

8.2 Реалізація мережевих вимірювальних модулів на базі малорозрядних мікроконтролерів

Особливості організації інформаційно-вимірювальних систем на основі стандарту LXI.

Використання в вимірювальних пристроях відкритих комунікаційних стандартів спрощує завдання інтеграції в інформаційно-вимірювальні системи, забезпечує сумісність з обладнанням відомих світових виробників. Розробка таких пристроїв на основі 8-розрядних мікроконтролерів дозволяє оптимізувати їх вартість та енергоефективність, прискорити та спростити розробку [196].

LXI (LAN eXtensions for Instruments) – галузевий комунікаційний стандарт для контрольно-вимірювальних пристроїв. Він відкритий і безкоштовний. Стандарт був розроблений консорціумом провідних виробників і користувачів контрольно-вимірювального обладнання, створеним у вересні 2004 року і нараховує вже більше 40 членів. У Консорціум LXI входять провідні компанії галузі контрольно-вимірювального обладнання, включаючи засновників – Agilent Technologies і інших виробників як Tektronix, Rohde&Schwarz, Keithley Instruments, Racal Instruments і Yokogawa.

Специфікація LXI [197] заснована на добре відпрацьованому мережевому стандарті Ethernet і описує взаємодію будь-яких LXI приладів, незалежно від виробника, що вирішує проблеми, пов'язані із сумісністю пристроїв, дозволяє швидко і ефективно проводити збір даних та їх обробку.

Стандарт LXI став логічним продовженням розробки серії стандартів VXI і PXI для вимірювальних систем. У 2004 році стандарт був опублікований і в тому ж році був створений Консорціум LXI – об'єднання виробників, розробників систем, системних інтеграторів і користувачів для розвитку стандарту і вдосконалення обладнання LXI. При використанні стандарту LXI збільшується швидкодія мережевих вимірювальних систем, зменшується їх вартість, геометричні розміри, а також знижуються трудовитрати на розробку і налаштування систем [198, 199].

Така системна архітектура заснована на використанні попередніх досягнень комп'ютерних технологій, насамперед переваг технологій передачі Ethernet, мережевих можливостей інтернет, протоколів локальних мереж (LAN) та ін. Високі швидкості вводу/виводу, відсутність крейтів та інтерфейсних кабелів, сучасне програмне забезпечення разом з добре зарекомендованими стандартами готових засобів вимірювальної техніки визначає ефективність стандарту LXI.

Системи LXI підтримують практично всі стандартні інтерфейси – як власне LXI/LAN, так і VXI, PXI, RS232, а також GPIB. Така різноманітність підтримуваних пристроїв забезпечує універсальність систем LXI, швидкий і простий перехід на стандарт LXI для вже існуючих систем, а також повну сумісність LXI систем з іншими.

Вимірювальні системи LXI можуть мати ті розміри, які вимагаються для конкретного приміщення або виконання необхідних завдань. На відміну від VXI і PXI, які жорстко обмежені розмірами крейтів, стандарт LXI містить лише рекомендації до дотримання специфікацій IEC.

Інтерфейс LXI дозволяє з'єднувати настільні пристрої з фронтальною панеллю, стійкові модулі без панелі, окремі вимірювальні пристрої, в тому числі вбудовані, настільні та настінні. Стандарт LXI відрізняється від PXI і VXI тим, що кожен модуль або пристрій системи має власний блок живлення, охолодження, систему запуску, захист від перешкод і підключення до мережі Ethernet. Якщо використовується змінна напруга, вони використовують загальноприйняті стандарти живлення (100 В-240 В і 47-66 Гц), а при постійній напрузі вони працюють при 48 В. Також допускається живлення через Ethernet. Охолодження повинно здійснюватися з боків або з передньої/задньої частин, щоб було можливо встановлювати один модуль над іншим без зниження продуктивності. Кожен модуль повинен відповідати вимогам з електромагнітних завад, існуючим в країні, де він застосовується. Пристрій LXI повинен мати порт LAN і відповідати правилам IEEE 802.3 стандарту Ethernet. Стандарт рекомендує певне положення кабелю живлення (на задній панелі праворуч) і вимикача

живлення (на задній панелі знизу праворуч), конектора Ethernet (на задній панелі в дальньому кінці праворуч), роз'ємів підключення до сигналів (на передній панелі) і індикаторів LAN/живлення /IEEE 1588 (LXI часова синхронізація) зліва на передній панелі. Для приладів, що мають приладову панель, індикацію LAN / живлення / IEEE 1588 можна відображати на дисплеї, а прилади, що мають клавіатуру, можуть здійснювати функцію перезавантаження LAN безпосередньо з неї. Таким чином, окремі модулі обладнання LXI можуть бути використані незалежно від системи.

LXI-прилади мають світлодіодну (LED) індикацію живлення, індикацію LAN та індикацію IEEE 1588. За специфікацією LXI стандарту індикатор живлення повинен розміщуватися на передній панелі приладу. Деякі прилади можуть працювати в режимі очікування (STANDBY). З точки зору безпеки, рекомендується цей режим відображати на індикаторі живлення. Для пристроїв, які використовують режим STANDBY, індикатор живлення повинен бути трьох становий двохколірний (помаранчевий/зелений). Індикатор стану мережевого підключення повинен бути двокольоровим (червоний/зелений). Світлодіоди забезпечують дві функції: індикація несправності і ідентифікація пристрою. Індикатор статусу годинника IEEE 1588 призначений для відображення як статусу так і типу годинника. Індикатор може спалахувати з двома різними швидкостями.

Запуск і синхронізація LXI – це одна з головних особливостей LXI, яка дозволяє поєднати переваги телекомунікацій на базі Ethernet, стандарт синхронізації в часі IEEE 1588 [200] і можливість використання апаратної шини запуску VXI. Можливості запуску та синхронізації є найбільш цікавими аспектами шини LXI. Використовуючи можливості LAN і протокол часової синхронізації IEEE 1588, стандарт LXI визначає режими запуску, який відсутній в GPIB, PXI або VXI. Три класи пристроїв LXI – C, B і A – реалізують ці можливості в зростаючій мірі.

Клас C є базовим класом і він включає всі вимоги, що стосуються інтерфейсу LAN і протоколів, функції LAN discovery, інтерфейсу драйвера IVI і

веб-сторінок приладу, також рекомендації, що відносяться до живлення, охолодження, розмірів, індикаторів, клавіші Reset (скидання). Усі вимірювальні прилади LXI повинні дотримуватися вимог класу C.

Клас B включає всі вимоги класу C і додає протокол часової синхронізації стандарту IEEE 1588. Це робить можливим досягнення субмікросекундної синхронізації пристроїв LXI, розташованих в будь-яких точках локальної мережі або віддалених місцях. Клас B додає також протоколи повідомлень однорангової мережі та багатоадресної передачі повідомлень (вимагається в класах B і A, дозволяється в класі C). Режими передачі "точка-точка" і "запуск групи" (бродкастинг) у старті подій або процесів LAN застосовуються в стандарті LXI спеціально для запуску пристроїв та приладів, що надає системам LXI гнучкість, недоступну колишнім архітектурам. У стандарті IEEE 1588 пристрої LXI класу B синхронізують свої тактові генератори для досягнення єдиної установки часу вимірів або вихідних сигналів без використання спеціальних кабелів синхронізації і можливості ставити відмітку часу на всі події і дані. Стандарт IEEE 1588 в поєднанні з запуском по LAN дозволяє передавати інформацію і дані про час без використання комп'ютера, що працює в режимі реального часу. Потенціал цих можливостей ще тільки досліджується.

Обладнання класу A задовольняє всі вимоги класів C і B, до яких додана шина апаратного запуску. Восьмиканальна апаратна шина M-LVDS (Multipoint-Low Voltage Differential Signaling) – багатоточкова диференціальна шина сигналів низької напруги може з'єднувати пристрої, що знаходяться на близькій відстані один від одного за схемою послідовного ланцюга або зірки, а також комбінуючи ці дві топології. Шина запуску забезпечує проходження сигналу між приладами з вкрай незначною затримкою – порядку 3 нс / м.

Використання стандарту IEEE 1588 надає можливість забезпечити загальносистемне точне відстеження часу, що створюється за допомогою синхронізованого годинника реального часу в кожному пристрої LXI. Такі відстеження може бути використано для різних функцій, в тому числі:

- позначення даних мітками часу для подальшого аналізу;

- генерація LXI подій для точного запуску і синхронізації з іншими LXI приладами;
- генерація журналу подій, що дозволяє впорядкувати події в усіх частинах системи LXI;
- створення синхронних сигналів в декількох приладах LXI.

Кожний прилад LXI, що реалізує IEEE 1588, повинен забезпечувати функціональність, що повністю відповідає стандарту IEEE 1588 і профілю LXI 1588. Стандарт LXI версії 1.3 і пізніша специфікація вимагають від приладів LXI підтримки IEEE 1588-2008 і більш нових версій цього стандарту. Стандарт LXI версії 1.2 і старіші версії вимагають підтримки стандарту IEEE 1588-2002.

IEEE 1588 повинен реалізовуватись з достатньою для задач системи точністю. За специфікацією LXI прилади можуть забезпечити синхронізацію часу з точністю до 40 нс. Реалізація на базі програмного забезпечення має меншу точність. Функціональність IEEE 1588 можливо реалізувати використовуючи тільки програмне забезпечення. Однак, апаратна реалізація є кращою і є необхідною для забезпечення точності в 40 нс. Часова точність системи LXI приладів може бути обмежена й іншими чинниками, наприклад, якістю LAN-мостів в системі. В цьому випадку рекомендується використання граничного або прозорого годинника стандарту IEEE 1588. Якщо це не доступно, то необхідно використання LAN-комутаторів, які мають низьку затримку для пакетів UDP.

В усіх приладах LXI, що реалізують IEEE 1588, повинні бути доступні імпульси з періодом в одну секунду. Механічна та електрична специфікація цього виходу визначається замовником, але вихід повинен генерувати передній фронт синхронно з секундним переходом годинника IEEE 1588. Цей імпульс з періодом в одну секунду призначений для порівняння з відповідним імпульсом інших годинників в системі для перевірки синхронізації роботи. Для синхронізації годинників LXI приладів за стандартом IEEE 1588 потрібно реалізувати протокол Precision Time Protocol (PTP) [201]. Апаратна підтримка протоколу в різних пристроях може бути реалізована по-різному. Мінімум, що

вимагається для реалізації РТР – вміння апаратної частини проставляти в мережевому пакеті значення часу, що відповідає моменту отримання повідомлення до пристрою. Проставлений час буде використано для обчислення помилки.

