пристроями або терміналами.

Для отримання даних по температурі використовується цифровий датчик DS18B20. Для відображення передачі даних і наявності збоїв системи застосовується світлодіодна індикація. Представлення системи моніторингу у вигляді структурної схеми дозволяє наочно розділити пристрій на ряд виконавчих елементів. Система моніторингу здійснює контроль температури, має модуль передачі даних, блоки індикації та вводу, обчислювальний пристрій і порт налагодження низькорівневого програмного забезпечення.

Пристрій має такі характеристики:

- підключення до роутеру за допомогою витої пари;
- мережа 1-wire, до якої можна підключити до 20 датчиків DS18B20;
- управління і налаштування пристрою за допомогою браузера;
- можливість встановлювати ім'я для каналів і датчиків, зберігається в незалежній пам'яті.

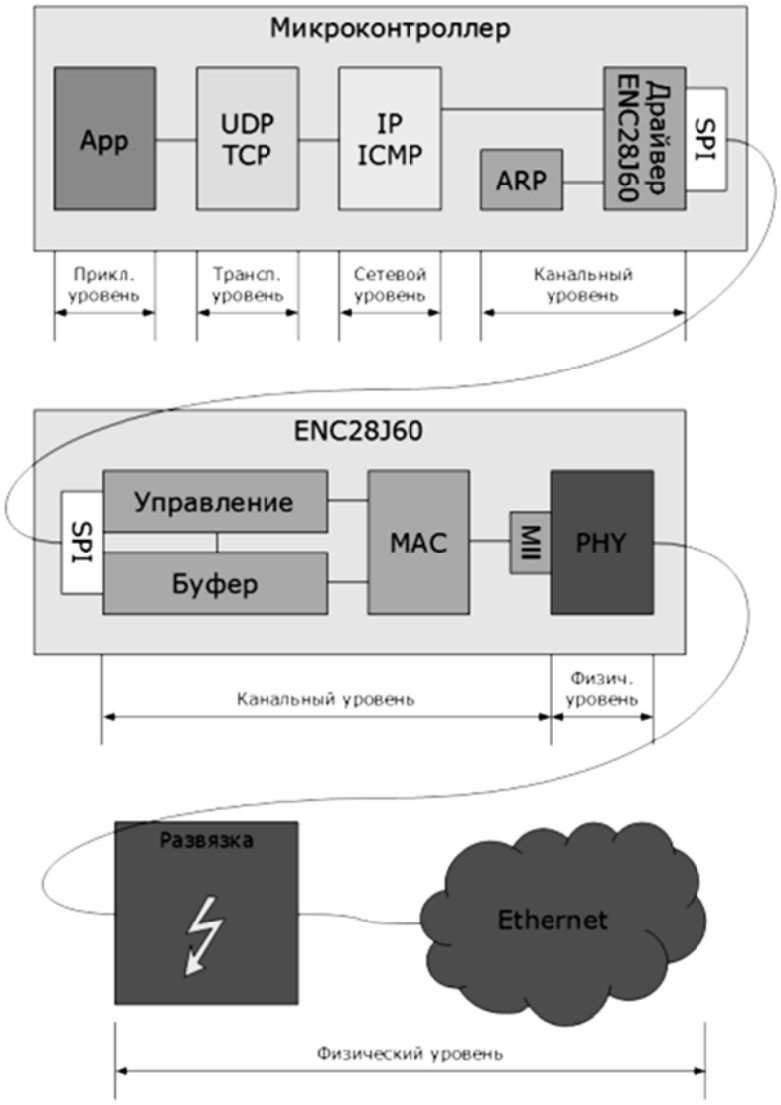

Рисунок 2.1 – Структурна схема

В якості обчислювального пристрою використовувався мікроконтролер фірми ATMEL – ATmega. Даний вибір зумовлений наявністю необхідної периферії, малого токовикористання та порівняно високих обчислювальних можливостей.

Ключовим вузлом системи є мікроконтролер. При виборі мікроконтролеру приділялася велика увага його надійності, вартості, універсальності, швидкодії й поширеності застосування.

2.2. Архітектура ENC28J60

На рис.2.2 представлені основні блоки ENC28J60.

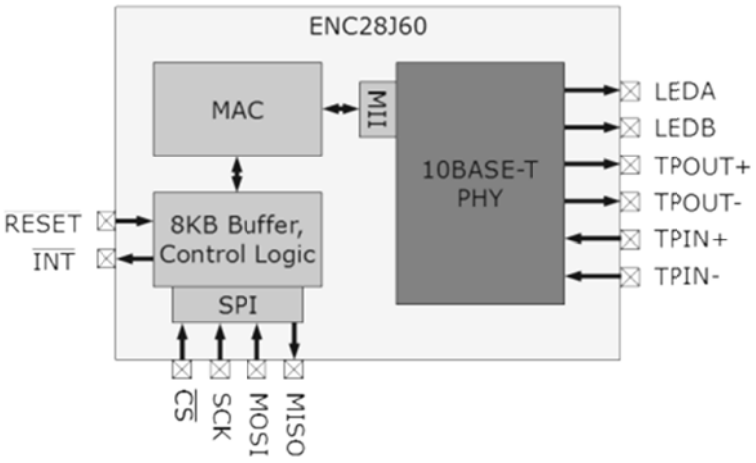

Рисунок 2.2 – Основні блоки ENC28J60

PHY – фізичний рівень (приймач, передавач, драйвери) – все, що необхідне для роботи з певним середовищем передачі даних, в даному випадку – з витою парою за стандартом 10BASE-T. Доступ до PHY відбувається виключно через MII – Medium Independent Interface. MII задуманий так, щоб наступний (канальний) рівень міг абстрагуватися від типу середовища передачі даних. PHY має свій набір 16-бітових регістрів (специфічних для середовища передачі даних), доступ до яких здійснюється через MII. MII – це всього лише набір регістрів, через які управляється PHY.

MAC (Medium Access Controller) – канальний рівень. У нього входить вся логіка, необхідна для відправки і прийому пакетів в мережі. MAC займається адресацією, розрахунком контрольної суми, фільтрацією пакетів, що приймаються, дозволом колізій (у напівдуплескному режимі). Обмінюється з наступним, мережевим рівнем готовими пакетами, а з фізичним – «сирими» байтами, що відправляються і приймаються.

Керуюча логіка обслуговує буфер, з якого MAC бере дані, що відправляються, і складає прийняті. Вся пам'ять в ENC28J60 ділиться на буфер для даних, керуючі регістри і регістри PHY.

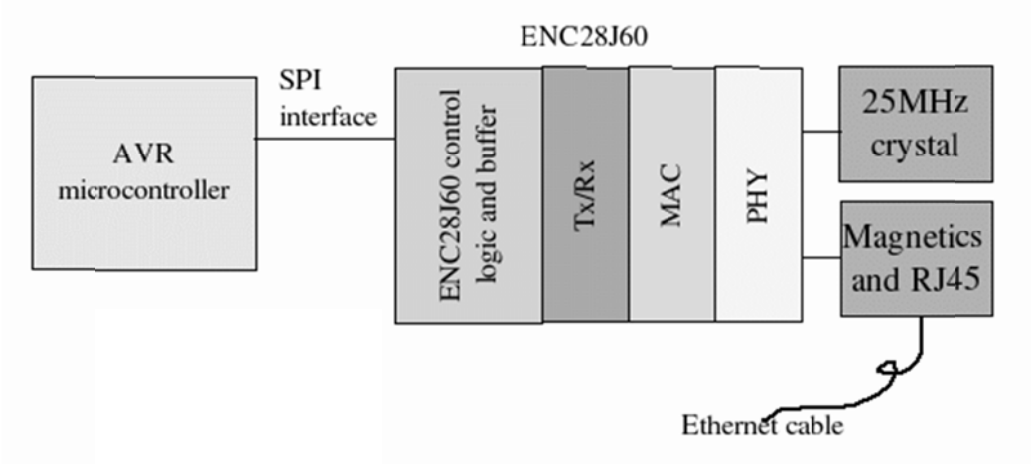

Рисунок 2.3 – Структурна схема підключення ENC28J60

Мікросхемі ENC28J60 потрібний трансформатор із співвідношенням витків 1:1, атестований для 10base- T. Існує декілька видів дуже хороших роз'ємів RJ45, що називаються «Magjack», які мають інтегрований трансформатор і додаткові світлодіоди. Стандартна схема включення ENC28J60 представлена на рис.2.4.

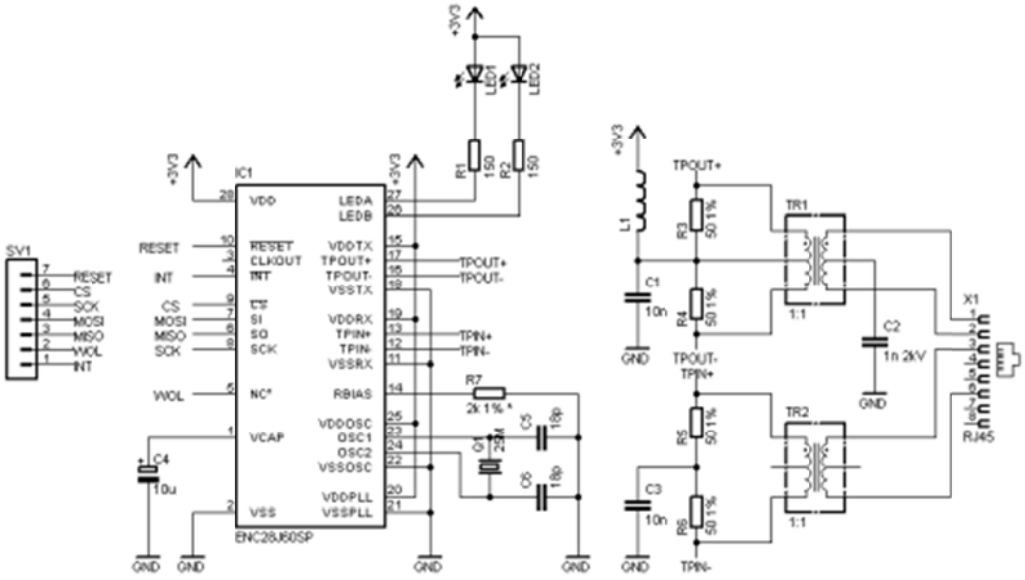

Рис.2.4. Стандартна схема включення ENC28J60

2.3. Схема електрична принципова

Принципова схема була розроблена з використання пакета Protues. Розроблена схема реалізує роботу з цифровими протоколами (1-Wire, TCP/IP), індикацію станів.

На рис. 2.5 представлена схема підключення датчика температури DS18B20 до мікроконтролера.

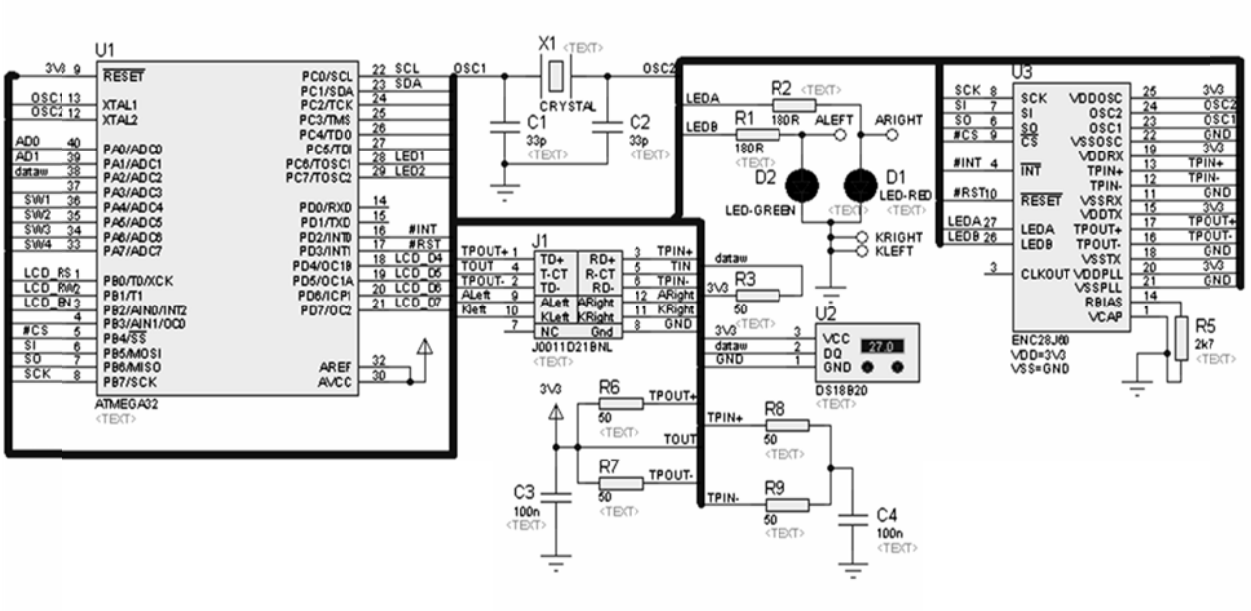

Рисунок 2.5 – Принципова схема

Живлення схеми – 3,3 В, але входи мікросхеми сумісні з 5-вольтовими TTL рівнями. Споживає схема 250 мА. Є режим «зниженого енергоспоживання», коли вся «силова» частина вимикається. VCAP – вихід вбудованого перетворювача на 2,5 В (саме така напруга використовується при передачі даних по мережевому кабелю). R5 (RBIAS) – резистор для балансування номіналом 2,7 кОм з допуском 1%. TR1 і TR2 – спеціальні ethernet-фільтри (Ethernet magnetics). Являють собою систему з декількох котушок на феритових кільцях. Зазвичай випускаються у вигляді готових складань (обидва фільтри в одному корпусі, сумісному з DIP-16). Потрібні вони для розв'язки, захисту від статики (мережевий кабель може мати довжину до 100 м. ENC28J60 автоматично визначає полярність підключених світлодіодів. Причому полярність світлодіоду, підключеного до виведення LEDB впливає на

дуплексний режим роботи мікросхеми. Якщо світлодіод підключений як показано на схемі ENC28J60 ініціалізувався в повнодуплексному режимі. Відповідно, якщо підключений анодом – те в напівдуплексному. Якщо світлодіод не підключений – стан не визначений. Втім, дуплексний режим можна змінити при ініціалізації.