Помилки можуть з'явитися з багатьох причин. Генератори частоти в пристроях різні і дуже мала ймовірність того, що два різних пристрої будуть працювати ідеально такт в такт. Сюди ж можна додати мінливі умови навколишнього середовища, що впливають на згенеровану частоту. Тому програмна реалізація синхронізації часто не дозволяє домогтися бажаної точності. Час, що минув з моменту отримання повідомлення (точніше отримання сигналу на прийом повідомлення у пристрої) до переходу на точку входу в переривання або на виклик callback-функції, не може бути строго визначеним. Апаратні рішення з підтримкою РТР здатні проставляти ці таймштампи самостійно (наприклад, чіпи від Micrel).

Крім таймштампів до апаратної підтримки також можна віднести можливість налаштовувати кварцовий генератор (щоб вирівняти частоту з майстром), або можливість підстроювання годинника (кожен такт збільшувати значення годинника на X нс). РТР є клієнт-серверним протоколом синхронізації, тобто для реалізації протоколу потрібно як мінімум 2 пристрої. Отже, нехай Master-пристрій – еталонний годинник, а Slave-пристрій – годинник, який необхідно змусити працювати точно. Announce message (повідомлення анонсу) містить інформацію, що відправляється майстром всім Slave-пристроєм. Slave пристрій за допомогою цього повідомлення може вибрати кращого майстра (для цього розроблено алгоритм BMC – Best Master Clock). Цей алгоритм наведено в стандарті [200].

Повідомлення Sync/Follow Up, DelayResp, PDelayResp/PDelayFollowUp відправляються майстром. DelayReq, PDelayReq – запити Slave пристроїв. Master надає практично всю інформацію сам. Відправка здійснюється на Multicast-адреси (при бажанні можна використовувати Unicast режим), визначені у стандарті. Для PDelay повідомлень є окрема адреса (01-80-C2-00-00-0E для

Ethernet і 224.0.0.107 для UDP). Решта повідомлення відсилаються на 01-1B-19-00-00-00 або 224.0.1.129. Пакети відрізняються полями ClockIdentity (ідентифікатор годинника) і SequenceId (ідентифікатор пакету).

Цикл синхронізації починається з того, що Master відправляє повідомлення Sync і одночасно записує час відправки t_1 . Існує одно-і двоетапні режими роботи. Якщо присутні повідомлення FollowUp – то ми маємо справу з двоетапною реалізацією, пунктирною стрілкою показані необов'язкові повідомлення. FollowUp повідомлення відправляється за Sync і містить час t_1 . Якщо здійснюється передача в один етап, то Sync містить t_1 в тілі повідомлення. У будь-якому випадку t_1 буде отримано пристроєм.

У момент отримання повідомлення Sync на Slave генерується таймштамп t_2 . Таким чином визначається t_1 , t_2 . Slave генерує повідомлення DelayReq одночасно з генерацією t_3 . Master отримує DelayReq повідомлення, одночасно генеруючи t_4 . Значення t_4 відправляється Slave-пристроєм в DelayResp повідомленні. На рис.1 показана процедура обміну між головним і синхронізованим пристроєм.

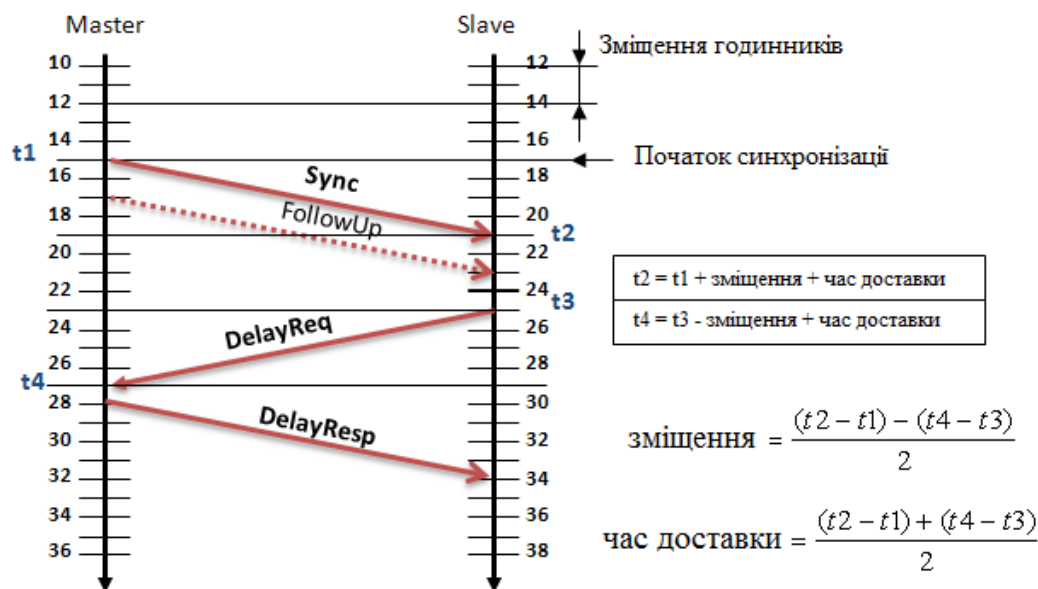


Рисунок 1. Процедура синхронізації пристроїв

Досягти нульової розбіжності в часі можна, якщо кварц генерує ідеально однакові частоти для синхронізованих пристроїв. На практиці ж тактова частота для годинників різна, тобто на одному пристрої за 1 секунду значення часу

збільшитися на 1 секунду, а на іншому, наприклад, на 1,000001 секунду. Звідси з'являється розбіжність часу. У стандарті [200] описано приклад обчислення відношення часу, що минув для Master і на Slave за певний інтервал. Це відношення буде коефіцієнтом для частоти Slave-пристрою. При цьому надана вказівка, що підстроювання може здійснюватися декількома способами. Насамперед можливо:

- змінити тактову частоту Slave-пристрою;
- не змінювати тактову частоту, але за кожен такт тривалістю T значення годин буде збільшуватися не на T , а на $T + \Delta t$.

Для реалізації обох способів потрібно обчислити різницю в значеннях часу на Master-пристрої за певний інтервал, а також різницю в часі, за цей же інтервал на Slave-пристрої. Визначення коефіцієнту за першим способом зображено на рис.2.

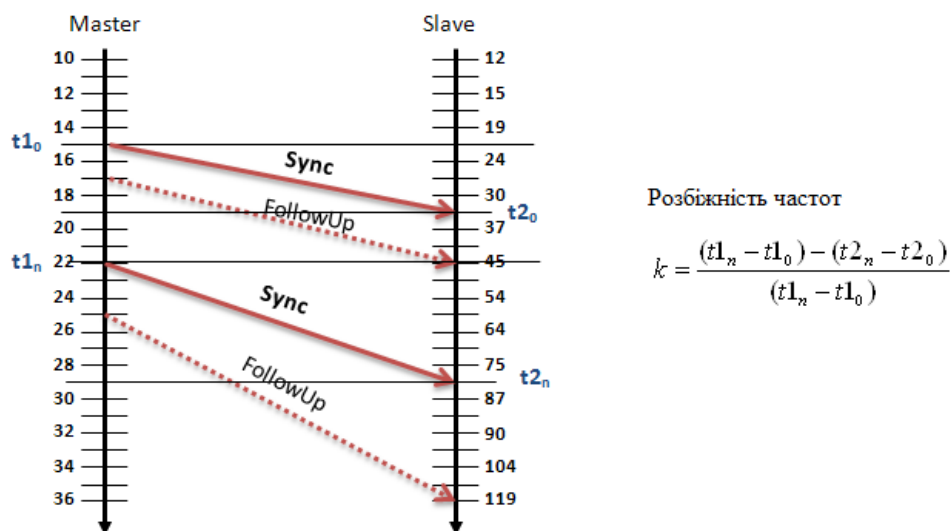


Рисунок 2. Розбіжність частот тактових генераторів приладів LXI

Для реалізації другого способу потрібно обчислення Δt . Δt – величина, яка буде впливати із значення часу для кожного певного інтервалу. Наприклад, на діаграмах рис.2 в той час як на майстрі минуло $22 - 15 = 7$ секунд, на Slave-пристрої минуло $75 + (87-75) / 2 - (30 + (37-30) / 2) = 47.5$ секунд.

В стандарті IEEE 1588 наведено кілька способів обчислення часу доставки. Існують 2 режиму роботи для PTPv2. Це режим E2E (End-to-End), та режим P2P (Peer-to-Peer). При реалізації стандарту можна використовувати будь-який з

режимів, але їх не можна поєднувати в одній мережі. У режимі E2E час доставки обчислюється за повідомленнями, які пройшли через декілька пристроїв, кожен з яких пропоставляє в полі корекції повідомлення Sync або FollowUp (якщо передача двоетапна) час, на який пакет затримався на цьому пристрої. Якщо пристрої підключені безпосередньо, час корекції не пропоставляється. Використовуються повідомлення Sync / FollowUp, DelayReq / DelayResp. У режимі P2P в полі корекції заноситься не тільки час, на який затримався пакет, а до нього додається значення $(t_2 - t_1)$. Використовуються повідомлення Sync/FollowUp, PDelayReq/PDelayResp/PDelayRespFollowUp.

Відповідно до протоколу PTP, годинник, крізь який PTP-повідомлення проходять із зміною поля корекції, називаються Transparent Clock (TC). На рис.3 наведено схему передачі повідомлення в режимі E2E.

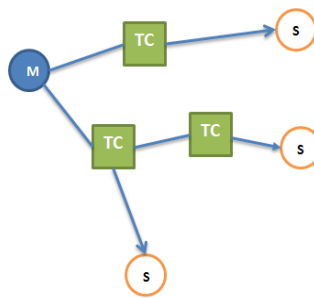


Рисунок 3. Повідомлення в режимі End-to-End

Товстими стрілками вказані повідомлення Sync і FollowUp. На рис.4 зображено схему передачі повідомлення в режимі P2P.

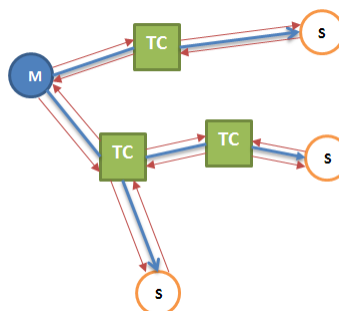


Рисунок 4. Режим Peer-to-Peer

Тонкими лініями показані повідомлення PDelayReq, PDelayResp і PDelayFollowUp. Сеанс обміну цими повідомленнями наведено на рис.5.

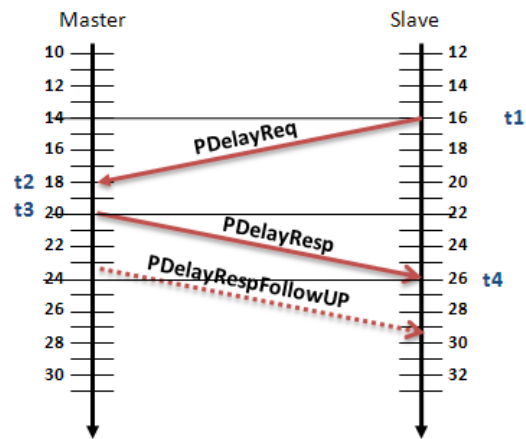


Рисунок 5. Обмін повідомленнями в режимі P2P

Використання стандарту IEEE 1588 є важливою перевагою розподіленої технології середовища LXI, так як дозволяє досягти точності синхронізації приладів в часі, раніше властиву виключно крейтовим системам з центральною шиною.

Розробка апаратного забезпечення мікроконтролерного LXI-сумісного інтерфейсного блоку на базі модуля CP2201EB.

Апаратне забезпечення розробленого мікроконтролерного LXI-сумісного інтерфейсного блоку реалізовано на основі мережевого модуля CP2201EB компанії Silicon Laboratories. Зовнішній вигляд модуля наведено на рис.6.

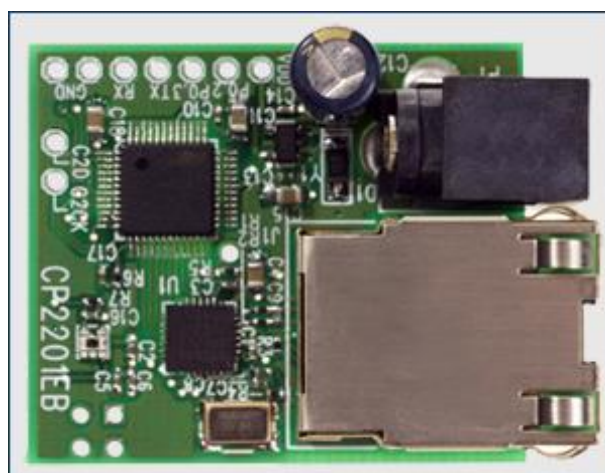


Рисунок 6. Зовнішній вигляд мережевого модуля CP2201EB

Модуль CP2201EB реалізовано на базі сучасного швидкісного 8-розрядного мікроконтролера C8051F340 (Silicon Laboratories) [202], та

мережевого Ethernet-контролера Silicon Laboratories CP2201 для з'єднання мікроконтролера з мережею Ethernet [203]. Модуль має в своєму складі наступні компоненти: роз'єм RJ45 (гальванічно розв'язаний); мікроконтролер C8051F340; мережевий драйвер CP2201; датчик освітлення EL7900ICL.

Технічні характеристики мікроконтролера C8051F340. Аналогові модулі. 10-розрядний АЦП (до 200 тис. перетворень в секунду); вбудований аналоговий мультиплексор, що працює в одноканальному і диференційному режимі; живлення від зовнішнього та внутрішнього джерела; вбудований датчик температури; можливість зовнішнього запуску; два компаратори; внутрішня опорна напруга. Вбудований USB-контролер. USB специфікація 2.0; підтримка повної швидкості або низької швидкості; вбудоване відновлення синхронізації, для підтримки повної або низької швидкості не потрібно зовнішніх генераторів; буферна пам'ять USB; вбудований приймач не потребує зовнішніх резисторів. Вбудоване налагодження контролера. Забезпечуються точки зупинки, покроковий запуск команд, контроль/модифікація пам'яті і регістрів; висока продуктивність систем відладки з використанням емуляції.

Високошвидкісний процесор серії MCS8051: конвеєрна архітектура виконання програми; 70% інструкцій виконується в 1 або 2 системних такти; швидкодія процесора – 48 MIPS; розширений обробник переривань.

Пам'ять: 4352 байт пам'яті RAM; 64К байт пам'яті Flash, яку можна перепрограмувати під час роботи блоками по 512 байт.

Цифрові модулі: 40 портів вводу-виводу; апаратні інтерфейси SPI, SMBus, два послідовних порти UART; чотири 16-розрядних таймера/лічильника загального призначення; 16-бітний програмований масив лічильників з п'ятьма модулями порівняння; інтерфейс підключення зовнішньої пам'яті (EMIF). Підтримка інтерфейсів SPI, SMBus(I2C), UART(RS-232/485) надає можливість побудови на основі модуля CP2201EB гетерогенних(різномірних) мережевих систем на основі сполучення мережі Ethernet та промислових мереж.

Джерела опорної частоти: внутрішній генератор, що має похибку 0,25% і можливість відновлення тактової частоти, а також підтримує всі режими USB і

UART; зовнішній генератор: кварцовий, RC, або C; внутрішній генератор низької частоти (80 кГц).

Модуль CP2201EB містить в собі датчик освітленості EL7900 фірми Intersil. EL7900 – це оптичний датчик, що об'єднує фотодіод і підсилювач струму на одній монолітній мікросхемі. Вихідний струм прямо пропорційний інтенсивності світла, що діє на фотодіод. Його чутливість вище, ніж у фототранзистора. Особливості мікросхеми EL7900: монолітна мікросхема; містить фотодіод і підсилювач; діапазон від 1 до 8000 люкс; перетворення світла в струм; напруга живлення від 2,7 В до 5,5 В, малий струм живлення- 1 мкА; малий час відгуку – 200мкс; вихідна характеристика близька до лінійної; компактний і легкий корпус для поверхневого монтажу. Принципову схему модуля CP2201EB наведено на рис.7.

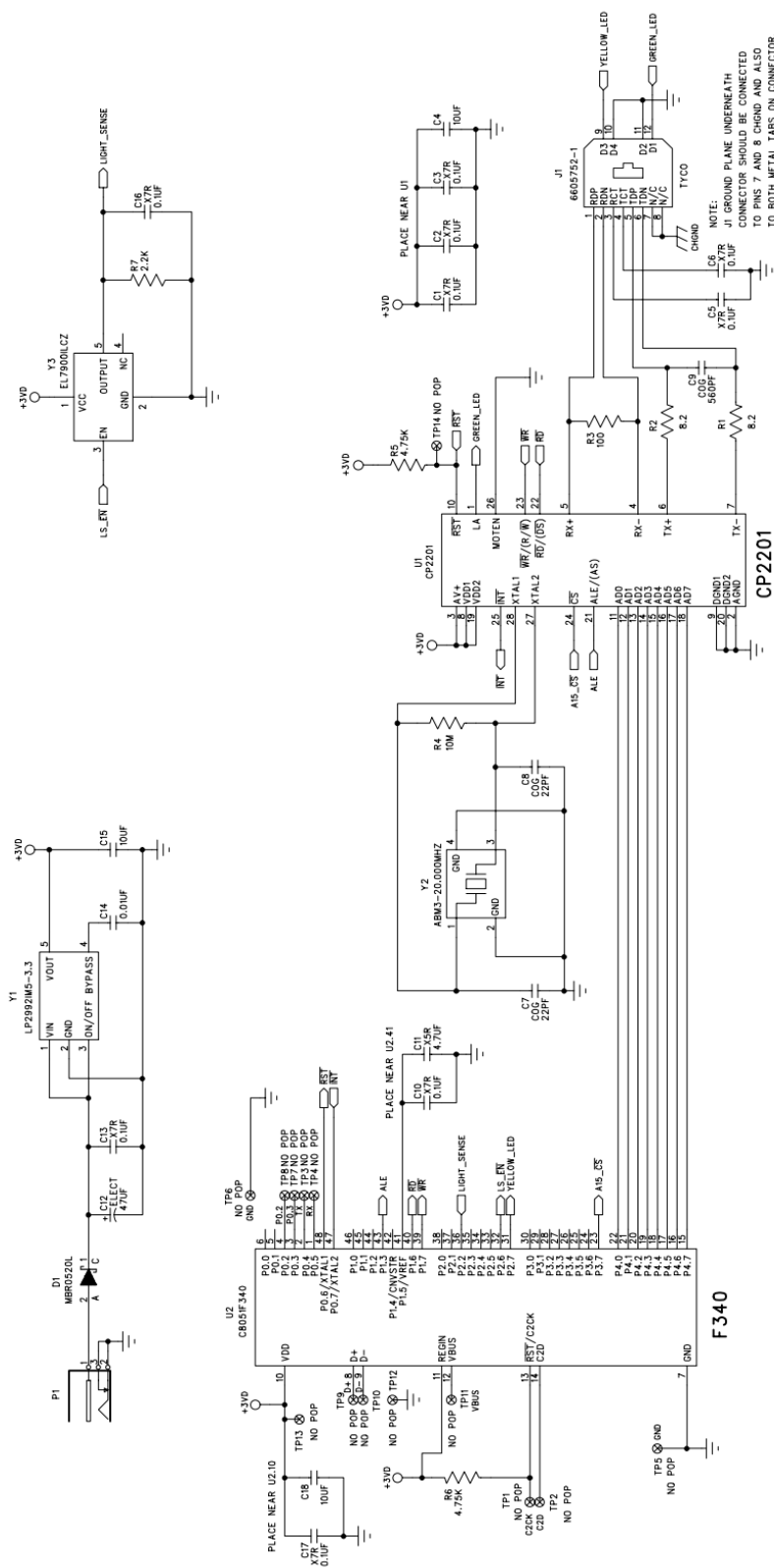


Рисунок 7. Принципова схема мережевого модуля CP2201EV

Блок-схему мікроконтролера C8051F340 наведено на рис.8.