2.4. Розробка друкованої плати

Печатна плата в зборі з елементами є кінцевим продуктом проектування системи моніторингу. Плата виконує роль несучої конструкції пристрою, на якій розташовуються всі елементи. Використання P-CAD дозволяє позбавитися помилок, полегшує процес розташування елементів на платі та з'єднання між ними. Крім того, з'являється можливість виготовлення плати на автоматичному технологічному обладнанні.

Після того, як визначені розміри плати і монтаж елементів, потрібно виготовити плату. Існує багато способів виготовлення печатних плат. Серед них є наступні: гальванохімічний метод, метод хімічного травлення фольгованого діелектрика, метод переносу схеми із запресуванням в ізоляційну основу та ін. Для виготовлення зразка системи я використав метод хімічного травлення фольгованого діелектрика. Суть метода полягає в нанесенні малюнка схеми кислотостійкою фарбою з послідуючим витравленням фольги з пробільних ділянок. Цей метод забезпечує отримання чітких ліній провідників печатної плати, потребує менших трудових затрат в порівнянні з іншими методами, має високу роздільну здатність (до 10 ліній на 1 мм), легко піддається автоматизації. До недоліків цього метода можна віднести хімічну дію розчину на ізоляційну основу, що погіршує діелектричні властивості матеріалу.

В якості матеріла основи печатної плати був використаний склотекстоліт. Технологічний процес виготовлення печатної плати розділений на наступні основні операції: підготовка поверхні (знежирення), нанесення малюнку

кислотостійкою речовиною (фарба), хімічне травлення, видалення кислотостійкого шару, механічна обробка плати та нанесення захисного шару (покриття лаком).

В якості реактиву для травлення використовується водний розчин технічного хлорного заліза. Підвищення температури розчину прискорює процес травлення. Однак травлення хлорним залізом має ряд недоліків. Фенольні смоли, які містять печатні плати, забруднюються залізом, що призводить до погіршення ізоляційних властивостей гетинаксових основ. Ускладнена також регенерація міді, оскільки в склад використаного розчину входять два метали: мідь та залізо. В якості травлячого розчину може бути використаний водний розчин хлорної міді, який містить соляну кислоту. При цьому збільшується час травлення плати. Після механічної обробки плата промивається і висушується.

Для забезпечення достатньої жорсткості при впливі на неї механічних зусиль, друкована плата повинна бути прямокутної форми зі співвідношенням сторін не більш 1:2. Жолоблення плати не повинне перевищувати 1,3 мм по всій довжині плати. Для виготовлення друкованої плати необхідно використовувати позитивний спосіб.

Для розробки друкованої плати пристрою використовувався програмний продукт P-CAD 2006 SP2 PCB. Спочатку були розроблені посадочні місця під кожний елемент, та сформована бібліотека. Потім були рознесені та розташовані елементи на платі та проведена розводка електричних зв’язків у ручному режимі.

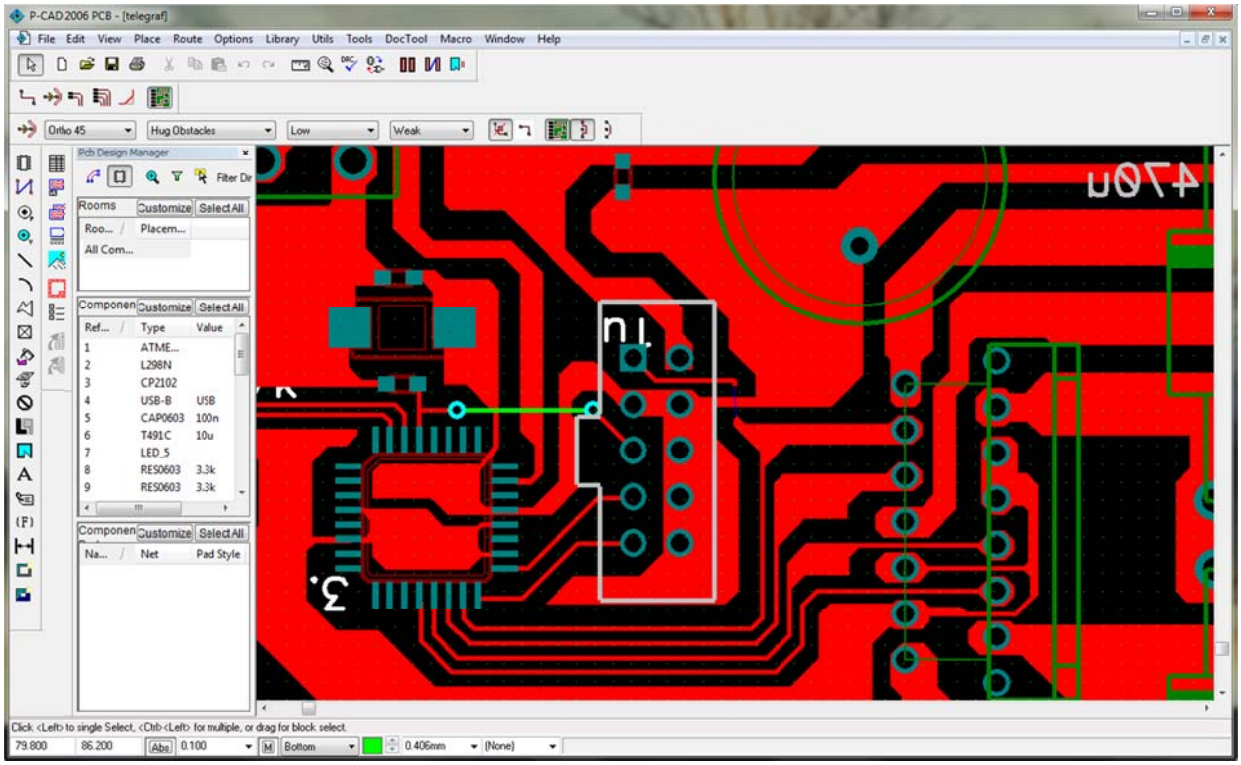

Рисунок 2.6 – Середовище розробки друкованої плати

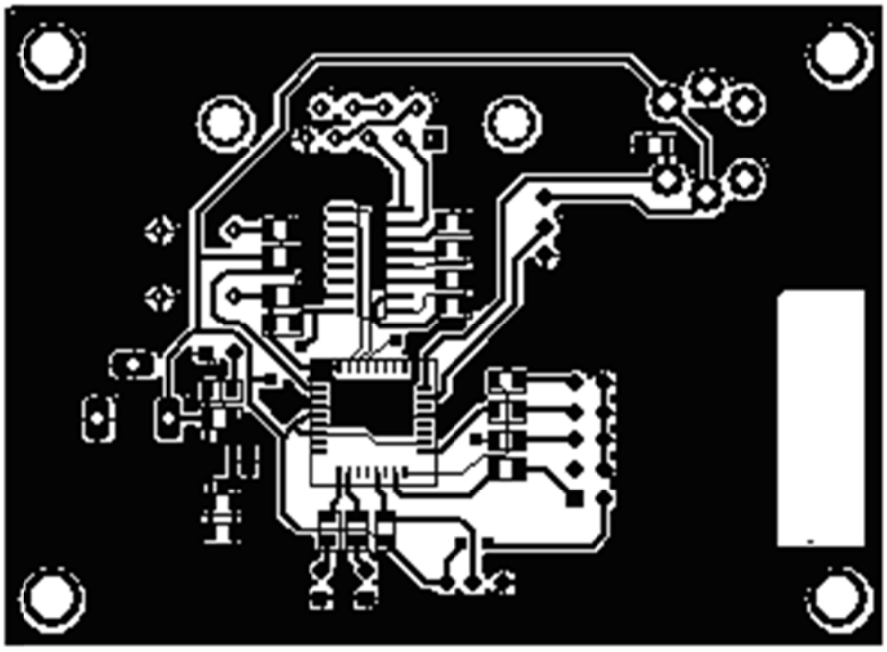

Рисунок 2.7 – Верхній шар друкованої плати

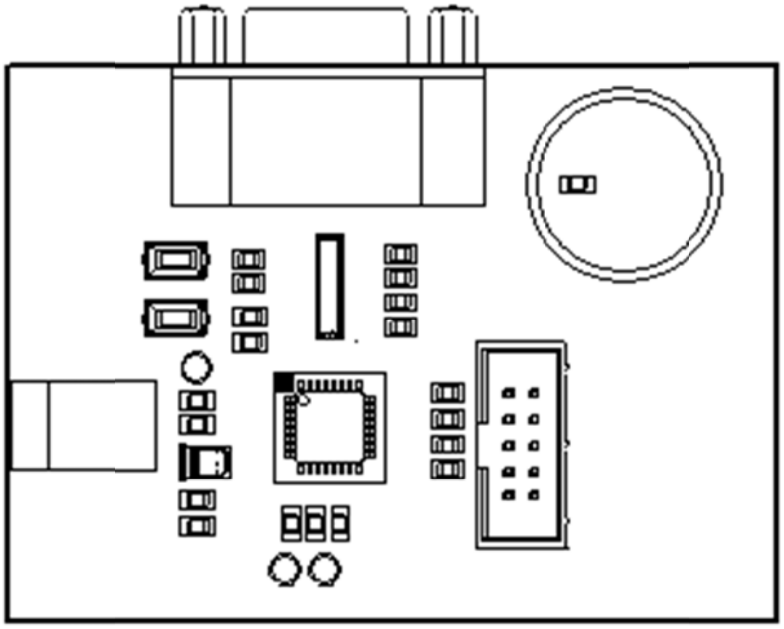

Рисунок 2.8 – Загальне розміщення компонентів друкованої плати

2.5. Висновки

У розділі розглянута апаратна частина системи моніторингу температури, виконана розробка структурної та принципової схеми системи моніторингу температури і розробка складального креслення друкованої плати.

Використання сучасних технологій на основі мікропроцесорних систем відкриває нові можливості в підвищенні якості контролю температури, дозволяє прогнозувати та запобігати некоректним ситуаціям на основі статистичної бази, яка утворюється в процесі роботи системи.

Проведене тестування електронної системи моніторингу температури показало дієздатність апаратної частини пристрою, експериментальна перевірка працездатності виявила високі експлуатаційні показники в цільовому застосуванні пристрою, досить широкі можливості до вдосконалення, налагодження системи і подальшого впровадження її в сферу конкурентоспроможних систем моніторингу.

					172.4397ст3.КР.ПЗ.Р3	
Змн.	*Арк.*	*№ документу*	*Підпис*	*Дата*		
Розробник		Воскобойніков В. І.			РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ МІКРОПРОЦЕСОРНОЇ СИСТЕМИ МОНІТОРИНГУ ТЕМПЕРАТУРИ ПО	*Літ.* / *Арк.* 34 / *Аркушів*
Консультант		Рябенький В. М.				
Н. контр.		Добрінова Л. П.				
Керівник		Іхсанов Ш. М.				НУК імені адмірала Макарова
Зав. каф.		Рябенький В. М.				

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ МІКРОПРОЦЕСОРНОЇ СИСТЕМИ МОНІТОРИНГУ ТЕМПЕРАТУРИ ПО МЕРЕЖІ INTERNET

У сучасному проектуванні спостерігається тенденція збільшення переносу частини завдань із апаратних засобів на програмні. Цьому сприяє бурхливий розвиток мікропроцесорних систем, розвиненого програмного забезпечення, а також зниження їхньої вартості. Програмно-апаратна система поєднує в собі всі необхідні функціональні вузли і програмні компоненти для ефективної роботи оператора. Програми з розвиненими засобами візуального інтерфейсу не потребують від користувача тривалого навчання. А працюючи в комплексі з іншими програмами здатні вирішувати найширше коло науково-технічних і прикладних завдань.

У даному розділі вирішуються завдання створення програмного забезпечення для мікроконтролера. Для цих цілей використається програмне середовище WinAVR. Програма для мікроконтролера складена мовою AVR C++. Використовувався компілятор GCC.

3.1. Протокол обміну даними по інтерфейсу 1-Wire

Однопровідна шина оперує з TTL-рівнями, тобто логічна одиниця представлена рівнем напруги близько 5 В, а логічний нуль – напругою поблизу 0 В. У вихідному стані на лінії присутній рівень логічної одиниці, що забезпечується резистором номіналом близько 5 кОм.

Ініціатором обміну по однопровідній шині завжди виступає майстер. Всі пересилання починаються із процесу ініціалізації. Ініціалізація виконується в наступній послідовності (рис. 3.1).