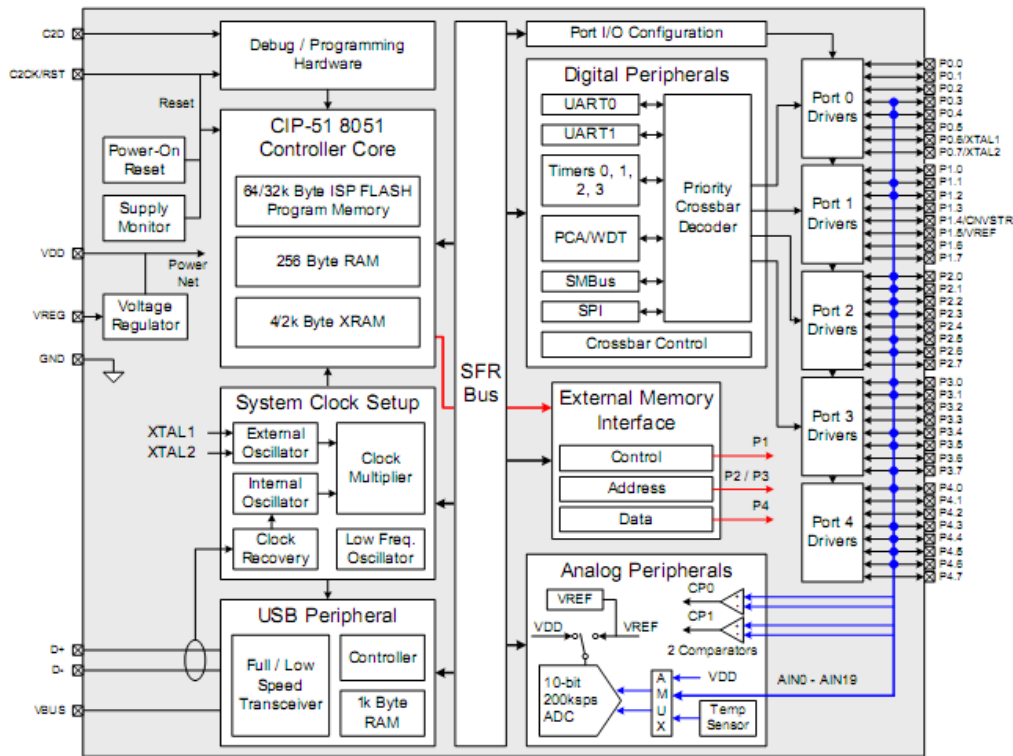


Рисунок 8. Блок-схема мікроконтролера C8051F340

Ethernet контролер CP2201 підтримує мережі 100Мбіт і 1Гбіт, що відповідає вимогам стандарту LXI, має невеликий розмір і високу швидкість роботи. CP2201 – це Ethernet контролер в мініатюрному корпусі з низьким енергоспоживанням і розширеною функціональністю. Містить сумісні з специфікацією IEEE 802.3 блоки MAC і 10BASE-T PHY, має 8K байт Flash-пам'яті. Мікросхема CP2201 дозволяє забезпечити Ethernet доступ для мікроконтролерів і хост-процесорів, задіявши 11 ліній вводу-виводу. 8-бітний паралельний інтерфейс працює як в мультиплексному, так і в звичайному режимі. Вбудована Flash-пам'ять може використовуватися для розміщення даних користувача, контенту веб-сервера або як звичайна енергонезалежна пам'ять; також в Flash-пам'яті зберігається унікальна 48-бітна MAC-адреса, запрограмована виробником.

CP220x являють собою малогабаритні однокристальні IEEE802.3 10-BaseT Ethernet-контролери, призначені для інтеграції інтерфейсу Ethernet в різні системи. Дані контролери реалізують функції MAC і PHY- рівнів стеку мережних протоколів і забезпечують доступ до Ethernet мереж будь-яким

мікроконтролерам чи хост-машинам, що мають одинадцять і більше вільних ліній вводу/виводу. CP2200 випускається в 48-ми вивідному корпусі TQFP розміром 9х9 мм. [203]. Структурну схему CP2201 наведено на рис.9.

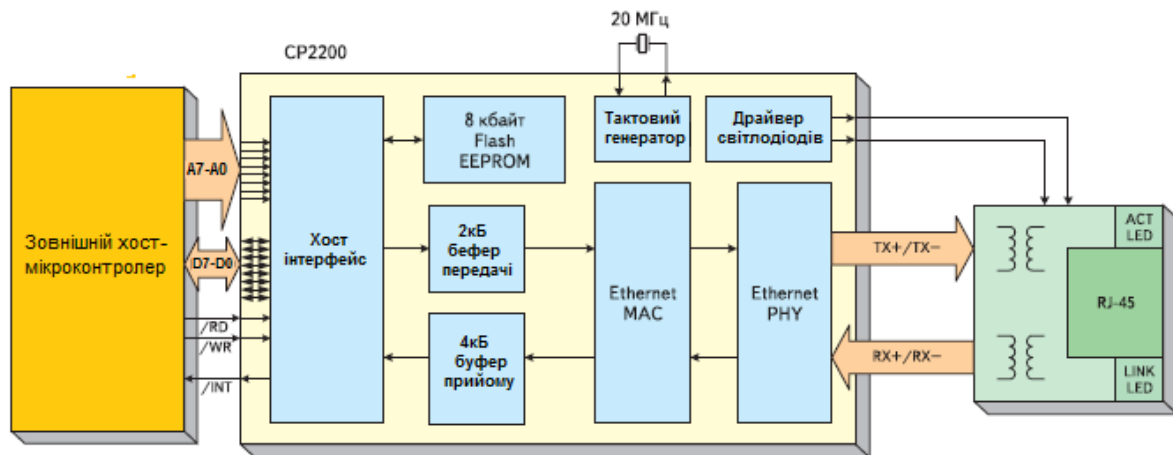


Рисунок 9. Структурна схема Ethernet-контролера CP2201

CP220x містить паралельний хост-інтерфейс, 8К байт енергонезалежної флеш пам'яті, 2-кілобайтовий буфер передачі, 4-кілобайтовий буфер прийому (FIFO), тактовий генератор 20МГц, Ethernet контролери MAC і PHY рівнів, а також драйвер світлодіодів.

Модуль паралельного хост-інтерфейсу дозволяє передавати дані зі швидкістю до 30 Мбіт/с, працює з шинами формату Intel (Motorola) і підтримує 8-розрядні режими взаємодії з зовнішньою хост-системою. Модуль забезпечує доступ до регістрів спеціального призначення, за допомогою яких здійснюються прийом і передача даних, а також контроль стану всіх модулів CP220x. Крім того, хост-інтерфейс здатний генерувати сигнал переривання, який може «пробуджувати» хост-процесор з режиму «сну» для отримання пакетів або при підключенні CP220x до мережі (Wake-on-LAN).

Передача даних по каналу Ethernet здійснюється за допомогою буфера передачі об'ємом 2К байт. CP220x забезпечує простий інтерфейс передачі TCP(UDP) пакету, що вимагає від хост-системи лише завантаження у буфер передачі MAC-адреси відправника та одержувача пакету, довжини і типу пакета, а також самих даних (46-1500 байт).

Інтерфейс прийому Ethernet-даних, крім буфера прийому FIFO об'ємом 4К

байт, містить також буфер швидкого перетворення адреси (TBL), приймальний фільтр і хеш-таблицю. Буфер TBL складається з 8 записів, кожен з яких містить початкову адресу (в буфері прийому), довжину та іншу інформацію (тип, ознаки помилки контрольної суми, неповного пакету та інше) одного отриманого пакету, тобто в буфері прийому може знаходитися до 8 пакетів даних одночасно. Приймальний фільтр і хеш-таблиця дозволяють налаштувати CP220x таким чином, щоб у процесі прийому небажані пакети відкидалися автоматично. Приймальний фільтр відкидає пакети за їх типом, проте підтримуються не всі типи пакетів (тільки ширококомвні, групові пакети, пакети з помилкою контрольної суми і неповні – довжиною менше 64 байт). Тому в CP220x реалізована можливість довільного доступу до даних у буфер прийому, що дозволяє хост-процесору прочитати певні байти прийнятого пакету і вирішити, чи копіювати весь пакет даних у пам'ять хост-системи. За допомогою хеш-таблиці можна встановити діапазон MAC-адрес, які будуть дозволені для запису в буфер прийому; пакети з адресами, не записаними у встановлений діапазон будуть відкидатися.

Серед інших функцій Ethernet-інтерфейсу, реалізованих вбудованими контролерами RHY і MAC: розширення коротких пакетів до мінімальної довжини; обчислення і перевірка контрольної суми (CRC); перевірка довжини пакету; повна сумісність зі стандартами 100/1000 BaseT; підтримка стандартів Plug-and-Play (auto-negotiations, auto-polarity); дуплексний і напівдуплексний режими роботи; виявлення колізій з автоматичним повтором передачі; тестування та налагодження в режимі зворотної петлі (можливі два режими: з замиканням каналу передачі на RHY-рівні або на MAC-рівні); перевірка зв'язку з мережею; виявлення і фільтрація шумів при прийомі; відсікання некоректних пакетів при передачі.

Буфери прийому та передачі Ethernet-інтерфейсу відображені в одному і тому ж діапазоні адресного простору і звернення до них здійснюється за допомогою однієї і тієї ж адреси, однак реєстри даних (для операцій читання-запису) використовуються різні. Крім буферів прийому та передачі в CP220x є ще один

блок пам'яті – вбудована Flash-пам'ять об'ємом 8К байт. Вона розміщена в окремому адресному просторі і доступ до неї здійснюється за допомогою власного показника адреси і регістра даних. Ця пам'ять може використовуватись для зберігання констант користувача, контенту веб-сервера, а також в якості незалежної пам'яті загального призначення. При виготовленні в останні шість комірок цієї пам'яті записується унікальна 48-розрядна MAC-адреса. Стирання Flash-пам'яті виконується секторами по 512 байт, запис реалізується побайтно. Управління операціями запису-стирання Flash-пам'яті здійснюється автоматично апаратними засобами і не перешкоджає роботі хост-інтерфейсу.

Для тактування CP220x може використовуватися або вбудований тактовий генератор (у цьому випадку необхідно підключити зовнішній кварцовий резонатор 20 МГц) або зовнішній сигнал тактування. При зникненні з яких-небудь причин тактових імпульсів буде згенерований сигнал скидання тривалістю близько 1мс, у тому числі на зовнішньому виводі RST, який можна використовувати для скидання зовнішніх модулів. Сигнал скидання генерується також при зникненні напруги живлення або зниженні її порогової величини. В CP220x реалізовано чотири режими електроживлення: нормальний режим живлення – в цьому режимі всі модулі ввімкнені і CP2200 повністю зберігає свою функціональність; режим очікування підключення до мережі – в цьому режимі передавач відключений; режим підтримки пам'яті – в цьому режимі можливий доступ до Flash-пам'яті, а також до буфера прийому та передачі, приймач і передавач Ethernet-інтерфейсу відключені; режим простоювання – в цьому режимі всі модулі CP2200 відключені, тактовий генератор зупинений (даний режим характеризується найменшим енергоспоживанням, вийти з режиму простоювання можна тільки по сигналу скидання).

На рис.10. наведено схему підключення Ethernet-контролера CP2201 до мікроконтролера C8051F340 в макеті мікроконтролерного LXI-модуля.

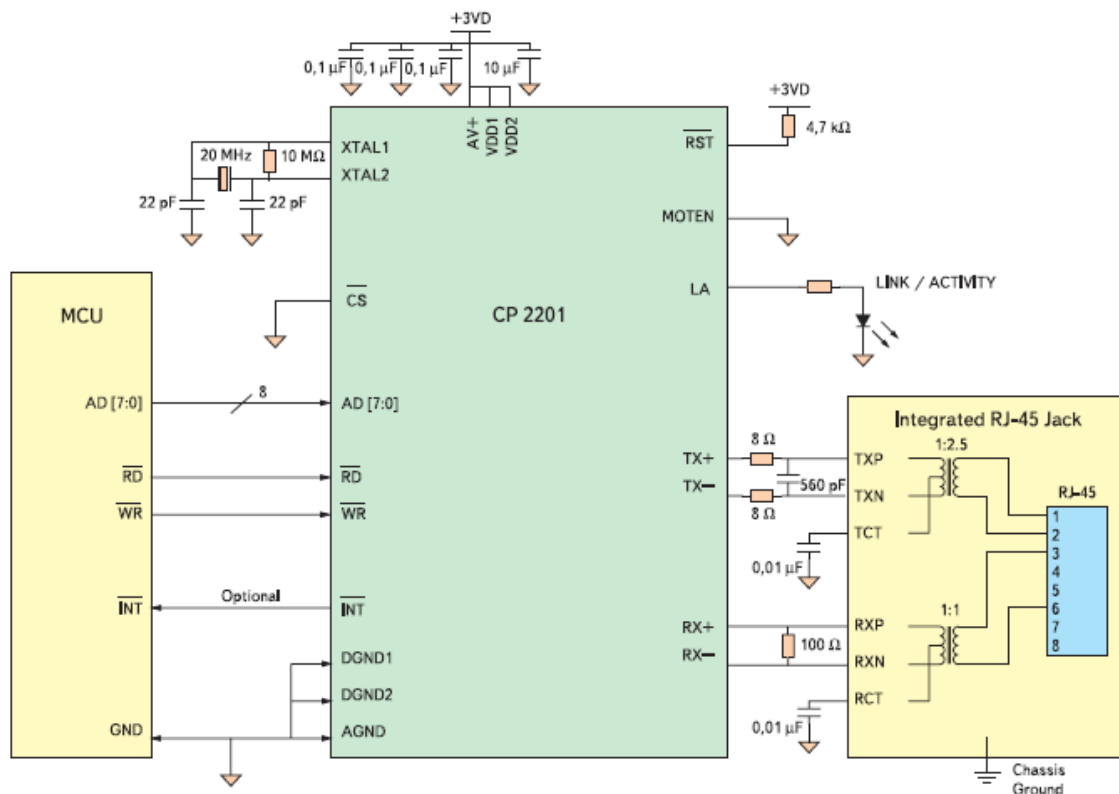


Рисунок 10. Схема підключення CP2201 до мікроконтролера C8051F340

Розробка програмного забезпечення інформаційного блоку демонстраційного мікроконтролерного LXI-сумісного модуля.

Сигнали від вимірювальних каналів подаються на аналоговий вхід P2.2 мікроконтролера C8051F340. В мікроконтролер вбудований мультиплексор, який здійснює комутацію входів до одного входу АЦП. 10-розрядний АЦП перетворює аналогові сигнали в цифровий код. Кодовані сигнали обробляються мікроконтролером, створюється пакет даних та передається на Ethernet-контролер. Після створення пакету даних Ethernet, він передається на роз'єм типу RJ-45. Використано вбудований в мікроконтролер C8051F340 датчик температури та датчик освітленості EL7900, що підключений до лінії P2.2. Для одержання даних розроблено функцію `meashure()` [204]. Частина функції `meashure()`, яка реалізує вимірювання температури, наведено на рис.11.

```

case CAPTURE_TEMP:
    accumulator += ADC0;
    ADC_sample_counter++;
    if(ADC_sample_counter == NUM_TEMPSENSOR_SAMPLES){
        // Accumulator contains (adc_reading << NUM_TEMPSENSOR_SAMPLES_EXP)
        // or, adc_reading [16-bit]
        //
        // Step 1. Voltage [mV] = (adc_reading / 65536 * Vref[mV]);
        //
        // Step2. (Temperature*1000)=(Voltage[mV]-Offset[mV]) *
        //         gain_inverted [degC/mV * 1000]
        //
        // Step 1. Calculate voltage [mV]
        accumulator >>= (NUM_TEMPSENSOR_SAMPLES_EXPONENT-6);
        accumulator *= VREF_MV;      // multiply by Vref[mV]
        accumulator >>= 16;          // divide by 65536

        // Step 2. Calculate temperature * 1000
        accumulator -= TEMP_SENSOR_OFFSET;
        accumulator *= (TEMP_SENSOR_GAIN);

        // accumulator contains the temperature in thousands of a degree C
        TEMPERATURE_INTEGER = accumulator / 1000;
        TEMPERATURE_FRACTION = (accumulator / 100) -
            [(TEMPERATURE_INTEGER*10)];

        // Change State and reset sample counter
        ADC_state = SWITCH_TO_LIGHT;
        AMXOP = LIGHT_SENSOR_MUX;
        accumulator = 0;
        ADC_sample_counter = 0;
    }
    break;

```

Рисунок 11. Вимірювання температури

Частину функції meashure(), яка реалізує вимірювання освітленості, наведено на рис.12.

```

case WAIT_RESULT_LIGHT:
    while(ADOBUSY);
    ADC_state = CAPTURE_LIGHT;
    break;

case CAPTURE_LIGHT:
    accumulator += ADC0;
    ADC_sample_counter++;
    if(ADC_sample_counter == NUM_LIGHTSENSOR_SAMPLES){

        // Calculate light intensity
        temp_int = accumulator;
        temp_int >>= 1;
        if(temp_int > 0xFF) temp_int = 0xFF;
        LIGHTSENSOR_BYTE = temp_int;
        LIGHTSENSOR_PERCENT =
            (LIGHTSENSOR_BYTE==0xFF)?100:
            (((unsigned int)LIGHTSENSOR_BYTE*100)>>8));

        // Change State
        ADC_state = DATA_SEND;
        AMXOP = TEMP_SENSOR_MUX;
        accumulator = 0;
        ADC_sample_counter = 0;
    }
    break;

```

Рисунок 12. Вимірювання освітленості

Розробка програмного забезпечення мережевого блоку мікроконтролерних модулів.

Для мікроконтролерної реалізації базової функціональності стеку протоколів TCP/IP існує декілька поширених бібліотек. Зокрема – це бібліотека стеку uIP різних версій і бібліотека TCP/IP стеку фірми Silicon Laboratories. Бібліотека uIP надає реалізацію базових протоколів стека TCP/IP і орієнтована на малорозрядні 8- і 16-розрядні мікроконтролери. Вона надає базові протоколи для мережевого зв'язку з невеликим розміром коду і вимогами до пам'яті. Розмір коду uIP складає декілька кілобайт пам'яті кодів і потребує декількох сотень байт оперативної пам'яті. Бібліотека uIP є програмним забезпеченням з відкритим вихідним кодом, вона розроблена на мові програмування C. Документацію і вихідні коди можна вільно використовувати і поширювати для комерційного та некомерційного використання з умовою збереження посилання на автора. Бібліотеку було портовано на широкий спектр 8-розрядних мікроконтролерів, вона використовується у великій кількості продуктів та проектів [205].

Бібліотека uIP має такі особливості: добре документований і добре прокоментований вихідний код; малий розмір коду; низькі вимоги до пам'яті, які налаштовуються під час компіляції; підтримка протоколів ARP, SLIP, IP, UDP, ICMP і TCP; бібліотека включає в себе набір прикладів додатків: веб-сервер, клієнт електронної пошти (SMTP клієнт), Telnet сервер, DNS; підтримується будь-яка кількість активних з'єднань TCP, максимальне число яких налаштовується під час компіляції; підтримується будь-яка кількість пасивних слухачів (серверів) TCP, максимальне число яких налаштовується під час компіляції; безкоштовність для комерційного та некомерційного використання; RFC-сумісна реалізація протоколу TCP і IP.

Програмне забезпечення мережевих блоків мікроконтролерних LXI-модулів реалізовано на базі стека uIP версії 0.9. Цей вибір зумовлений тим, що для бібліотеки uIP надаються відкриті вихідні коди, що забезпечує можливість адаптації базового бібліотечного коду до вимог стандарту LXI.

Програмне забезпечення мікросервера складається з двох частин

серверного та клієнтського. Серверне програмне забезпечення виконує керування всією внутрішньою периферією модуля CP2201EB. Клієнтське програмне забезпечення реалізує запити на обмін з серверною частиною. Базовим програмним забезпеченням серверної частини є TCP/IP стек uIP версії 0.9, весь інший функціонал в залежності від поставлених задач втілюється розробником.

Структурну схему організації програмного забезпечення мікроконтролерного LXI-сумісного інтерфейсного блоку наведено на рис.13.

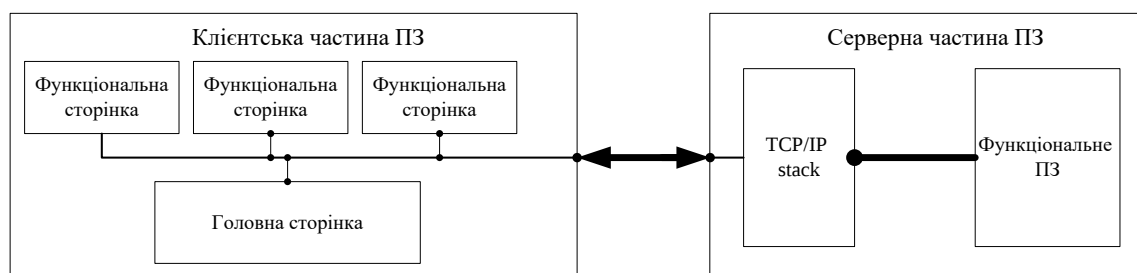


Рисунок 13. Структура програмного забезпечення мережевого мікроконтролерного блоку

Для сумісної роботи Ethernet-контролера CP2201 та бібліотеки uIP розроблено програмне забезпечення, що реалізує обмін повідомленнями між канальним рівнем стеку uIP та фізичним рівнем Ethernet-контролера CP2201 [206]. Цю бібліотеку розміщено в файлах cp2201.c та cp2201.h. В файлі заголовку cp2201.h знаходяться визначення регістрів Ethernet-контролера та структури даних.