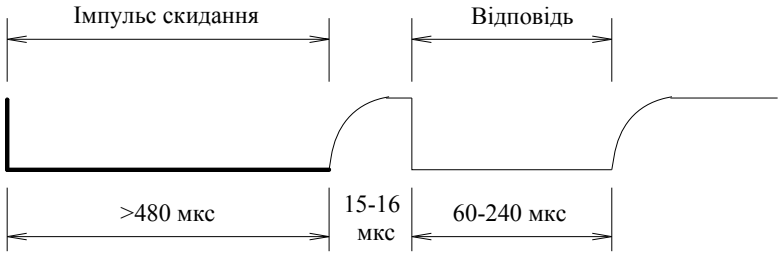

Рисунок 3.1 – Процес ініціалізації

Майстер посилає імпульс скидання (reset pulse) – сигнал низького рівня тривалістю не менше 480 мкс. За імпульсом скидання відбувається відповідь підлеглого пристрою (presence pulse) – сигнал низького рівня тривалістю 60..240 мкс, що генерується через 15..60 мкс після завершення імпульсу скидання.

Відповідь підлеглого пристрою для головного пристрою означає, що на шині присутній термометр і він готовий до обміну даними. Після того, як майстер виявив відповідь, він може передати термометру одну з команд. Передача ведеться шляхом формування майстром спеціальних часових інтервалів (time slots). Кожен часовий інтервал служить для передачі одного біта. Першим передається молодший біт. Інтервал починається імпульсом низького рівня, тривалість якого лежить у межах 1..15 мкс. Оскільки перехід з одиниці в нуль менш чутливий до ємності шини (він формується відкритим транзистором, у той час як перехід з нуля в одиницю формується резистором), саме цей перехід використовують однопровідні пристрої для синхронізації з майстром. У підлеглому пристрої запускається схема часової затримки, що визначає момент зчитування даних. Номінальне значення затримки дорівнює 30 мкс, однак воно може коливатися в межах 15..60 мкс. За імпульсом низького рівня видається черговий біт для передачі. Він повинен утримуватися майстром на шині протягом 60..120 мкс від початку інтервалу. Часовий інтервал завершується переводом шини в стан високого рівня на час не менший 1 мкс. Слід відзначити, що обмеження на цей час не накладається. Аналогічним чином формуються часові інтервали для всіх інших біт (рис. 3.2).

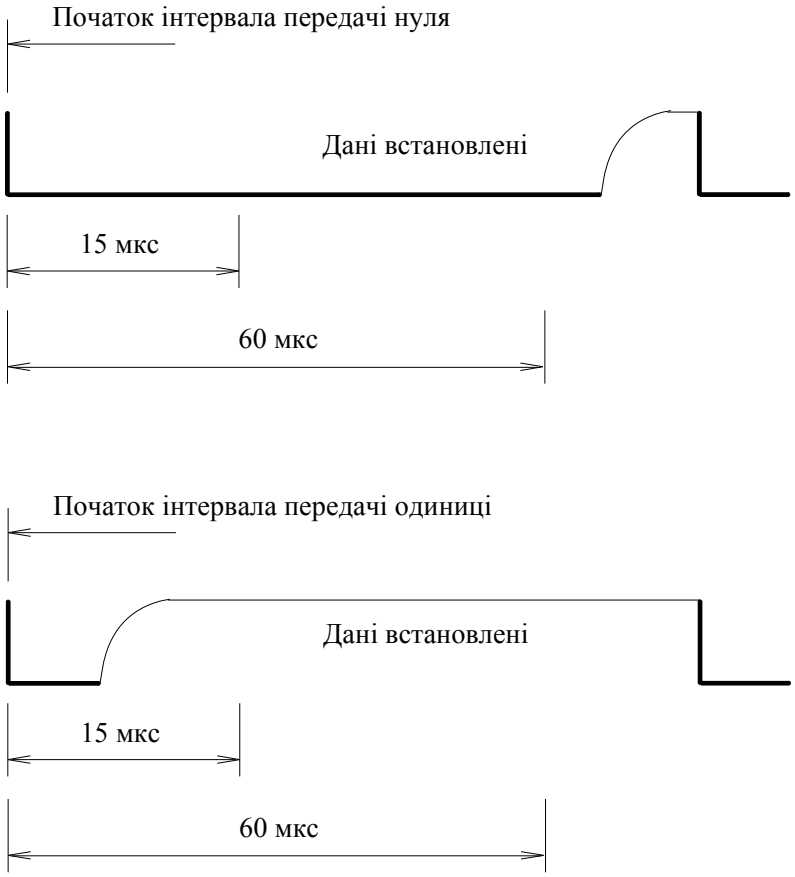

Рисунок 3.2 – Формування майстром часових інтервалів для передачі команд

Першою командою, яку повинен передати майстер для DS18B20 після ініціалізації, є одна з команд функцій ПЗП. Усього DS18B20 має 5 команд функцій ПЗП:

- Read ROM [33h]. Ця команда дозволяє прочитати вміст ПЗП. У відповідь на цю команду DS18B20 передає 8-бітний код сімейства (10h), потім 48-бітний серійний номер, а потім 8-бітне значення CRC для перевірки правильності прийнятої інформації;

- Match ROM [55h]. Ця команда дозволяє адресувати на шині конкретний термометр. Після цієї команди майстер повинен передати потрібний 64-бітний код, і тільки той термометр, що має такий код, буде «відгукуватися» до наступного імпульсу скидання;

- Skip ROM [CCh]. Ця команда дозволяє пропустити процедуру порівняння серійного номера, і тим самим заощадити час у системах, де на шині є всього один пристрій;

- Search ROM [F0h]. Ця досить складна у використанні команда дозволяє визначити серійні номери всіх термометрів, що присутні на шині.;

- Alarm Search [ECh]. Ця команда аналогічна попередній, але «відгукуватися» будуть тільки ті термометри, у яких результат останнього виміру температури виходить за встановлені межі TH та TL.

Оскільки в нашому прикладі всього один пристрій, найбільш підходящою є функція Skip ROM. Крім неї ще може бути корисною функція Read ROM, що дозволяє ідентифікувати підключений на шину пристрій по його коду сімейства та серійному номеру.

При прийомі даних від підлеглого пристрою часові інтервали для прийнятих біт також формує майстер. Інтервал починається імпульсом низького рівня тривалістю 1..15 мкс. Потім майстер повинен звільнити шину, щоб дати можливість термометру вивести біт даних. По переходу з одиниці в нуль DS18B20 виводить на шину біт даних і запускає схему часової затримки, що визначає, як довго біт даних буде присутній на шині. Цей час лежить у межах 15..60 мкс. Для того щоб дані на шині, яка завжди має деяку ємність, гарантовано встановилися, потрібний деякий час. Тому момент зчитування даних майстром повинен відстояти якнайдалі, але не більше ніж на 15 мкс від початку часового інтервалу (рис. 3.3).

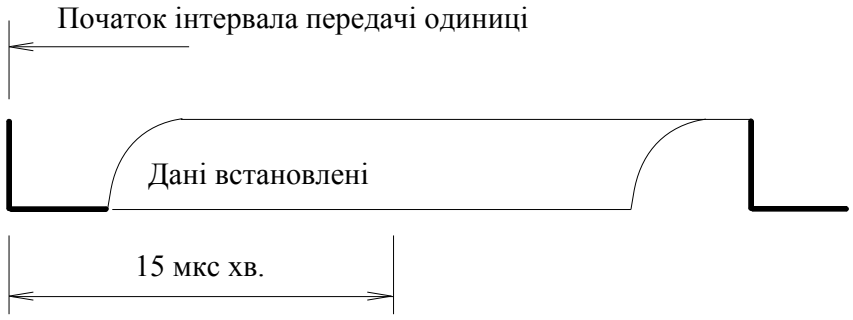

Рисунок 3.3 – Момент зчитування даних майстром

Прийом байта починається з молодшого біта. Спочатку передається байт коду сімейства. За кодом сімейства передається 6 байт серійного номера, починаючи з молодшого. Потім пердається байт контрольної суми (CRC). В обчисленні байта контрольної суми беруть участь перші 7 байт, або 56 переданих біт. Після прийому даних майстер повинен обчислити контрольну суму та порівняти значення з CRC. Якщо ці значення збігаються – прийом даних пройшов без помилок.

Після обробки однієї з команд функцій ПЗП, DS18B20 здатний сприймати ще кілька команд:

- Write Scratchpad [4Eh]. Ця команда дозволяє записати дані в проміжний ОЗП DS18B20;

- Read Scratchpad [BEh]. Ця команда дозволяє зчитувати дані із проміжного ОЗП;

- Copy Scratchpad [48h]. Ця команда копіює байти TH і TL із проміжного ОЗП в енергонезалежну пам'ять. Ця операція вимагає близько 10 мс;

- Convert T [44h]. Ця команда запускає процес перетворення температури;

- Recall E2 [B8h]. Ця команда діє зворотним чином відносно команди Copy Scratchpad, тобто вона дозволяє зчитувати байти TH і TL з енергонезалежної пам'яті в проміжний ОЗП. При включенні живлення ця команда виконується автоматично;

- Read Power Supply [B4h]. Ця команда дозволяє перевірити, чи використає DS18B20 паразитне живлення. Справа в тому, що DS18B20 можна підключати всього за допомогою двох проводів. У цьому випадку для живлення використовується лінія даних.

При використанні DS18B20 тільки для виміру температури потрібні всього дві із цих команд: Convert T і Read Scratchpad. Послідовність дій при вимірі температури повинна бути наступною:

1) посилаємо імпульс скидання і приймаємо відповідь термометра;

2) посилаємо команду Skip ROM [CCh];

3) посилаємо команду Convert T [44h];

4) формуємо затримку мінімум 750 мс;

5) посилаємо імпульс скидання і приймаємо відповідь термометра;

6) посилаємо команду Skip ROM [CCh];

7) посилаємо команду Read Scratchpad [BEh];

8) читаємо дані із проміжного ОЗП (8 байт) і CRC;

9) перевіряємо CRC, і якщо дані зчитані вірно, обчислюємо температуру.

3.2. Структура IP-пакету

IP-пакети складаються з даних верхнього рівня та IP-заголовку. За специфікацією протоколу, пакет має бути не більший за 65535 бітів (з заголовком і даними включно).

Версія (Version) – 4-бітове поле, що описує використовувану версію протоколу IP. Всі пристрої зобов'язані використовувати протокол IP однієї версії, пристрій що використовує іншу версію буде відкидати пакети.

Довжина IP-заголовку (IP header Length – HLEN) – 4-бітове поле, що описує довжину заголовку пакету в 32-бітових блоках. Це значення – це повна довжина заголовку з врахуванням двох полів змінної довжини.

Тип обслуговування (Type of Service – TOS) – 8-бітове поле, що вказує на степінь важливості інформації, яка присвоєна протоколом верхнього рівня.

Загальна довжина (Total Length) – 16-бітове поле, що описує довжину пакету в байтах, із заголовком і даними включно. Для того щоб вирахувати довжину блока даних, потрібно від повної довжини відняти значення поля HLEN.

Ідентифікація (Identification) – шістнадцятибітове поле, що зберігає ціле число, яке описує даний пакет. Це число являє собою послідовний номер.

Флаги (Flags) – 3-бітове поле, в якому два молодших біта контролюють фрагментацію пакетів. Перший біт визначає чи був пакет фрагментовано, а другий чи є цей пакет останнім фрагментом в серії фрагментів.

Зміщення фрагментації (Fragment Offset) – 13-бітове поле, що допомагає зібрати разом фрагменти пакетів. Це поле дозволяє використовувати 16 бітів в сумі для флагів фрагментації.

Час життя (Time-to-Live – TTL) – 8-бітове поле – лічильник, в якому зберігаються послідовно зменшуване значення кількості пройдених вузлів (роутетів, що їх ще іноді в цьому випадку називають хопами(hops)) на шляху до місця призначення. У випадку коли лічільник пройдених хопів дорівнюватиме нулю – пакет буде відкинуто, таким чином попереджується нескінченна циклічна пересилка пакетів.

Протокол (Protocol) – 8-бітове поле, що вказує на те, який протокол верхнього рівня отримає пакет, після завершення обробки IP-протоколом. Наприклад TCP або UDP.

Контрольна сума заголовку (Header Checksum) – 16-бітове поле, що допомагає перевірити цілісність заголовку пакету.

IP-адреса відправника (Source IP address) (адресант, сорс, відправник) – 32-бітове поле, що зберігає IP-адресу вузла-відправника.

IP-адреса отримувача (Destination IP adress) (адресат, дест, отримувач) – 32-бітове поле, що зберігає адресу вузла призначення (отримувача).

Опції (Options) – поле змінної довжини, що дозволяє протоколу IP реалізувати підтримку різних опцій, зокрема засобів безпеки.

Підкладка (Padding) – поле, що використовується для вставки додаткових нулів, для гарантування кратності IP-заголовку 32 бітам.

Дані (Data) – поле змінної довжини (64 Кбіт макс.), що зберігає інформації для верхніх рівнів.

Таблиця 3.1. Структура пакету даних

Біти 0-3	4-7	8-15	16-18	19-23	24-31
Версія	HLEN	Тип обслуговування	Загальна довжина		
Ідентифікація			Флаги	Зміщення фрагментації	
Час життя		Протокол	Контрольна сума заголовку		
IP-адреса відправника					
IP-адреса отримувача					
Опції					Додаток
Дані (65535 мінус заголовок)					
...					