Файл cp2201.c складається з трьох функцій – InitCP2201(), що виконує ініціалізацію контролера, CP2201_transmit(), що виконує передачу даних і CP2201_receive(), що виконує прийом даних. Крім того, так як для обробки переривання від CP2201 використовується зовнішнє переривання INT0 мікроконтролера, до вказаних функцій додано функцію обробки переривання ExtInt0_ISR() та службову функцію _wait_ms(count), що призначена для генерації часових затримок [204].

Організація передачі повідомлень для LXI подій.

В повідомленнях LXI подій використовується формат даних для прямої передачі даних від модуля до модуля. Ці повідомлення є повідомленнями подій LXI (LXI Event Messages), які передаються багатьом адресатам в локальній мережі за протоколом UDP або передаються через TCP з'єднання точка-точка. Для UDP-обміну загальний розмір даних не повинен перевищувати розмір одного пакету даних LAN мережі. Повідомлення LXI Event Messages повинно кодуватися в одному пакеті локальної мережі для досягнення мінімальної затримки і уникнення ускладнень при багатоадресному спілкуванні по UDP.

Якщо LAN пакети занадто великі, то вони фрагментуються. Однак, максимальна довжина – параметр, що задається користувачем. Це значення за замовчуванням – 1400 байт. Його можна зменшувати або збільшувати, якщо це необхідно. Обов'язковою вимогою до LAN приладів є їхня змога приймати пакети не менші за 512 байт без фрагментації. Якщо всі пристрої в системі сконфігуровані відповідним чином, то розмір пакету може бути більшим.

Розмір пакету залежить від параметрів LAN мережі, яка використовується, та від того, чи реалізовано протокол IPv6. В цілому є підстави очікувати, що принаймні 450 байт будуть доступні для пакетів даних. Частина цього простору виділяється для обов'язкових полів даних, таких як мітки. Інший простір доступний для користувацьких цілей за умови дотримання форматування, опис якого наведено нижче. Для TCP спілкування будь-які фрагменти пакетів будуть автоматично зібрані стеком приймаючого модуля. В цьому випадку, передача LXI Event Messages може включати в себе дані будь-якого розміру. Проте, чим довші дані, тим повільніше вони надходять до приймача. Тому при обміні, для якого час доставки є критичним, використовуються пакети з невеликим розміром.

LXI Event Messages повинні містити в собі поля, які вказано на рис.14.

HW Detect	Domain	Event ID	Sequence	Timestamp	Epoch	Flags	Data Fields...	0 (два байта)
--------------	--------	-------------	----------	-----------	-------	-------	-------------------	------------------

Рисунок 14. Структура пакету LXI Event Messages

Вважається, що розмір одного байта – 8 біт (октет). Крім того, всі багатобайтові поля розташовані у зворотному порядку (найстарший байт передається першим). У кожному байті передається першим найменший значущий біт. Для масиву октетів полів найбільш значуще поле передається першим. Таким є поле з індексом 0.

HW Detect. Масив октетів довжиною 3. Використовується в якості «магічного числа», щоб визначити коректні пакети, а також зарезервовано для майбутніх апаратних реалізацій LXI Event пакетів. Це поле повинно містити значення «LXI».

Domain. UInteger8. Значення за замовчуванням повинно бути нулем.

Event ID. Масив октетів довжиною 16. Містить ідентифікатор LXI події. Він повинен включати перші 16 байт імені LXI Event (послідовність ASCII – символів), визначених в стандарті LXI. Імена довші за 16 байт будуть урізані до перших 16 символів. В іменах, що менші за 16 байт, невикористані октети будуть встановлені в 0x00.

Sequence. UInteger32. Містить порядковий номер. Кожен сеанс передачі даних від приладу повинен підтримувати незалежний порядковий лічильник(и):

- один для кожної комбінації багатоадресового мережевого UDP інтерфейсу;
- один для кожного з'єднання TCP.

Після передачі повідомлення прилад повинен збільшувати значення лічильника. Початкове значення лічильника допускається довільним.

Timestamp. 10 октетів. Часова мітка, яка ідентифікує час, коли відбулась чи відбудеться подія LXI Event. Поле секунд – молодші 32 біта поля секунд за стандартом IEEE 1588. Поле наносекунд – поле наносекунд часових відміток за стандартом IEEE 1588. Поле наносекунд завжди менше, ніж 10^9 . Дробове поле (fractional_nanoseconds) містить частку наносекунди, яка забезпечена механізмом часових відміток годинника IEEE 1588. Інтерфейс обміну програми та локального годинника може не містити дробового значення наносекунди. В такому випадку це поле повинно містити нуль.

Наприклад, +2.0 секунди повинні бути представлені наступним чином: seconds = 0x00000002, nanoseconds = 0x00000000; -2.0 секунди: seconds = 0x00000002, nanoseconds = 0x80000000; +2.000000001 секунди: seconds = 0x00000002, nanoseconds = 0x00000001.

Значення відмітки часу «нуль» повинно тлумачитися як «зараз», тобто час, коли одержувач обробляє повідомлення.

Epoch. UInteger16. Поле, яке містить найбільш значущі 16 біт поля секунд за стандартом IEEE 1588. Для пристроїв, що не здатні встановлювати відмітки часу, це поле дорівнює нулю.

Flags. UInteger16. Поле, що містить дані про пакет. Визначається таким чином:

Біт 0 – Повідомлення про помилку. Якщо біт встановлений в 1, пакет є повідомленням про помилку.

Біт 1 – Зарезервований. Повинен бути встановлений в нуль.

Біт 2 – Апаратне значення. Логічне значення, яке характеризує тригер подій (зокрема, апаратні події).

Біт 3 – Підтвердження. Якщо біт встановлений в 1, то попередній пакет був успішно отриманий. Це дозволяє LXI системам реалізовувати на основі UDP протоколи handshaking (з підвищеною надійністю). Модулі не зобов'язані реалізовувати цю можливість, однак, вони повинні ігнорувати пакети, в яких цей біт встановлено.

Біт 4 – Подія без стану. Якщо біт встановлено в 0, вміст апаратного значення (біт 2) має контролюватися модулями приймачів. Якщо встановлено в 1, то подія LXI передається без стану і вміст апаратного значення (біт 2) повинен ігноруватися приймачами повідомлення.

Біт 5-15 – Зарезервовано. Всі біти повинні бути встановлені в 0.

Data Fields. Довільна кількість байт. Кожне поле даних повинно мати наступний формат:

Data Length (UInteger16) – розмір даних користувача. Якщо пакет не містить даних, то це поле повинно бути нульовим.

Identifier (Integer8) – ідентифікатор типу даних, що передаються. Числа від 0 до 127 доступні для користувацьких цілей. Від’ємні числа зарезервовані специфікацією консорціума LXI.

User Data – масив байтів, розмір якого задається полем Data Length.

LXI прилади повинні ігнорувати будь-які повідомлення LXI Event Message, якщо значення поля HW не співпадає зі значенням, що має відношення до специфікації LXI. На даний момент поле HW повинно містити «LXI». Якщо поле Event ID містить нулі, то подія вважається «нульовою». Всі LXI прилади повинні ігнорувати такі події, але вони повинні бути записані в лог-файл.

Один із можливих способів підвищення надійності передачі даних по UDP полягає в програмуванні приймальних модулів для підтвердження отримання пакету. Якщо це реалізовано, то повинен встановлюватися прапор підтвердження (біт 3 поля Flags). Якщо не реалізовано, то пакети з встановленим прапором будуть ігноруватися.

Прилад також повинен підтримувати статичне і динамічне призначення IP-адреси. При статичному призначенні IP-адреси всі необхідні параметри (маска мережі, ім'я, шлюз, DNS) повинні бути вказані вручну. Такий підхід досить поширений і використовується в невеликих мережах, коли немає необхідності підтримувати велику базу використаних адрес.

Ідентифікатори типу даних наведені в табл.1.

Таблиця 1.

Ідентифікатори типу даних

Значення	Тип даних	Розмір	Примітка
-1 (0xFF)	ASCII Data	1	Рядок ASCII символів
-2 (0xFE)	int8	1	Доповнювальний код
-3 (0xFD)	uint8	1	Big-endian
-4 (0xFC)	int16	2	Доповнювальний код; багатобайтові поля з big-endian

Продовження таблиці 1

C-5 (0xFB)	uint16	2	Багатобайтові поля з big-endian
-6 (0xFA)	int32	4	Доповнювальний код; декількох багатобайтові поля з big-endian
-7 (0xF9)	uint32	4	Багатобайтові поля з big-endian
-8 (0xF8)	int64	8	Доповнювальний код; багатобайтові поля з big-endian
-9 (0xF7)	uint64	8	Багатобайтові поля з big-endian
-10 (0xF6)	float32	4	Формат IEEE 754; багатобайтові поля з big-endian
-11 (0xF5)	float64	8	Формат IEEE 754; багатобайтові поля з big-endian
-12 (0xF4)	float128	16	Формат IEEE 754; багатобайтові поля з big-endian
-13(0xF3)	UTF-8 Data	1	Unicode String Data закодований в UTF-8
-14 (0xF2)	UTF-8 JSON	1	JSON закодований в UTF-8
-15 (0xF1)	UTF-8 XML	1	XML закодований в UTF-8
-16 (0xF0)	Octet	n	Неінтерпретований октет

При зростанні кількості комп'ютерів/обладнання в мережі, більш доцільно використання протоколу DHCP (Dynamic Host Configuration Protocol) – протоколу динамічної конфігурації мережі. Стандарт LXI також рекомендує включати підтримку протоколу DDNS – динамічних серверів імен, щоб надати можливість обладнанню мати власні імена в мережі.

Оснащення LXI-пристроїв вбудованим веб-сервером забезпечує користувачу доступ до веб-сторінок налаштувань і управління. Такий підхід не тільки дозволяє уникнути необхідності встановлення додаткового програмного забезпечення на клієнтський комп'ютер, а є кросплатформеним рішенням, адже контент сервера може бути прочитаний будь-яким браузером в будь-якій операційній системі. Використовуючи динамічні веб-сторінки користувач може змінити всі параметри налаштування пристрою. Одна з особливостей динамічних веб-сторінок полягає в тому, що крім статичної інформації в них

можна вбудовувати програми, створені на мовах Java або Flash. Ці додатки можуть реалізовувати складний графічний інтерфейс користувача. Це робить їх придатними для створення панелей оператора, які імітують лицьову панель реального приладу.