IP-пакет складається з даних протоколу верхнього рівня і заголовку, що має описану вище структуру. Хоча основною частиною заголовку є адреси відправника і призначення, саме інші частини заголовку роблять протокол таким надійним і гнучким. Інформація, що зберігається в полях заголовку задає дані пакету і призначена для протоколів верхніх рівнів

3.3. Структура протоколу HTTP

HTTP – протокол прикладного рівня, схожими на нього є FTP і SMTP. Обмін повідомленнями йде за звичайною схемою «запит-відповідь». Для ідентифікації ресурсів HTTP використовує глобальні URI. На відміну від багатьох інших протоколів, HTTP не зберігає свого стану. Це означає відсутність збереження проміжного стану між парами «запит-відповідь». Компоненти, що використовують HTTP, можуть самостійно здійснювати збереження інформації про стан пов'язаний з останніми запитами і відповідями. Браузер, що посилає запити, може відстежувати затримки відповідей. Сервер може зберігати IP-адреси і заголовки запитів останніх клієнтів. Проте згідно з протоколом клієнт та сервер не мають бути обізнаними про попередні запити і відповіді, в протоколі не передбачена внутрішня підтримка стану і він не ставить таких вимог до клієнта та сервера.

Кожен запит/відповідь складається з трьох частин:

- стартовий рядок;

- заголовки;

- тіло повідомлення, що містить дані запиту, запитаний ресурс або опис проблеми, якщо запит не виконано.

Обов'язковим мінімумом запиту є стартовий рядок. Починаючи з HTTP/1.1 обов'язковим став заголовок Host: (щоб розрізнити кілька доменів, які мають одну і ту же IP-адресу).

Запит

Стартові рядки розрізняються для запиту і відповіді. Рядок запиту виглядає так:

‹Метод› ‹URI› HTTP/‹Версія› ,

де *‹Метод›* можливо:

OPTIONS – повертає методи HTTP, які підтримуються сервером. Цей метод може служити для визначення можливостей веб-сервера;

GET – запрошує вміст вказаного ресурсу. Запитаний ресурс може приймати параметри (наприклад, пошукова система може приймати як параметр шуканий рядок). Вони передаються в рядку URI (наприклад: http://www.example.net/resource?param1=value1&param2=value2). Згідно зі стандартом HTTP, запити типу GET вважаються ідемпотентними – багаторазове повторення одного і того ж запиту GET повинне приводити до однакових результатів (за умови, що сам ресурс не змінився за час між запитами). Це дозволяє кешувати відповіді на запити GET;

HEAD – аналогічний методу GET, за винятком того, що у відповіді сервера відсутнє тіло. Це корисно для витягання мета-інформації, заданої в заголовках відповіді, без пересилання всього вмісту. Зокрема, клієнт чи проксі, перевіривши заголовок Last-Modified: (останній час модифікації), таким чином може переконатися, що сторінка на сервері не змінилася від часу попереднього запиту;

POST – передає призначені для користувача дані (наприклад, з HTML-форми) заданому ресурсу. Наприклад, в блогах відвідувачі зазвичай можуть вводити свої коментарі до записів в HTML-форму, після чого вони переда-

ються серверу методом POST, і він поміщає їх на сторінку. При цьому передані дані (у прикладі з блогами – текст коментаря) включаються в тіло запиту. На відміну від методу GET, метод POST не вважається ідемпотентним, тобто багатократне повторення одних і тих же запитів POST може повертати різні результати (наприклад, після кожного відправлення коментаря з'являтиметься одна копія цього коментаря);

PUT – завантажує вказаний ресурс на сервер;

DELETE – видаляє вказаний ресурс;

TRACE – повертає отриманий запит так, що клієнт може побачити, що проміжні сервери додають або змінюють в запиті;

CONNECT – для використання разом з проксі-серверами, які можуть динамічно перемикатися в тунельний режим SSL.

В основному використовуються методи GET і POST.

Відповідь сервера

Перший рядок відповіді виглядає так:

HTTP/‹Версія› ‹Код статусу› ‹Опис статусу›

Найтиповіші статуси:

- 200 OK – запит виконаний успішно;
- 403 Forbidden – доступ до запитаного ресурсу заборонений;
- 404 Not Found – запитаний ресурс не знайдений.

3.4. Алгоритм роботи системи

Для спрощення розробки системи моніторингу був розроблений алгоритм роботи системи він дозволяє уникнути тотожних ділянок. Програмне забезпечення було розроблено в середі WinAVR 2009 з використанням компілятора GCC. На основі даного алгоритму була написана програма для мікроконтролера. Програмування та налагодження роботи контролеру відбувалося за допомогою програматору JTAG. Блок-схему алгоритму програми наведено на рис.3.4. Лістинг програми приводиться в додатку А.

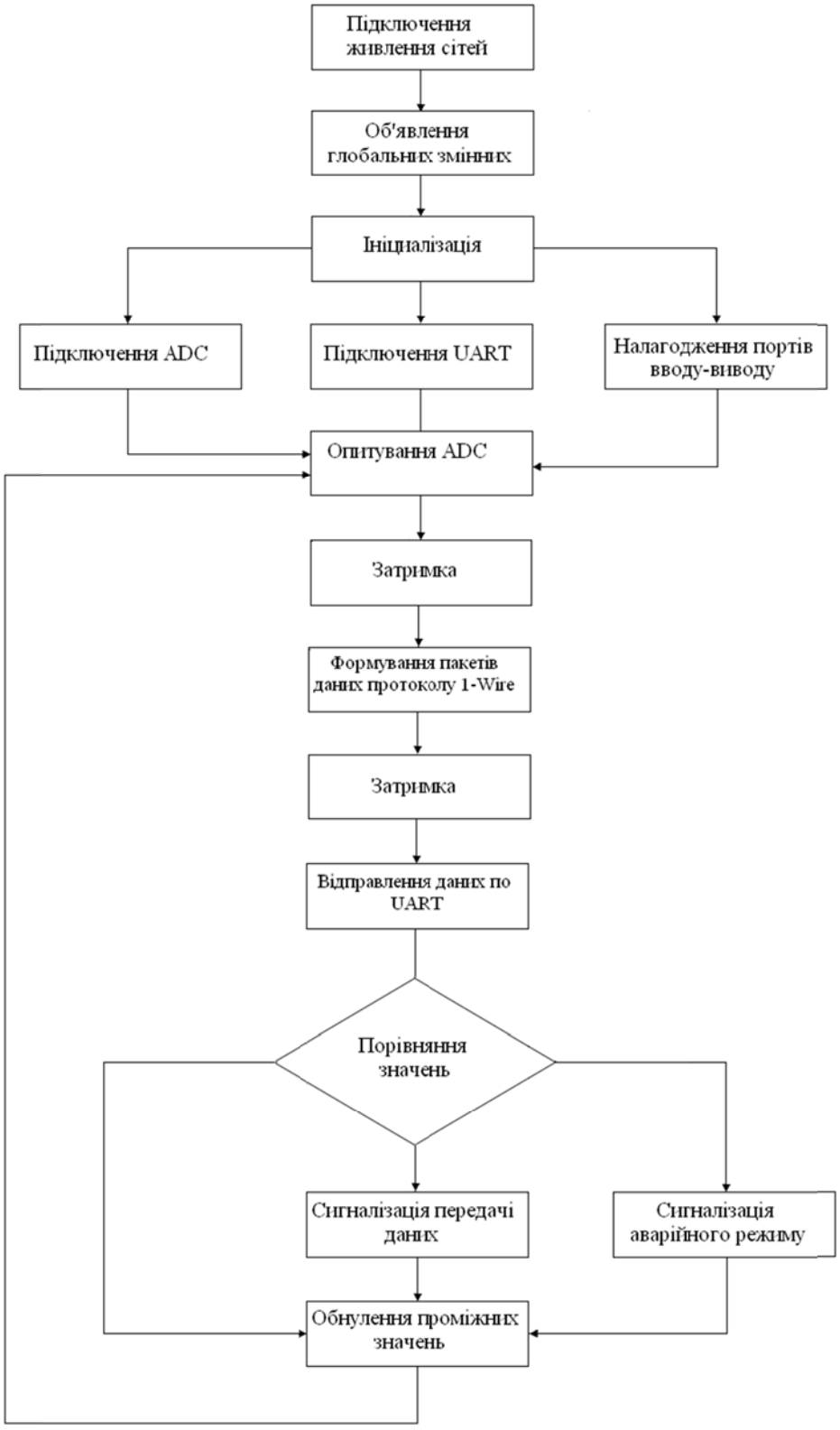

Рисунок 3.4 – Блок-схема алгоритму програми для мікроконтролера

3.5. Керуючі регістри ENC28J60

Керуючі регістри діляться на 4 банка. Кожен банк має розмір в 32 регістри, причому останні 5 комірок (0x1b..0x1f) завжди вказують на одні і ті ж регістри, незалежно від того, який банк вибраний.

Таблиця 3.2 – Керуючі регістри ENC28J60

	Банк 0	Банк 1	Банк 2	Банк 3
0x00	ERDPTL	EHT0	MACON1	MAADR1
0x01	ERDPTH	EHT1	MACON2	MAADR0
0x02	EWRPTL	EHT2	MACON3	MAADR3
0x03	EWRPTH	EHT3	MACON4	MAADR2
0x04	ETXSTL	EHT4	MABBIPG	MAADR5
0x05	ETXSTH	EHT5	—	MAADR4
0x06	ETXNDL	EHT6	MAIPGL	EBSTSD
0x07	ETXNDH	EHT7	MAIPGH	EBSTCON
0x08	ERXSTL	EPMM0	MACLCON1	EBSTCSL
0x09	ERXSTH	EPMM1	MACLCON2	EBSTCSH
0x0A	ERXNDL	EPMM2	MAMXFLL	MISTAT
0x0B	ERXNDH	EPMM3	MAMXFLH	—
0x0C	ERXRDPTL	EPMM4	Reserved	—
0x0D	ERXRDPTH	EPMM5	MAPHSUP	—
0x0E	ERXWRPTL	EPMM6	Reserved	—
0x0F	ERXWRPTH	EPMM7	—	—
0x10	EDMASTL	EPMCSL	Reserved	—
0x11	EDMASTH	EPMCSH	MICON	—
0x12	EDMANDL	—	MICMD	EREVID
0x13	EDMANDH	—	—	—
0x14	EDMADSTL	EPMOL	MIREGADR	—
0x15	EDMADSTH	EPMOH	Reserved	ECOCON
0x16	EDMACSL	EWOLIE	MIWRL	Reserved
0x17	EDMACSH	EWOLIR	MIWRH	EFLOCON
0x18	—	ERXFCON	MIRDL	EPAUSL
0x19	—	EPKTCNT	MIRDH	EPAUSH
0x1A	Reserved	Reserved	Reserved	Reserved
0x1B	EIE	EIE	EIE	EIE
0x1C	EIR	EIR	EIR	EIR
0x1D	ESTAT	ESTAT	ESTAT	ESTAT
0x1E	ECON2	ECON2	ECON2	ECON2
0x1F	ECON1	ECON1	ECON1	ECON1

Основна частина регістрів має префікс E (Ethernet). Регістри MAC – з префіксом MA, регістри MII – з префіксом MI. Регістри можна розділити на функціональні групи.

	Банк 0	Банк 1	Банк 2	Банк 3
0x00	ERDPTL	EHT0	MACON1	MAADR1
0x01	ERDPTH	EHT1	MACON2	MAADR0
0x02	EWRPTL	EHT2	MACON3	MAADR3
0x03	EWRPTH	EHT3	MACON4	MAADR2
0x04	ETXSTL	EHT4	MABBIPG	MAADR5
0x05	ETXSTH	EHT5	–	MAADR4
0x06	ETXNDL	EHT6	MAIPGL	EBSTSD
0x07	ETXNDH	EHT7	MAIPGH	EBSTCON
0x08	ERXSTL	EPMM0	MACLCON1	EBSTCSL
0x09	ERXSTH	EPMM1	MACLCON2	EBSTCSH
0x0A	ERXNDL	EPMM2	MAMXFLL	MISTAT
0x0B	ERXNDH	EPMM3	MAMXFLH	–
0x0C	ERXRDPTL	EPMM4	Reserved	–
0x0D	ERXRDPTH	EPMM5	MAPHSUP	–
0x0E	ERXWRPTL	EPMM6	Reserved	–
0x0F	ERXWRPTH	EPMM7	–	–
0x10	EDMASTL	EPMCSL	Reserved	–
0x11	EDMASTH	EPMCSH	MICON	–
0x12	EDMANDL	–	MICMD	EREVID
0x13	EDMANDH	–	–	–
0x14	EDMADSTL	EPMOL	MIREGADR	–
0x15	EDMADSTH	EPMOH	Reserved	ECOCON
0x16	EDMACSL	EWOLIE	MIWRL	Reserved
0x17	EDMACSH	EWOLIR	MIWRH	EFLOCON
0x18	–	ERXFCON	MIRDL	EPAUSL
0x19	–	EPKTCNT	MIRDH	EPAUSH
0x1A	Reserved	Reserved	Reserved	Reserved
0x1B	EIE	EIE	EIE	EIE
0x1C	EIR	EIR	EIR	EIR
0x1D	ESTAT	ESTAT	ESTAT	ESTAT
0x1E	ECON2	ECON2	ECON2	ECON2
0x1F	ECON1	ECON1	ECON1	ECON1

Основное
Указатели буфера
DMA
Фильтрация пакетов
MAC-адрес
Регистры MAC
Регистры MII
Управление потоком
Прерывания
Идентификатор ревизии (e.g. 0x06 = ревизия 7)
Управление ножкой CLKOUT
«Встроенный тест»

Рисунок 3.5 – Функціональні групи регістрів

ECON1

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TXRST	**RXRST**	**DMAST**	**CSUMEN**	**TXRTS**	**RXEN**	**BSEL1**	**BSEL0**
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

BSEL1:BSEL0 – вибір банка регістрів.