Програмне забезпечення для управління LXI-пристроями будується на основі технології VISA (Virtual Instrument Software Architecture), яка визначає набір команд і функцій для роботи з введенням-виведенням і передачею даних по мережі. Використання технології VISA надає переваги стандартизації процесу програмування. Стандарт LXI вимагає від виробника виробу розробки одного з двох типів VISA-драйверів – IVI-COM і IVI-C. Перший заснований на поширеній архітектурі COM, що була розроблена фірмою Microsoft і дозволяє використовувати будь-яку з сучасних мов програмування. Структура драйверу являє собою ієрархію класів. Це дозволяє отримати всі переваги об'єктно-орієнтованого підходу до програмування. IVI-C драйвера втілюють більш низькорівневе рішення та створюються на мовах C або C++ [207].

У порівнянні з іншими архітектурами використання стандарту LXI для створення систем дистанційного збору і обробки експериментальних даних надає ряд суттєвих переваг.

Скорочення накладних витрат. Незалежно від обраного конструктивного виконання, прилади LXI сприяють зниженню вартості системи. Модульні пристрої не вимагають базових блоків, або спеціалізованих інтерфейсів. Модулі приладів можна налаштувати за допомогою програм для емуляції широкого кола вимірювальних задач. Дистанційні пристрої на базі LXI забезпечують економічний спосіб розміщення датчиків, камер, мікрофонів і інших пристроїв. Настільне виконання забезпечує створення точних приладів з можливістю підключення до LAN.

Повторне використання існуючих приладів та програмного забезпечення. В деяких випадках вимірювальні прилади можуть використовуватися для розробки та налагодження програм випробувань, які потім можуть використовуватися у системі, побудованій з використанням пристроїв LXI. Крім

того, при використанні необхідних програмних модулів нові функціональні блоки можуть виконувати роботу декількох приладів одночасно..

Скорочення часу налаштування. Завдяки використанню перевірених часом стандартів, таких як Ethernet і драйвери IVI, забезпечується сумісність усіх компонентів, і налаштування системи займає менше часу. При наявності веб-сторінок, вбудованих в кожен прилад LXI, за допомогою стандартного веб-браузера можна переглядати всю інформацію про пристрій, змінювати його конфігурацію, керувати вимірюваннями і одержувати їх результати.

Збільшення продуктивності системи. Стандарт LXI дозволяє створювати високошвидкісні розподілені системи, що складаються з приладів, які взаємодіють один з одним і працюють паралельно. Синхронізація пристроїв забезпечується способами запуску, передбаченими стандартами IEEE 1588 і локальної мережі, одноранговими і багатоадресними повідомленнями, а також сигналами апаратної шини запуску.

Зменшення розмірів системи. Модульні прилади LXI дозволяють зменшити масу і габарити системи і вимоги до приміщення для їх розміщення без погіршення якості вимірювань.

Вирішення проблем модернізації системи. В основі збільшення терміну служби систем на базі LXI закладено дві основні ідеї: можливість додавання нових функцій або спеціалізованих вимірювальних програм, а також можливість впровадження оновлених або більш досконалих технологій в системи. Ці можливості спрощують задачу підтримки системи.

При реалізації інформаційно-вимірювальної систем на основі стандарту LXI доцільною є розробка мікроконтролерних LXI-сумісних інтерфейсних блоків, що підтримують обмін за стандартом Ethernet згідно специфікаціям, визначеним стандартом. Таке рішення дозволить забезпечити інтеграцію різноманітних вимірювальних пристроїв в системи на основі стандарту LXI.

Розробка програмного забезпечення підтримки LXI-повідомлень мікроконтролерними мережевими модулями.

Для передачі повідомлень за стандартом LXI використовується спеціальний формат фрейму, що міститься в мережевому UDP-пакеті. Для забезпечення максимальної швидкодії та уникнення, бажано, щоб повідомлення LXI знаходились в одному пакеті UDP. Якщо пакети даних занадто великі, вони можуть бути фрагментованими. Зазвичай пакет має розмір у 1400 байтів. Цей розмір може вибиратись довільно, але не повинен бути меншим за 512 байт.

Для TCP-комунікацій, будь-які фрагментовані пакети автоматично дефрагментуються на стороні отримувача. В цьому випадку повідомлення LXI можуть включати в себе дані будь-якого розміру. Однак, чим більше розмір даних, тим більше часу потрібно, щоб їх передати. Для комунікацій, де час передачі є критичним, як правило, використовують невеликі розміри пакетів.

Відповідно до стандарту LXI, модуль повинен підтримувати UDP-пакети визначеного стандартом формату. Демонстраційний макет мікроконтролерного LXI-модуля підтримує:

- широкомовну відправку пакетів з інформацією про стан пристрою з періодом в 1 секунду;
- прийом пакетів з запитом даних та відповідь на них [208].

Широкомовний пакет для LXI-модуля має ідентифікатор події "Device state" і містить в собі три двобайтових цілих числа – результати вимірювань температури, освітлення та стан світлодіода.

Вхідний пакет повинен мати ідентифікатор події "Data Request" і містити в собі одне ціле однобайтове число, що відповідає наступним значенням: 0 – запит температури; 1 – запит освітленості; 2 – запит стану світлодіоду.

Залежно від вигляду вхідного пакету формується пакет відповіді, що може мати наступні ідентифікатори події:

- "Temperature data" – відповідь на запит температури;
- "Illumination data" – відповідь на запит освітленості;
- "LED data" – відповідь на запит стану світлодіоду.

Пакет відповіді в полі даних містить одне ціле двобайтове число.

Крім того, всі пакети, що приходять з флагом підтвердження ігноруються, незалежно від розміщених в ньому даних.

Генерація такого пакета здійснюється за допомогою програмного забезпечення, що знаходиться в файлах lxi.c та lxi.h. В файлі lxi.h описуються основні константи та структура даних фрейму (рис.15).

```
typedef struct{
    u8_t  HWDetect[3];
    u8_t  domain;
    u8_t  event_ID[16];
    u32_t seq;
    u32_t ts_sec;
    u32_t ts_nsec;
    u16_t ts_fr_nsec;
    u16_t epoch;
    u16_t flags;
    u16_t length;
    u8_t  identifier;
    u8_t  dta[128];
} lxi_event;

extern bit LXIEventPending;
extern unsigned int data illuminance;
extern unsigned int data temperature;
sbit LED_ON = P0^2;
```

Рисунок 15. Структура даних фрейму

Далі визначаються константи для визначення стану, тип запиту даних та прототипи функцій (рис.16).

```
void sendTest(void);
void lxi_app(void);
static void create_msg(char dta[], char len);
void lxi_request(void);
void lxi_appcall(void);
```

Рисунок 16. Прототипи функцій

В файлі lxi.c визначаються глобальні змінні, а також структура опису стану програми (рис.17).

```
bit LXIEventPending = 0;
xdata lxi_event *msg;
xdata u32_t sequence = 0;

xdata struct {
    u8_t state;
    struct uip_udp_conn *conn;
} s;
```

Рисунок 17. Опис стану програми

Функція `lxi_request()` викликається, коли треба надіслати до мережі широкомовний пакет з інформацією про стан LXI-модуля. Функція створює нове підключення і визначає стан програми (рис.18).

```
void lxi_request(void) {
    u16_t addr[2];
    s.state = LXI_STATE_BROADCAST;
    LXIEventPending = 0;
    uip_ipaddr(addr, 255,255,255,255);
    if(s.conn == NULL) {
        s.conn = uip_udp_new(addr, HTONS(IANAportlxi_evntsvc));
        uip_udp_bind(s.conn, HTONS(IANAportlxi_evntsvc));
    }
}
```

Рисунок 18. Запит широкомовного пакету

Функція `lxi_aprcall()` викликається, якщо створене підключення, або надійшли вхідні дані на порт 5044. Функція визначає, чи встановлено флаг підтвердження (рис.19).

```
void lxi_appcall(void) {
    char i = 0;
    char buf[8];
    msg = (struct lxi_event *) uip_appdata;

    if( (msg->HWDetect[0] == 'L') &&
        (msg->HWDetect[1] == 'X') &&
        (msg->HWDetect[2] == 'I') ) {
        if(msg->flags && 0x08) {
            msg->flags = 0;
            if(s.conn != NULL) {
                s.conn = NULL;
            }
            s.state = LXI_STATE_NONE;
            return;
        }
    }
```

Рисунок 19. Перевірка флагу підтвердження

Визначається, чи має вхідний пакет ідентифікатор події "Data Request" (рис.20).

```
if( (msg->event_ID[0] == 'D') &&
    (msg->event_ID[1] == 'a') &&
    (msg->event_ID[2] == 't') &&
    (msg->event_ID[3] == 'a') &&
    (msg->event_ID[4] == ' ') &&
    (msg->event_ID[5] == 'R') &&
    (msg->event_ID[6] == 'e') &&
    (msg->event_ID[7] == 'q') &&
    (msg->event_ID[8] == 'u') &&
    (msg->event_ID[9] == 'e') &&
    (msg->event_ID[10] == 's') &&
    (msg->event_ID[11] == 't')) {
```

Рисунок 20. Перевірка ідентифікатора події

Якщо це так, то визначається тип фрейму, що буде відправлено у відповідь (рис.21).

```
switch (msg->data[0]) {
    case LXI_STATE_DATAREQ_TEMP:
        LXIEventPending = 1;
        s.state = LXI_STATE_DATARESP_TEMP;
        break;
    case LXI_STATE_DATAREQ_LIGH:
        LXIEventPending = 1;
        s.state = LXI_STATE_DATARESP_LIGH;
        break;
    case LXI_STATE_DATAREQ_LED:
        LXIEventPending = 1;
        s.state = LXI_STATE_DATARESP_LED;
        break;
    default:
        if(s.conn != NULL) {
            s.conn = NULL;
        }
        s.state = LXI_STATE_NONE;
        break;
}
```

Рисунок 21. Визначення типу фрейму відповіді

Визначається стан програми. В залежності від цього формуються дані, що будуть міститись в пакеті. Після створення пакет передається до мережі. Наприклад, для широкомовного пакету передається інформація про температуру, освітленість і стан світлодіода (рис.22).

```
if (uip_poll()) {
    switch (s.state) {
        case LXI_STATE_BROADCAST:
            buf[0] = temperature >> 8;
            buf[1] = temperature & 0xFF;
            buf[2] = illuminance >> 8;
            buf[3] = illuminance & 0xFF;
            if(LED_ON) {
                buf[4] = 0xFF;
                buf[5] = 0xFF;
            } else {
                buf[4] = 0x00;
                buf[5] = 0x00;
            }
            buf[6] = 0x00;
            buf[7] = 0x00;
            create_msg(buf, 8);
            uip_udp_send(41+8);
            if(s.conn != NULL) {
                s.conn = NULL;
            }
            s.state = LXI_STATE_NONE;
            break;
    }
}
```

Рисунок 22. Створення широкомовного пакету

Широкомовний пакет, що сформовано мікроконтролерним LXI-модулем, наведено на рис.23.