RXEN – дозволяє прийом даних.

TXRTS – дозволяє відправку пакета (автоматично скидається після того, як відправка пакета буде завершена).

ECON2

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AUTOINC	**PKTDEC**	**PWRSV**	–	**VRPS**	–	–	–
R/W-1	W-0	R/W-0	U-0	R/W-0	U-0	U-0	U-0

VRPS – вирішує переведення стабілізатора живлення в економічний режим при включенні режиму зниженого енергоспоживання (біт PWRSV). Даний біт можна встановити при ініціалізації і забути про нього.

PWRSV – включає режим зниженого енергоспоживання. Перш ніж встановлювати цей біт, слід заборонити прийом нових пакетів і переконатися що прийом даних завершений. Після виходу з режиму зниженого енергоспоживання, потрібно почекати 1 мс, щоб PHY увійшов до робочого режиму.

PKTDEC – при установці цього біта значення лічильника пакетів зменшується на 1.

AUTOINC – включає автоматичне інкрементування покажчиків читання і запису буфера (для зручності послідовного читання і запису даних). Цей біт встановлений після скидання, й не треба його чіпати.

ESTAT	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	INT	r	r	LATECOL	—	RXBUSY	TXABRT	CLKRDY
	R-0	R/C-0	R-0	R/C-0	U-0	R-0	R/C-0	R/W-0

TXABRT – прапор завершення передачі з помилкою.

RXBUSY – ознака роботи приймача (встановлений, якщо приймаються дані).

Регістр EPKTCNT – лічильник прийнятих пакетів. Автоматично інкрементується при успішно прийнятому пакеті. Зменшується вручну, установкою біта PKTDEC в регістрі ECON2.

Типова послідовність ініціалізації ENC28J60 виглядає приблизно так:

1. Налаштовуємо розмір FIFO для прийому даних (ERXST, ERXND), ініціалізуємо вказівник для читання даних з FIFO (ERXRDPT).

2. Налаштовуємо фільтрацію вхідних пакетів. За замовчуванням, ENC28J60 пропускає пакети, що приходять на нашу MAC-адресу і широкомовні пакети. В принципі, можна так і залишити.

Налаштовуємо MAC:

1. Очищаємо MACON2.MARST щоб зняти скидання MAC.

2. Встановлюємо MACON1.MARXEN щоб дозволити прийом даних MAC.

3. Встановлюємо MACON1.RXPAUS і MACON1.TXPAUS для включення апаратного управління потоком.

4. Налаштовуємо біти PADCFG, TXCRCEN в MACON3. Для більшості додатків підійде вирівнювання пакету до 60 байт і автоматичне додавання контрольної суми.

5. Встановлюємо максимальний розмір фрейму в регістрах MAMXF.

6. Встановлюємо розмір проміжку між фреймами в регістрах MABBIPG, MAIPGL і MAIPGH.

7. Встановлюємо MAC-адресу в регістрах MAADR.

Налаштовуємо PHY:

1. Включаємо біт PHCON2.HDLDIS, якщо не хочемо отримувати свої пакети в напівдуплексному режимі.

2. Вибираємо як на різні події реагуватимуть світлодіоди LEDA і LEDB в регістрі PHLCON.

3. Налаштовуємо дуплексний режим, якщо хочемо перевизначити значення, визначуване полярністю світлодіода LEDB. Для включення повного дуплексу встановлюємо біти PHCON1.PDPXMD і Macon3.FULDPX.

4. Дозволяємо прийом пакетів.

3.6. Налаштування зв'язку

Стандартні налаштування мережі:

IP: 192.168.1.170.

MAC: 84.85.88.16.0.41.

Порт: 80

Необхідно відкрити Командний рядок і написати ping 192.168.1.170. Повинно прийти 4 відповіді (рис.3.6).

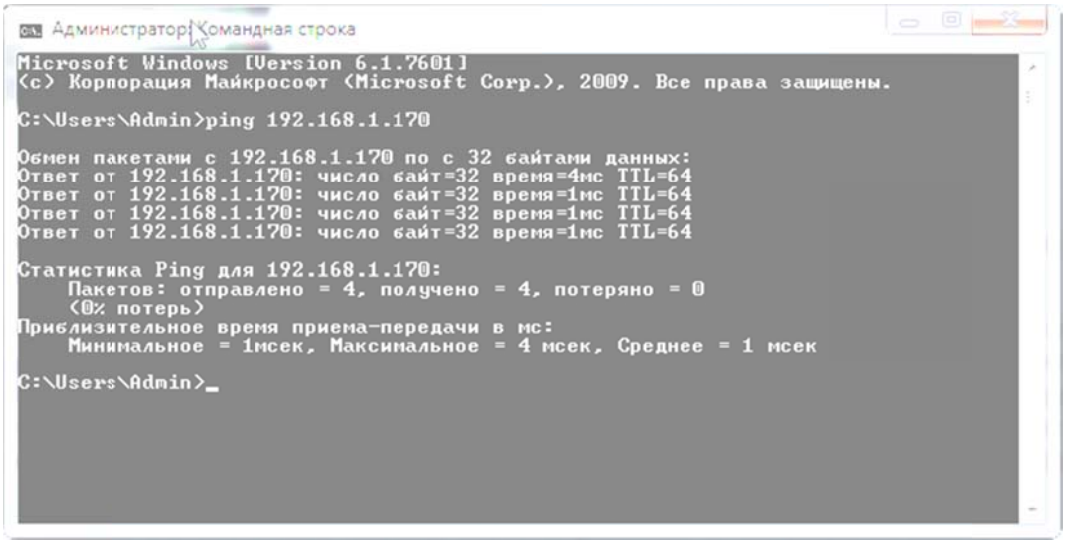

Рисунок 3.6 – Перевірка зв’язку з пристроєм

Далі необхідно відкрити браузер і перейти за адресою http://192.168.1.170/, тоді з’явиться вікно, показане на рис.3.7.

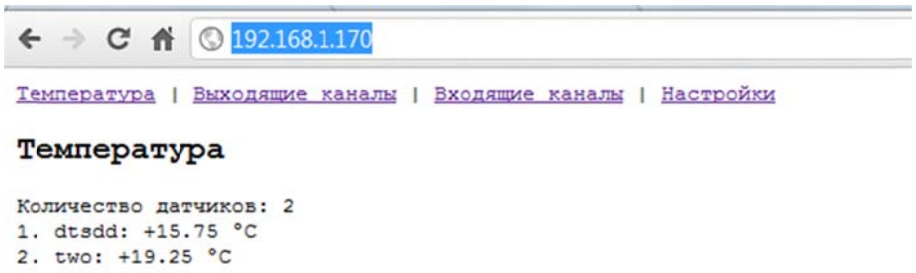

Рис. 3.7. Відображення температури у вікні браузера

На рис.3.8 представлена сторінка налаштувань пристрою.

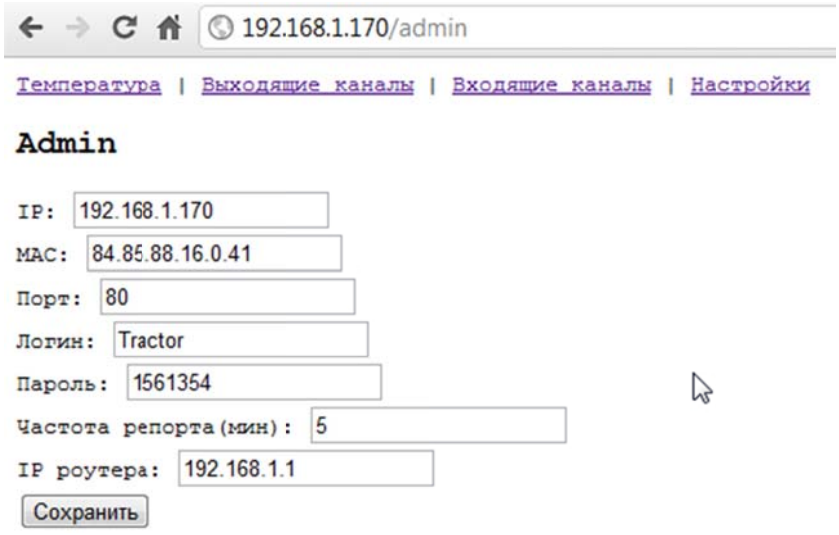

Рисунок 3.8 – Налаштування системи моніторингу в мережі

3.7. Висновок

У розділі виконана розробка алгоритмів та написане програмне забезпечення для мікроконтролера та комп’ютера. Основна складність при написанні програмного забезпечення для мікроконтролера полягає в його відлагодженні. За допомогою приборів неможливо визначити стан мікроконтролера, значення регістрів. При непередбаченій поведінці програми виникає проблема пошуку причини: помилка в програмі або в апаратній частині, яка призвела до зміни характеру вхідних сигналів в мікроконтролер.

За допомогою сполучення системи з комп’ютером вдалося значно підвищити швидкість налагодження програмного забезпечення. З’явилася можливість слідкувати за станом внутрішніх регістрів мікроконтролера та аналізувати поведінку системи.

					172.4397ст3.КР.ПЗ.Р4			
Змн.	*Арк.*	*№ документу*	*Підпис*	*Дата*				
Розробник		Воскобойніков В. І.			ОХОРОНА ПРАЦІ	*Літ.*	*Арк.*	*Аркушів*
Консультант		Тубальцев А. М.					52	
Н. контр.		Добрінова Л. П.				НУК імені адмірала Макарова		
Керівник		Іхсанов Ш. М.						
Зав. каф.		Рябенький В. М.						

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людини в процесі праці.

Закон України «Про охорону праці» від 14 жовтня 1992 року визначає основні положення щодо реалізації конституційного права громадян на охорону їх життя і здоров'я в процесі трудової діяльності, регулює за участю відповідних державних органів відносини між власником підприємства, установи і організації або уповноваженим ним органом і працівником з питань безпеки, гігієни праці та виробничого середовища і встановлює єдиний порядок організації охорони праці в Україні. Дія Закону поширюється на всі підприємства, установи і організації незалежно від форм власності та видів їх діяльності, на усіх громадян, які працюють, а також залучені до праці на цих підприємствах.

Законодавство про охорону праці складається з Закону України «Про охорону праці», Кодексу законів про працю України та інших нормативних актів. У разі, коли міжнародними договорами або угодами, в яких бере участь Україна, встановлено більш високі вимоги до охорони праці, ніж ті, що передбачено законодавством України, то застосовуються правила міжнародного договору або угоди.

Державна політика в галузі охорони праці базується на принципах:

- пріоритету життя і здоров'я працівників відповідно до результатів виробничої діяльності підприємства, повної відповідальності власника за створення безпечних і нешкідливих умов праці;

- комплексного розв'язання завдань охорони праці на основі національних програм з цих питань та з урахуванням інших напрямів економічної і соціальної політики, досягнень в галузі науки і техніки та охорони навколишнього середовища;

- соціального захисту працівників, повного відшкодування шкоди особам, які потерпіли від нещасних випадків на виробництві і професійних захворювань;

- встановлення єдиних нормативів з охорони праці для всіх підприємств, незалежно від форм власності і видів їх діяльності;

- використання економічних методів управління охороною праці, проведення політики пільгового оподаткування, що сприяє створенню безпечних і нешкідливих умов праці, участі держави у фінансуванні заходів щодо охорони праці;

- здійснення навчання населення, професійної підготовки і підвищення кваліфікації працівників з питань охорони праці;

- забезпечення координації діяльності державних органів, установ, організацій та громадських об'єднань, що вирішують різні проблеми охорони здоров'я, гігієни та безпеки праці, а також співробітництва і проведення консультацій між власниками та працівниками (їх представниками), між усіма соціальними групами при прийнятті рішень з охорони праці на місцевому та державному рівнях;

- міжнародного співробітництва в галузі охорони праці, використання світового досвіду організації роботи щодо поліпшення умов і підвищення безпеки праці.

4.1. Аналіз небезпечних і шкідливих факторів, що впливають на людину при виробництві та експлуатації системи збору даних

Небезпечним виробничим фактором називається такий виробничий фактор, вплив якого на працюючого у визначених умовах приводить до травми або до іншого раптового різкого погіршення здоров'я. Результатом впливу цього фактора на людину є нещасні випадки.

Шкідливим виробничим фактором називається такий виробничий фактор, що приводить до захворювання або зниження працездатності. Результатом впливу цього фактора на людину є професійні захворювання.

Проведемо аналіз небезпечних і шкідливих факторів, характерних при розробці і використанні зняття показників навантажень на механічних конструкціях з відображенням інформації на екран монітору за допомогою акселерометрів, за допомогою персонального комп'ютера.

4.1.1. Мікроклімат на робочому місці

Кліматичні умови є важливим фактором нормальної роботи електронної техніки і високої продуктивності праці працівників. З метою створення нормальних умов на робочому місці встановлені норми виробничого мікроклімату. Основні параметри мікроклімату на ділянці по збору електронного обладнання представлені в таблиці 4.1.

Таблиця 4.1. Основні параметри мікроклімату.

Період року	Параметри повітряного середовища								
	Фактичні			Оптимальні			Припустимі		
	t, °C	Відносна вологість, %	Швидкість повітря, м/с	t, °C	Відносна вологість, %	Швидкість повітря, м/с	t, °C	Відносна вологість, %	Швидкість повітря, м/с
Холодний	18	66	0,2	20-22	40-60	0,2	18-22	70	0,3
Теплий	24	60	0,3	20-25	40-60	0,4	28	60-65	0,5

З таблиці 4.1 випливає, що параметри мікроклімату в лабораторії знаходяться в межах норми.

4.1.2. Забруднення повітря на робочому місці

При виробництві системи вимірювання вібрації на механічних конструкціях з відображенням інформації на екран монітору за допомогою акселерометрів можна виділити наступні основні технологічні операції:

- нанесення малюнка друкованих плат;

- травлення друкованих плат у розчині хлорного заліза (FeCl3);

- свердлення отворів;

- монтаж електронних компонентів на друкованих платах;

- покриття друкованих плат вологостійким поліуретановим лаком.

Під час виконання перерахованих технологічних операцій в атмосферу викидається ряд речовин, більшість з яких є небезпечними для здоров'я людини. Найбільш шкідливі з них приведені в таблиці 4.2.

Таблиця 4.2. Шкідливі речовини.

Речовина	Забруднюючі компоненти	Характер дії, області поразки	Клас небезпеки	Гранично допустима концентрація в повітрі		Викид в робочій зоні, мг/м^3
				Разова мг/м^3	Середньодобова, мг/м^3	
Припої олов'яно-свинцеві ПОС-40 ПОС-60	Pb(свинець) Sn(олово)	Центральна нервова система, полові органи	I	-	0,0003	0,0001
	Ni(нікель)	Канцероген	I	0,0002	0,0002	-
	каніфоль	Дихальні шляхи	IV	0,3	0,3	0,2
Флюси	Етиловий спирт	Центральна нервова система	IV	0,1	0,1	-
Лаки	Полімери	Центральна нервова система	IV	0,35	0,35	0,05

Як видно з таблиці 4.2, зміст шкідливих речовин в атмосфері лабораторії не перевищує гранично допустиму концентрацію.

Слід зазначити, що шкідливі речовини викидаються в атмосферу тільки в процесі виробництва системи вимірювання навантажень на механічних конструкціях з відображенням інформації на екран монітору за допомогою акселерометрів, але не в процесі її налагодження й експлуатації.

Незважаючи на те, що середньодобовий викид свинцю й олова в атмосферу менше гранично допустимого, необхідно враховувати здатність цих речовин, як і ряду інших важких металів, накопичуватися в організмі людини і відповідно викликати важкі поразки організму.

4.1.3. Виробниче освітлення

Виробниче освітлення має істотне значення в зниженні виробничого травматизму, зменшує небезпеку багатьох виробничих факторів ризику, знижує небезпеку захворювання органів зору, створює нормальні умови роботи і підвищує загальну працездатність організму.

Існує три виду освітлення: природне, штучне і сполучене. Природне освітлення буває: бічне, котре у свою чергу може бути однобічним і двостороннім; верхнє і комбіноване. Штучне освітлення по конструктивному виконанню буває: загальне, для освітлення приміщень; місцеве, для освітлення одного робочого місця; комбіноване освітлення. Також, штучне освітлення по функціональному призначенню буває: робоче, аварійне, евакуаційне, чергове і охоронне. Сполучене освітлення представляє сукупність природного і штучного освітлення.

Як джерела світла застосовуються лампи накалювання і газорозрядні лампи. Лампи накалювання бувають вакуумні і газонаповнені. До переваг застосування ламп накалювання відносяться: зручність в експлуатації, не потрібно додаткових пристроїв для включення в мережу, простота у виготовленні, широкий діапазон потужностей і робочих напруг, відсутність періоду загорання. До недоліків ламп накалювання відносяться: низький ККД і низька світлова віддача, малий термін служби (2000-2500 годин), у спектрі цих ламп переважають жовті і червоні промені, що сильно відрізняє їхній спектральний склад від природного світла. До переваг газорозрядних ламп відносяться високий ККД, великий термін служби (8000-12000 годин) і можливість імітації денного світла. До недоліків газорозрядних ламп відносяться наявність пульсацій світлового потоку і стробоскопічного ефекту, необхідність наявності додаткового пристрою для включення в мережу, токсичність парів ртуті при руйнуванні лампи. Основним типом газорозрядної лампи є люмінесцентна лампа.

4.1.4. Електронебезпека

При розробці системи вібродіагностування на механічних конструкціях з відображенням інформації на екран монітору за допомогою акселерометрів одним з небезпечних виробничих факторів є ураження людини електричним струмом при дотику до струмоведучих частин.

Вплив електричного струму на людський організм може призвести до травм та навіть гибелі. Це залежить від величини та тривалості проходження струму, від роду та частоти, електричного опору тіла людини, прикладеної напруги.

Вплив електричного струму призводить до електричних травм:

а) місцеві електротравми: електричні опіки, знаки, металізація шкіри під дією електричної дуги, механічне пошкодження, електрофтальмія;

б) електричні удари:

- 1 ступень: скорочення м’язів без втрати тями;

- 2 ступень: скорочення м’язів з втратою тями;

- 3 ступень: втрата тями та порушення роботи серця;

- 4 ступень: клінічна смерть.

Відомо, що постійний струм у середньому в 4-5 раз безпечніше змінного струму f=50 Гц, але при напрузі до 500 В. Якщо напруга вище 500 В, то ураження постійним струмом небезпечніше ніж змінним.

Визначають чотири інтервали при змінному струмі f=50 Гц:

1) I=0,6-1,5 мА – пороговий відчутний струм;

2) I=10-15 мА – пороговий невідпускаємий струм, тому що виникає удома м’язів рук та голосових зв’язок;

3) I=25-50 мА – дія струму розповсюджується на м’язи грудної клітини, тобто можливо припинення дихання людини, що потрапила під напругу, якщо струм діє декілька хвилин;

4) I=500-100 мА – фібриляційний струм, який впливає на сердечні м’язи; при тривалості декілька секунд можлива зупинка серця.

Причинами ураження електричним струмом можуть стати: випадковий дотик або наближення на небезпечну відстань до струмоведучих частин; поява напруги на механічних частинах устаткування у результаті пошкодження ізоляції; поява напруги на відключених струмоведучих частинах устаткування внаслідок помилкового ввімкнення обладнання; виникнення крокової напруги на поверхні землі у результаті замикання проводу на землю.

4.1.5. Небезпека при роботі з ПЕОМ і пристроєм його сполучення з датчиками

Зняття параметрів вібрації на механічних конструкціях з відображенням інформації на екран монітору за допомогою акселерометрів відбувається за допомогою персональної електронної обчислювальної машини (ПЕОМ). Небезпечними й шкідливими факторами при роботі на ПЕОМ і пристроєм його сполучення з датчиками є такі:

- ураження людини електричним струмом при дотику до незахищених (доступних) елементів;
- електромагнітне поле;
- електростатичне поле;
- вплив світлового потоку;
- відбитого світла.

Електромагнітне поле (ЕМП) створюється магнітними котушками системи, що відхиляє, що знаходяться біля цокольної частини електронно-променевої трубки монітора. Також слід відзначити вплив електромагнітного поля при використанні пристрою сполучення ПЕОМ з датчиками.

ЕМП має здатність біологічного, специфічного і теплового впливу на організм людини.

Біологічний вплив ЕМП залежить від довжини хвилі, інтенсивності, тривалості і режимів впливу, розміром і анатомічною будівлею органа, що піддається впливові ЕМП. ЕМП міліметрового діапазону поглинаються поверхневими шарами шкіри, сантиметрового – шкірою і прилягаючими до неї

тканинами, дециметрового – проникають на глибину 8-10 см. Для більш тривалих хвиль тканини тіла людини є добре провідним середовищем.

Механізм порушень, що відбуваються в організмі під впливом ЕМП, обумовлений їх специфічним (нетепловим) і тепловим впливом.

Специфічний вплив ЕМП обумовлений біохімічними змінами, що відбуваються в клітках і тканинах. Найбільш чуттєвими є центральна і серцево-судинна системи. Спостерігаються порушення умовно-рефлекторної діяльності, зниження біоелектричної активності мозку, зміна міжнейронних зв'язків. Можливі відхилення з боку ендокринної системи.

У початковому періоді впливу може підвищуватися збудливість нервової системи, у наступному відбувається зменшення її функції, що виявляється в астенічних станах, тобто фізичної і нервово-психічної слабості. У зв'язку з цим для загальної клінічної картини хронічного впливу ЕМП характерні: головний біль, стомлюваність, погіршення самопочуття, гіпотонія, брадикардія, зміна провідності серцевого м'яза. Зазначені явища можуть бути слабко, помірковано або явно виражені. Можливі незначні і, як правило, нестійкі зміни в крові.

Тепловий вплив ЕМП характеризується підвищенням температури тіла, локальним виборчим нагріванням тканин, органів, кліток унаслідок переходу ЕМП у теплову енергію. Інтенсивність нагрівання залежить від кількості поглиненої енергії і швидкості відтоку тепла від ділянок тіла, що опромінюються. Відтік тепла утруднений в органах і тканинах з поганим кровопостачанням. До них у першу чергу відноситься кристалик ока. Під дією опромінення в ньому можуть відбуватися коагуляція білків або дифузійні зміни з наступним розвитком катаракти. Піддані тепловому опроміненню ЕМП також паренхіматозні органи (печінка, підшлункова залоза) і полі органи, що містять рідину (сечовий міхур, шлунок). Нагрівання їх може викликати загострення хронічних захворювань (виразок, кровотеч, перфорацій.)

Електростатичне поле виникає в результаті опромінення екрана потоком заряджених часток.

Неприємності, викликані їм, зв'язані з пилом, що накопичується в електростатично заряджених екранах, що летить на оператора під час роботи за монітором. Електростатичний потенціал, що виникає в тілі оператора при його роботі за монітором, різний і коливається в межах +0,6кв/м (однак він може бути і негативним). Потенціал оператора служить вирішальним фактором при осадженні часток пилу на поверхні тіла, що, у свою чергу, може служити причиною шкірних захворювань, псування контактних лінз, при катаракті розвивається помутніння мембрани кристалика (ока). Низьковольтний електромагнітний розряд здатний змінювати і переривати клітинний розвиток.

Біологічний вплив на оператора світлового потоку і відбитого світла. Світловий потік падає на екран монітора і відбиває від нього.

Одночасно екран монітора випромінює ЕМП у видимій частині спектра. Оскільки вектори лінійної поляризації світлового потоку, відбитого від екрана, і видимої області спектра збігаються, у результаті інтерференції цих двох світлових потоків відбувається посилення амплітуди результуючого світлового потоку, що виявляється в зниженні контрастності і появі відблисків. Подібні впливи викликають утома очного м'яза і згодом можуть стати причиною короткозорості.

4.2. Розрахунок загального освітлення

Основна задача освітлення у виробничих приміщеннях полягає в забезпеченні оптимальних умов для бачення. Ця задача вирішується вибором найбільш раціональної системи освітлення та джерел світла.

Розрахуємо штучне освітлення загального значення для лабораторії з розробки електронної апаратури.

Довжини А = 5 м, ширина В = 4, висота Н = 2,7 м. Висота робочої поверхні hp = 1м. Побілено стелю, стіни – світла шпалера, світла розрахункова поверхня. Напруга мережі U_c = 220 В.

Приймемо коефіцієнт мінімальної освітленості Emin = 300 лкс (робота високої точності). Вибираємо світильник типу УСП з двома люмінесцентними лампами ЛБ40, вмонтований у стелю.

Відстань від стелі до робочої поверхні :

$$H_0 = H - h_p = 2,7\text{-}1 = 1,7 \text{ м.}$$

Висота підвісу світильника над освітлюваною поверхнею h_p і відстань від стелі до робочої поверхні H_0 рівні, тобто $h_p = H_0 = 1,7$ м

Для досягнення найбільшої освітленості при розміщенні світильників приймаємо значення: $L/h_p = 1,5$

Тоді відстань між центрами світильників

$L = 1,5 \cdot h = 1,5 \cdot 1,7 = 2,55$ м

Визначимо індекс приміщення

$$i = \frac{A \cdot B}{h(A+B)} = \frac{5 \cdot 4}{1.7 \cdot 10} = 2$$

При i = 1,9; $\rho_п$ = 70 %; $\rho_з$ = 50 %; $\rho_р$ = 30 % для світильника типу УСП коефіцієнт використання світлового потоку η = 0,48. Світловий потік лампи ЛБ40 дорівнює $Ф_л$ = 3040 лм. $K_з$ – коефіцієнт запасу, що для приміщення, лабораторії дорівнює 1,5 для люмінесцентних ламп. Коефіцієнт нерівномірності для люмінесцентних ламп Z приймаємо 1,1.

Визначимо кількість світильників для забезпечення необхідної освітленості:

$$N = \frac{E_{\min} \cdot S \cdot K_з \cdot Z}{n \cdot Ф_л \cdot \eta} = \frac{300 \cdot 20 \cdot 1.5 \cdot 1.1}{2 \cdot 3040 \cdot 0.48} = 3.4,$$

де E_{min} – рівень мінімальної освітленості за нормами, n – кількість ламп у світильнику, S – площа лабораторії.

Приймаємо 4 світильники.

Загальна потужність освітлювальної установки

$P = P_л \cdot N \cdot n = 40 \cdot 2 \cdot 4 = 320$ Вт

4.3. Заходи по техніці безпеки

Мікроклімат на робочому місці. Для забезпечення нормальної роботи у відповідності установленими виробничими нормами, необхідно обігрівати і прохолоджувати виробниче приміщення у відповідний час року за допомогою кондиціонерів повітря, вентиляторів, центральної системи опалення або нагрівальних приладів.

Забруднення повітря на робочому місті. У процесі проведення виробничих робіт варто організовувати вентиляцію всього приміщення і витяжки на робочих місцях. У разі потреби використовувати спеціалізовані повітроочисні пристрої.

Виробниче освітлення. Для ліквідації стробоскопічного ефекту варто розділити існуючі світильники з газорозрядними лампами на три рівні групи і включити їх у різні фази трифазної мережі. У цьому випадку за рахунок фазового зрушення струмів у ланцюзі світильників різних груп буде відбуватися їхня взаємокомпенсація і сумарний світловий потік буде практично незмінним у часі.

В приміщенні при експлуатації ПЕОМ загальне висвітлення варто виконувати у виді суцільних або переривчастих ліній світильників, розташованих збоку від робочих місць, паралельно лінії зору користувача при рядному розташуванні ПЕОМ. При периметральном розташуванні комп'ютерів лінії світильників повинні розташовуватися локалізовано над робочим столом, ближче до його переднього краю, зверненому до оператора.

Необхідно обмежити прямий відблиск від джерел освітлення, при цьому яскравість світлих поверхонь, що знаходяться в поле зору, повинна бути не більш 200 кд/м^2.

Також необхідно обмежувати відбитий відблиск на робочих поверхнях за рахунок правильного вибору типів світильників і розташування робочих місць стосовно джерел природного і штучного висвітлення, при цьому яскравість відблисків на екрані не повинна перевищувати 40 кд/м^2 і яскравість стелі при застосуванні системи відбитого висвітлення не повинна перевищувати 200 кд/м^2.

Електробезпека. Електробезпечність забезпечується відповідною конструкцією електроустаткування, застосуванням технічних способів і засобів захисту, організаційними і технічними заходами. Конструкція електроустаткування повинна відповідати умовам його експлуатації, забезпечувати захист персоналу від зіткнення зі струмоведучими частинами й устаткування – від влучення усередину сторонніх предметів і води. Найбільш розповсюдженими технічними засобами захисту є захисне заземлення і занулення. Прокладка контуру заземлення виконувалася під час побудови лабораторії у відповідності з будівельними вимогами, тому робити додатковий розрахунок опору контуру немає потреби. Організаційні і технічні заходи щодо забезпечення електробезпечності полягають в основному у відповідному навчанні, інструктажі і допуску до роботи осіб, що пройшли медичний огляд, виконанні ряду технічних мір при проведенні робіт з електроустаткуванням, дотриманні особливих вимог при роботах з частинами, що находяться під напругою.

Електромагнітне поле. Одним з найбільш ефективних і часто застосовуваних методів захисту від низькочастотних і радіовипромінювань є екранування. Для екранів використовуються, головним чином, матеріали з великою електричною провідністю. Як засоби індивідуального захисту застосовуються спецодяг, виготовлена з металізованої тканини у виді комбінезонів, халатів, фартухів, курток з каптурами і вмонтованими в них захисними окулярами.

Робота з ПЕОМ. Вимоги до приміщень:

- приміщення з ПЕОМ повинно відноситься до категорії В;

- не допускається розміщення приміщення для роботи з ПЕОМ у підвалах і цокольних поверхах;

- площа на одне робоче місце, обладнане ПЕОМ повинно бути не менш 6м^2;

- приміщення з ПЕОМ повинні бути оснащені системою автоматичної пожежної сигналізації й вуглекислотними вогнегасниками по два на 20 м^2 усій площі приміщення.

Санітарно-гігієнічні вимоги.

- рівень освітленості на робочому столі повинний бути в межах 300-500 лк;

- температура повітря в приміщенні в холодний час року повинна знаходитись в межах від 21ºС до 24ºС, а в теплий час року – від 23ºС до 25ºС.

У приміщенні, де одночасно експлуатується або обслуговується більш 5 ПЕОМ, на видному і доступному місці встановлюється аварійний резервний вимикач, що може цілком відключити електроживлення приміщення, крім освітлення.

Користувач ПЕОМ повинен знаходитись на відстані 500-600 мм від монітора при діагоналі екрана 15» і на відстані 600-700 мм при діагоналі 17».

Згідно з наказом про охорону праці режим праці і відпочинку операторів, що працюють з ЕОМ, повинний організовуватися в залежності від виду категорії трудової діяльності.

Види трудової діяльності розділяються на три групи (А, Б, В):

А: робота зі зчитування інформації з екрана ЕОМ з попереднім запитом;

Б: робота з введення інформації;

В: творча робота в режимі діалогу з ЕОМ.

Перерви для відпочинку повинні надаватися в залежності від ступеня стомлення, своєчасність їх важливіше тривалості. Під час перерв необхідно виконувати спеціалізовані комплекси гімнастичних вправ.

4.4. Висновки

У розділі розглянуті питання, пов’язані з аналізом шкідливих та небезпечних виробничих факторів, впливові яких піддаються оператори, котрі задіяні при супроводженні розробленої системи збору даних про стан газодизельного агрегату.

Визначені шляхи вирішення цих проблем для покращення та створення безпечності умов праці обслуговуючого персоналу.

Виконаний розрахунок загального освітлення виробничого приміщення.

ВИСНОВКИ

Використання сучасних технологій на основі мікропроцесорних систем відкриває нові можливості в підвищенні якості контролю температури, дозволяє прогнозувати та запобігати некоректним ситуаціям на основі статистичної бази, яка утворюється в процесі роботи системи.

Проведене тестування електронної системи моніторингу температури показало дієздатність апаратної частини пристрою, експериментальна перевірка працездатності виявила високі експлуатаційні показники в цільовому застосуванні пристрою, досить широкі можливості до вдосконалення, налагодження системи і подальшого впровадження її в сферу конкурентоспроможних систем моніторингу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Practical design techniques for sensor signal conditioning. Part 8, 9. Analog devices –Autex Ltd. http://www.analog.com

2. Precision Voltage Regulator Controller ADP3310 – Datasheet – Analog Device https://www.analog.com/media/en/technical-documentation/data-sheets/adp3310.pdf

3. Design and realization of high-precision digital control direct current power supply / H. Jing et al. 2011 International Conference on Consumer Electronics, Communications and Networks (CECNet), Xianning, China, 16–18 April 2011. 2011. URL: https://doi.org/10.1109/cecnet.2011.5768894 (date of access: 09.05.2025).

4. Design of high precision and high power bidirectional adjustable power supply / H. Huang et al. Fusion Engineering and Design. 2021. Vol. 162. P. 112103. URL: https://doi.org/10.1016/j.fusengdes.2020.112103 (date of access: 09.05.2025).

5. Hegarty T. An overview of conducted EMI specifications for power supplies. Analog | Embedded processing | Semiconductor company | TI.com. URL: https://www.ti.com/lit/wp/slyy136/slyy136.pdf (date of access: 09.05.2025).

6. Yang W., Huang J., Dai Z. Design of a High-precision DC regulated power supply system. 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 12–14 June 2020. 2020. URL: https://doi.org/10.1109/itnec48623.2020.9084831 (date of access: 09.05.2025).

7. DC power supply with parallel active compensation function and tuneable inductive filter based on analogue control: Part 1. Bulletin of the Polish Academy of Sciences Technical Sciences. 2023. URL: https://doi.org/10.24425/bpasts.2024.148839 (date of access: 12.05.2025).

8. Onishi Hiroyuki, Nagaoka Shingo, Zaitsu Toshiyuki. Noise Cancellation for Miniaturization of Switched Mode Power Supply. OMRON TECHNICS. URL:

https://www.omron.com/global/en/assets/file/technology/omrontechnics/vol52/OMT_Vol52_003EN.pdf (date of access: 12.05.2025).

9. Orlando Murray. How to Reduce EMI and Shrink Power-supply Size with an Integrated Active EMI Filter. Texas Instruments. URL: https://www.ti.com/lit/ta/sszt179/sszt179.pdf (date of access: 12.05.2025).

10. Understanding DC Filters: Types, Applications & Benefits. Any-pcba.com. URL: https://www.anypcba.com/blogs/electronic-component-knowledge/understanding-dc-filters-types-applications-benefits.html (date of access: 12.05.2025).

11. Applications Engineer Texas Instruments. “The engineer’s guide to EMI in DC-DC converters (part 1): standards requirements and measurement techniques,” How2Power Today, December 2017 issue. Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265 Copyright © 2018, Texas Instruments Incorporated

12. P-Channel Logic Level Enhancement Mode Field Effect Transistor NDP6020P – Datasheet – Onsemi https://www.onsemi.com/pdf/datasheet/ndp6020p-d.pdf

13. Precision Voltage Regulator Controller ADP3310 – Datasheet – Analog Device https://www.analog.com/media/en/technical-documentation/datasheets/adp3310.pdf

14. Louis Vlemincq. Smart Ripple Canceller Offers Near-Zero Dropout https://www.radiolocman.com/shem/schematics.html?di=655183.

15. Шаруда В.Г. Практикум з теорії автогматичного управляння. Дніпропетровьк, НГУ, 2002, 414с.

ДОДАТОК А

Текст програми для мікроконтролеру

```
typedef struct tcp_state {

    // Код состояния
    tcp_status_code_t status;

    // Время последнего получения пакета
    //    (для таймаутов)
    uint32_t event_time;

    // Указатель потока
    //    (номер следующего отправляемого пакета)
    uint32_t seq_num;

    // Указатель подтверждения
    //    (докуда мы получили поток от удалённого узла)
    uint32_t ack_num;

    // Адрес и порт удалённого узла и наш порт
    uint32_t remote_addr;
    uint16_t remote_port;
    uint16_t local_port;
} tcp_state_t;

// Режим отправки пакета
//    используется в основном для оптимизации отправки
typedef enum tcp_sending_mode {
    // Отправляем новый пакет.
    // При этом резолвим MAC-адрес получателя,
    //    выставляем все поля пакета,
    //    залиыаем данные,
    //    считаем чексумму.
    // Самый медленный способ отправки.
    TCP_SENDING_SEND,

    // Только что получили пакет от удалённого узла и
    //    отправляем ему пакет в ответ.
    // При этом обмениваем в нём адреса получателя/отправителя,
    //    обновляем указатели seq/ack,
    //    заливаем новые данные,
    //    пересчитываем чексумму.
    //    Затем тупо пинаем пакет туда, откуда он пришёл
    TCP_SENDING_REPLY,

    // Только что отправляли пакет узлу и
    //    отправляем следом ещё один.
    // При этом заливаем новые данные,
    //    обновляем указатели seq/ack,
```

```
    //    пересчитываем чексумму,
    //    все адреса оставляем те же.
    // Самый быстрый способ отправки.
    // После отправки любого пакета, этот режим
    //    активируется автоматически.
    TCP_SENDING_RESEND
} tcp_sending_mode_t;

tcp_sending_mode_t tcp_send_mode;

// Признак отправки подтверждения
//    устанавливается при отправке пакета с установленым флагом ACK.
uint8_t tcp_ack_sent;

// Отправка пакета
//    Поле tcp.flags должно быть установлено вручную
uint8_t tcp_xmit(tcp_state_t *st, eth_frame_t *frame, uint16_t len)
{
    uint8_t status = 1;
    uint16_t temp, plen = len;

    ip_packet_t *ip = (void*)(frame->data);
    tcp_packet_t *tcp = (void*)(ip->data);

    // Отправляем новый пакет?
    if(tcp_send_mode == TCP_SENDING_SEND)
    {
        // Устанавливаем поля в заголовке
        ip->to_addr = st->remote_addr;
        ip->from_addr = ip_addr;
        ip->protocol = IP_PROTOCOL_TCP;

        tcp->to_port = st->remote_port;
        tcp->from_port = st->local_port;
    }

    // Отправляем пакет в ответ?
    if(tcp_send_mode == TCP_SENDING_REPLY)
    {
        // Меняем местами порты отправителя/получателя
        temp = tcp->from_port;
        tcp->from_port = tcp->to_port;
        tcp->to_port = temp;
    }

    if(tcp_send_mode != TCP_SENDING_RESEND)
    {
        // Устанавливаем поля в заголовке
        tcp->window = htons(TCP_WINDOW_SIZE);
        tcp->urgent_ptr = 0;
    }
```

```
// Если отправляем SYN, добавляем опцию MSS
//       (максимальный размер сегмента)
if(tcp->flags & TCP_FLAG_SYN)
{
   tcp->data_offset = (sizeof(tcp_packet_t) + 4) << 2;
   tcp->data[0] = 2;//MSS
   tcp->data[1] = 4;//длина
   tcp->data[2] = TCP_SYN_MSS>>8;
   tcp->data[3] = TCP_SYN_MSS&0xff;
   plen = 4;
}
else
{
   tcp->data_offset = sizeof(tcp_packet_t) << 2;
}

// Устанавливаем указатели потока
tcp->seq_num = htonl(st->seq_num);
tcp->ack_num = htonl(st->ack_num);

// Вычисляем контрольную сумму
plen += sizeof(tcp_packet_t);
tcp->cksum = 0;
tcp->cksum = ip_cksum(plen + IP_PROTOCOL_TCP,
   (uint8_t*)tcp - 8, plen + 8);

// Отправляем всю эту фигню в виде IP-пакета
switch(tcp_send_mode)
{
case TCP_SENDING_SEND:
   status = ip_send(frame, plen);
   tcp_send_mode = TCP_SENDING_RESEND;
   break;
case TCP_SENDING_REPLY:
   ip_reply(frame, plen);
   tcp_send_mode = TCP_SENDING_RESEND;
   break;
case TCP_SENDING_RESEND:
   ip_resend(frame, plen);
   break;
}

// Обновляем наш указатель потока
st->seq_num += len;
if( (tcp->flags & TCP_FLAG_SYN) || (tcp->flags & TCP_FLAG_FIN) )
   st->seq_num++;

// Устанавливаем флаг отправки подтверждения
if( (tcp->flags & TCP_FLAG_ACK) && (status) )
   tcp_ack_sent = 1;
```

```
    return status;
}// Отправить блок с установленным флагом PSH
#define TCP_OPTION_PUSH           0x01
// Отправить блок и закрыть соединение
#define TCP_OPTION_CLOSE       0x02

// Отправляет данные по TCP
void tcp_send(uint8_t id, eth_frame_t *frame, uint16_t len, uint8_t options)
{
    ip_packet_t *ip = (void*)(frame->data);
    tcp_packet_t *tcp = (void*)(ip->data);
    tcp_state_t *st = tcp_pool + id;
    uint8_t flags = TCP_FLAG_ACK;

    // Если соединение не устновлено, выходим – защита от дурака)
    if(st->status != TCP_ESTABLISHED)
        return;

    // Если нужно, добавляем флаг PUSH
    if(options & TCP_OPTION_PUSH)
        flags |= TCP_FLAG_PSH;

    // Если закрываем соединения, добавляем флаг FIN
    //    и начинаем ждать, пока удалённый
    //          узел тоже пришлёт FIN/ACK
    if(options & TCP_OPTION_CLOSE)
    {
        flags |= TCP_FLAG_FIN;
        st->status = TCP_FIN_WAIT;
    }

    // Пинаем пакет куда нужно
    tcp->flags = flags;
    tcp_xmit(st, frame, len);
}// Обработчик TCP-пакетов
void tcp_filter(eth_frame_t *frame, uint16_t len)
{
    ip_packet_t *ip = (void*)(frame->data);
    tcp_packet_t *tcp = (void*)(ip->data);
    tcp_state_t *st = 0, *pst;
    uint8_t id, tcpflags;

    // Пришло не на наш адрес (например, на широковещательный) – выбрасываем
    // Кто будет посылать TCP-пакеты на широковещательный адрес?
    // ХЗ, подстрахуемся на всякий случай))
    if(ip->to_addr != ip_addr)
        return;

    // Отнимаем размер заголовка и получаем количество данных в пакете
    len -= tcp_head_size(tcp);
```

```
// Отрезаем для удобства флаги SYN, FIN, ACK и RST
//    другие флаги нам не интересны
tcpflags = tcp->flags & (TCP_FLAG_SYN|TCP_FLAG_ACK|
   TCP_FLAG_RST|TCP_FLAG_FIN);

// Переключаем tcp_xmit в режим отправки пакетов «в ответ»
tcp_send_mode = TCP_SENDING_REPLY;
tcp_ack_sent = 0;

// Ищем к какому из активных соединений относится
//    полученный пакет
for(id = 0; id < TCP_MAX_CONNECTIONS; ++id)
{
   pst = tcp_pool + id;

   if( (pst->status != TCP_CLOSED) &&
      (ip->from_addr == pst->remote_addr) &&
      (tcp->from_port == pst->remote_port) &&
      (tcp->to_port == pst->local_port) )
   {
      st = pst;
      break;
   }
}

// Не нашли такого соединения - пакет является запросом
//    на соединение или просто каким-то заблудившимся пакетом
if(!st)
{
   // Получили запрос на соединение?
   if( tcpflags == TCP_FLAG_SYN )
   {
      // Ищем свободный слот, куда добавить новое соединение
      for(id = 0; id < TCP_MAX_CONNECTIONS; ++id)
      {
         pst = tcp_pool + id;

         if(pst->status == TCP_CLOSED)
         {
            st = pst;
            break;
         }
      }

      // Нашли слот, и приложение подтверждает это соединение
      if(st && tcp_listen(id, frame))
      {
         // Добавляем новое соединение
         st->status = TCP_SYN_RECEIVED;
         st->event_time = gettc();
```

```
            st->seq_num = gettc() + (gettc() << 16);
            st->ack_num = ntohl(tcp->seq_num) + 1;
            st->remote_addr = ip->from_addr;
            st->remote_port = tcp->from_port;
            st->local_port = tcp->to_port;

            // И пинаем обратно SYN/ACK
            tcp->flags = TCP_FLAG_SYN|TCP_FLAG_ACK;
            tcp_xmit(st, frame, 0);
        }
    }
}

// Полученный пакет относится к известному соединению
else
{
    // Получили RST (сброс), значит прибиваем соединение
    if(tcpflags & TCP_FLAG_RST)
    {
        if( (st->status == TCP_ESTABLISHED) ||
            (st->status == TCP_FIN_WAIT) )
        {
            tcp_closed(id, 1);
        }
        st->status = TCP_CLOSED;
        return;
    }

    // Проверяем в пакете указатели потока
    //    и подтверждения.
    // Пакет должен находиться в потоке строго
    //    после предыдущего полученного,
    // и подтверждать последний отправленный
    //    нами пакет
    if( (ntohl(tcp->seq_num) != st->ack_num) ||
        (ntohl(tcp->ack_num) != st->seq_num) ||
        (!(tcpflags & TCP_FLAG_ACK)) )
    {
        return;
    }

    // Обновляем наш указатель подтверждения,
    //    чтобы следующим отправленным пакетом
    //    подтвердить только что полученный пакет
    st->ack_num += len;
    if( (tcpflags & TCP_FLAG_FIN) || (tcpflags & TCP_FLAG_SYN) )
        st->ack_num++;

    // Сбрасываем счётчик таймаута
    st->event_time = gettc();
```

```
// Смотрим, в каком состоянии мы сейчас, и в зависимости
//    от этого реагируем на полученный пакет
switch(st->status)
{

// Состояние:
//    Мы отправили SYN
//    Теперь ждём SYN/ACK
case TCP_SYN_SENT:

   // Получили что-то, но не SYN/ACK
   if(tcpflags != (TCP_FLAG_SYN|TCP_FLAG_ACK))
   {
      // Прибиваем соединение
      st->status = TCP_CLOSED;
      break;
   }

   // Отправляем ACK
   tcp->flags = TCP_FLAG_ACK;
   tcp_xmit(st, frame, 0);

   // Соединение установлено
   st->status = TCP_ESTABLISHED;

   // Дёргаем коллбэк, в котором
   //    прога может посылать свои данные
   tcp_read(id, frame);

   break;

// Состояние:
//    Мы получили SYN
//    И отправили SYN/ACK
//    Теперь ждём ACK
case TCP_SYN_RECEIVED:

   // Если получили не ACK,
   if(tcpflags != TCP_FLAG_ACK)
   {
      // то прибиваем соединение
      st->status = TCP_CLOSED;
      break;
   }

   // Соединение установлено
   st->status = TCP_ESTABLISHED;

   // Приложение может отправлять данные
   tcp_read(id, frame, 0);
```

```
      break;

    // Состояние:
    //    Соединение установлено
    //    Ждём ACK или FIN/ACK
    case TCP_ESTABLISHED:

      // Получили FIN/ACK
      if(tcpflags == (TCP_FLAG_FIN|TCP_FLAG_ACK))
      {
        // Отдаём данные приложению
        if(len) tcp_write(id, frame, len);

        // Отправляем FIN/ACK
        tcp->flags = TCP_FLAG_FIN|TCP_FLAG_ACK;
        tcp_xmit(st, frame, 0);

        // Соединение закрыто
        st->status = TCP_CLOSED;
        tcp_closed(id, 0);
      }

      // Получили просто ACK
      else if(tcpflags == TCP_FLAG_ACK)
      {
        // Отдаём данные приложению
        if(len) tcp_write(id, frame, len);

        // Приложение может отправлять данные
        tcp_read(id, frame);

        // Если приложение не отправило ничего,
        //    но нам приходили данные,
        //    отправляем ACK
        if( (len) && (!tcp_ack_sent) )
        {
          tcp->flags = TCP_FLAG_ACK;
          tcp_xmit(st, frame, 0);
        }
      }

      break;

    // Состояние:
    //    Мы отправили FIN/ACK
    //    Ждём FIN/ACK или ACK
    case TCP_FIN_WAIT:

      // Получили FIN/ACK
      if(tcpflags == (TCP_FLAG_FIN|TCP_FLAG_ACK))
      {
```

```
            // Отдаём приложению данные
            if(len) tcp_write(id, frame, len);

            // Отправляем ACK
            tcp->flags = TCP_FLAG_ACK;
            tcp_xmit(st, frame, 0);

            // Соединение закрыто
            st->status = TCP_CLOSED;
            tcp_closed(id, 0);
         }

         // Получили данные и ACK
         // (типа удалённый узел отдаёт данные из буфера)
         else if( (tcpflags == TCP_FLAG_ACK) && (len) )
         {
            // Отдаём данные приложению
            tcp_write(id, frame, len);

            // Отправляем ACK
            tcp->flags = TCP_FLAG_ACK;
            tcp_xmit(st, frame, 0);
         }

         break;

      default:
         break;
      }
   }
}
// Пул соединений
typedef struct httpd_state {
   const prog_char *data;
   uint16_t numbytes;
} httpd_state_t;

httpd_state_t httpd_pool[TCP_MAX_CONNECTIONS];

// HTML-страничка вместе с HTTP-заголовком
const prog_char index_page[] =
   «HTTP/1.0 200 OK\r\n»
   «Content-Type: text/html; charset=windows-1251\r\n»
   «Connection: close\r\n»
   «\r\n»
   «<pre>Восьмибитный превед!\r\n\r\n</pre>\r\n»
   «<img src='i.gif'>«;

// Страничка 404
const prog_char error_page[] =
   «HTTP/1.0 404 Not Found\r\n»
```

```
    «Content-Type: text/html; charset=windows-1251\r\n»
    «Connection: close\r\n»
    «\r\n»
    «<h1>404 – Not Found</h1>«;

// Принимаем подключения на порт 80
uint8_t tcp_listen(uint8_t id, eth_frame_t *frame)
{
    ip_packet_t *ip = (void*)(frame->data);
    tcp_packet_t *tcp = (void*)(ip->data);
    return (tcp->to_port == htons(80));
}

// Принимаем данные по TCP
void tcp_write(uint8_t id, eth_frame_t *frame, uint16_t len)
{
    httpd_state_t *st = httpd_pool + id;
    ip_packet_t *ip = (void*)(frame->data);
    tcp_packet_t *tcp = (void*)(ip->data);
    char *request = (void*)tcp_get_data(tcp);
    char *url, *p;

    // Получен HTTP-запрос?
    if(!st->data)
    {
        // Выковыриваем из запроса URL
        url = request + 4;
        if( (!memcmp_P(request, PSTR(«GET «), 4)) &&
            (p = strchr(url, ' ')) )
        {
            *p = 0;

            // Запросили главную страничку?
            if(!strcmp_P(url, PSTR(«/»)))
            {
                st->data = index_page;
                st->numbytes = sizeof(index_page) - 1;
            }

            // Запросили картинку?
            else if(!strcmp_P(url, PSTR(«/i.gif»)))
            {
                st->data = imagedata;
                st->numbytes = sizeof(imagedata);
            }

            // Запросили что-то другое?
            else
            {
                st->data = error_page;
                st->numbytes = sizeof(error_page) - 1;
```

```
            }
        }
    }
}

// Отправка данных
void tcp_read(uint8_t id, eth_frame_t *frame)
{
    httpd_state_t *st = httpd_pool + id;
    ip_packet_t *ip = (void*)(frame->data);
    tcp_packet_t *tcp = (void*)(ip->data);
    char *buf = (void*)(tcp->data);
    uint8_t i, options;
    uint16_t blocksize;

    // Идёт отправка данных?
    if(st->numbytes)
    {
        // Отправляем пучок пакетов, до 4 штучек за раз
        for(i = 4; i; --i)
        {
            // До 512 байт в 1 пакете
            blocksize = 512;

            options = 0;

            // Если данных осталось <= 512 байт,
            //    отправляем сколько осталось
            //    и закрываем соединение
            if(st->numbytes <= blocksize)
            {
                options = TCP_OPTION_CLOSE;
                blocksize = st->numbytes;
            }

            // В посседнем пакете пучка
            //    выставляем флаг PSH
            else if(i == 1)
            {
                options = TCP_OPTION_PUSH;
            }

            // Заполняем пакет данными
            memcpy_P(buf, st->data, blocksize);
            st->data += blocksize;
            st->numbytes -= blocksize;

            // И отправляем
            tcp_send(id, frame, blocksize, options);

            // Если данных больше нету, выходим из цикла
```

```
            if(options == TCP_OPTION_CLOSE)
                break;
        }
    }
}

// Соединение закрыто/оборвалось
void tcp_closed(uint8_t id, uint8_t hard)
{
    httpd_state_t *st = httpd_pool + id;
    st->data = 0;
    st->numbytes = 0;
}
```