3366	ARP REQ	BC:AE:C5:14:5A:9A	Broadcast	?	0.0.0.0	?	10.108.3.1	N/A	N/A
3367	ARP REQ	BC:AE:C5:14:5A:9A	Broadcast	?	10.108.3.1	?	10.108.3.254	N/A	N/A
3368	IP/UDP	CygnalInte:00:33:60	Broadcast	?	10.108.3.2	?	255.255.255.255	5044	5044
3369	ARP REQ	BC:AE:C5:14:5A:9A	Broadcast	?	10.108.3.1	?	10.108.3.254	N/A	N/A
3370	IP/UDP	BC:AE:C5:14:5A:9A	Broadcast	?	10.108.3.1	?	10.108.3.255	netbios-ns	netbios-ns
3371	ARP REQ	BC:AE:C5:14:5A:9A	Broadcast	?	10.108.3.1	?	10.108.3.254	N/A	N/A

0x0000	FF FF FF FF FF FF 00 0B-3C 00 33 60 08 00 45 00	яяяяяя...<.3'..E.
0x0010	00 4D 07 B3 00 00 FF 11-A6 7F 0A 6C 03 02 FF FF	.M.i...я. 0.1..яя
0x0020	FF FF 13 B4 13 B4 00 39-00 00 4C 58 49 00 44 65	яя.r.r.9..LXI.De
0x0030	76 69 63 65 20 73 74 61-74 65 00 00 00 00 00 00	vice state.....
0x0040	00 02 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
0x0050	00 08 FB 00 3E 01 56 FF-FF 00 00	..м.>.Vяя..

температура освітленість світлодіод

Рисунок 23. Широкомовний пакет від LXI-модуля

Для тестування відповіді від мікроконтролерного LXI-модуля було сформовано та відправлено пакет запиту даних про освітленість, структуру якого наведено на рис.24.

152	IP/UDP	CygnalInte:00:33:60	Broadcast	?	192.168.88.249	?	255.255.255.255
153	IP/UDP	NexoCommun:BB:B...	CygnalInte:00:33:60	?	1.1.1.1	?	192.168.88.249
154	IP/UDP	CygnalInte:00:33:60	Broadcast	?	192.168.88.249	?	255.255.255.255
155	IP/UDP	NexoCommun:BB:B...	CygnalInte:00:33:60	?	1.1.1.1	?	192.168.88.249

0x0000	00 0B 3C 00 33 60 00 50-00 BB BB BB 08 00 45 00	..<.3'.P.»»»..E.
0x0010	00 4C 90 15 00 00 80 11-8E E8 01 01 01 01 C0 A8	.Lђ...Ђ.и....AЁ
0x0020	58 F9 13 B4 13 B4 00 38-C0 76 4C 58 49 00 44 61	Xм.r.r.8AvLXI.Da
0x0030	74 61 20 52 65 71 75 65-73 74 20 20 20 20 00 00	ta Request ..
0x0040	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
0x0050	00 01 FF 01 00 00 00 00-00 00	..я.....

Destination MAC: 00:0B:3C:00:33:60

Рисунок 24. Пакет запиту даних

Пакет з даними, що був коректно сформований мікроконтролерним LXI-модулем у відповідь наведено на рис.25.

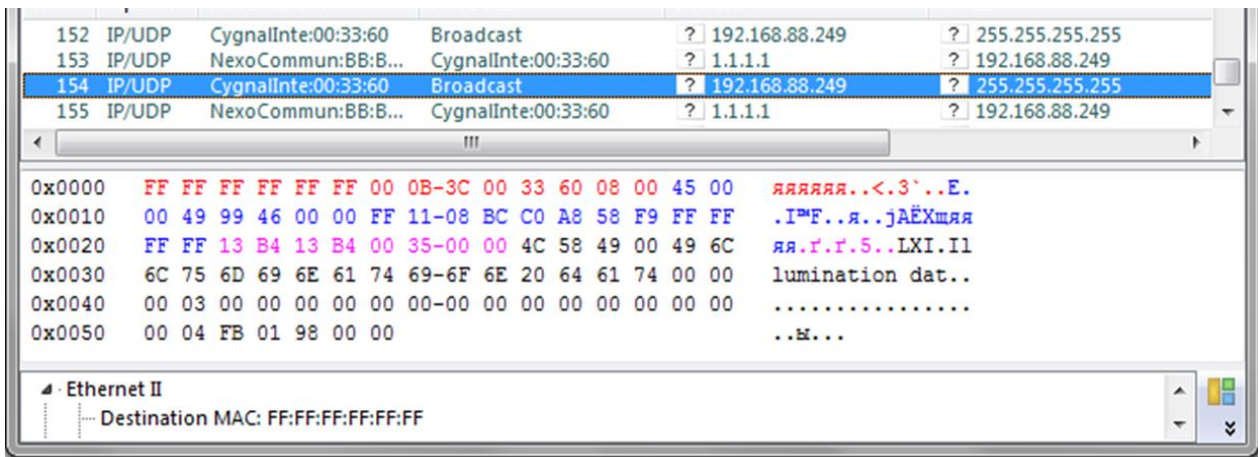


Рисунок 25. Пакет відповіді

Розробка програмного забезпечення динамічного конфігурування мікроконтролерних LXI-модулів на основі веб-форм за технологією CGI.

За стандартом LXI всі пристрої повинні підтримувати генерацію HTML веб-сторінки, яка повинна коректно відображатись W3C-сумісними браузером. Вбудований веб-сервер LXI-модуля повинен підтримувати протокол HTTP (версії 1.0 або вище). HTML-сторінки, що генеруються LXI-модулем повинні відповідати стандарту HTML (версії 4.01 або вище) або XHTML (версії 1.0 або вище).

LXI пристрої повинні підтримувати HTTP-з'єднання на порту 80 і відображати головну сторінку за замовчуванням. Навігаційні кнопки і гіперпосилання можуть надавати доступ до інших портів за бажанням авторів веб-сторінки.

На головній веб-сторінці приладу повинні бути відображені наступні дані: номер моделі приладу; інформацію про виробника; серійний номер приладу; опис приладу; LXI-клас приладу (A, B або C); версію стандарту LXI, згідно якої реалізовувався даний пристрій; назву хоста; MAC-адресу; TCP/IP адреси; номер версії програмної прошивки або програмного забезпечення; час IEEE 1588 (тільки для приладів класів A і B); адресу пристрою LXI (згідно стандарту VISA). Головна веб-сторінка приладу повинна містити гіперпосилання, що дозволять користувачеві налаштовувати параметри пристрою LXI.

Перша пов'язана веб-сторінка повинна містити налаштування параметрів мережі, а друга пов'язана веб-сторінка повинна містити налаштування синхронізації пристрою. Сторінка налаштування мережі повинен містити наступні параметри: назву хоста; опис; режим конфігурації TCP/IP; IP-адресу; маску підмережі; шлюз за замовчуванням; DNS-сервер(и). Режим конфігурації вказує яким чином буде назначено IP-адресу пристрою – автоматично чи вручну. В ручному режимі для налаштування мережі користувач вводить статичну IP-адресу, маску підмережі і шлюз за замовчуванням. В автоматичному режимі для налаштування мережі використовується DHCP сервер або Dynamic Link Local Addressing (автоматичний IP).

Стандартом LXI рекомендується, щоб користувач мав доступ до сторінки статусу пристрою. Сторінка статусу повинна містити наступну інформацію: статус пристрою; помилки/попередження. Поле статусу повинно містити статус «зайнято» для будь-якого стану очікування для приладу, або будь-яку іншу інформацію статусу, специфічну для даного пристрою.

Веб-сторінки пристрою були розроблена на базі бібліотеки httpd, яка є частиною стеку uIP. Ця бібліотека була адаптована для роботи з запитамі методу POST, а також CGI-функціями для виведення динамічних даних.

Розроблені CGI-функції для одержання результатів вимірювань освітленості і температури наведені на рис.26.

```
static u8_t get_illuminance(u8_t next) {
    if(next) {
        return 1;
    }
    uip_send(uip_appdata, sprintf((char *)uip_appdata, "%d",
        illuminance));
    return 0;
}

static u8_t get_temperature(u8_t next) {
    if(next) {
        return 1;
    }
    uip_send(uip_appdata, sprintf((char *)uip_appdata, "%d",
        temperature));
    return 0;
}
```

Рисунок 26. Функції одержання результатів вимірювань

CGI-функція для відображення стану LXI-модуля надає наступну інформацію: дані про температуру та освітленість, стан світлодіоду, кількість відправлених і отриманих повідомлень LXI (рис.27).

```
static u8_t device_state(u8_t next){
    char data ledState[30];

    if(next) {
        return 1;
    }
    if(LED_00){
        sprintf((char *)ledState,"<font color=\"green\">ON</font>\0");
    } else {
        sprintf((char *)ledState,"<font color=\"red\">OFF</font>\0");
    }
    uip_send(uip_appdata, sprintf((char *)uip_appdata,
        "<p><b>Temperature value:</b> %d °C</p>"
        "<p><b>Illuminance value:</b> %d lux</p>"
        "<p>The LED is %s</p>"
        "<p><a href=\"/cgi/led_on\">Turn the LED on</a></p>"
        "<p><a href=\"/cgi/led_off\">Turn the LED off</a></p>"
        "<p><b>LXI event messages sent:</b> %d</p>"
        "<p><b>LXI event messages received:</b> %d</p>",
        temperature,
        illuminance,
        ledState,
        sent_lxi_packets,
        received_lxi_packets
    ));
    return 0;
}
```

Рисунок 27. Функція відображення стану модуля

Веб-сторінка відображення стану для макету мікроконтролерного LXI-модуля наведено на рис.28.

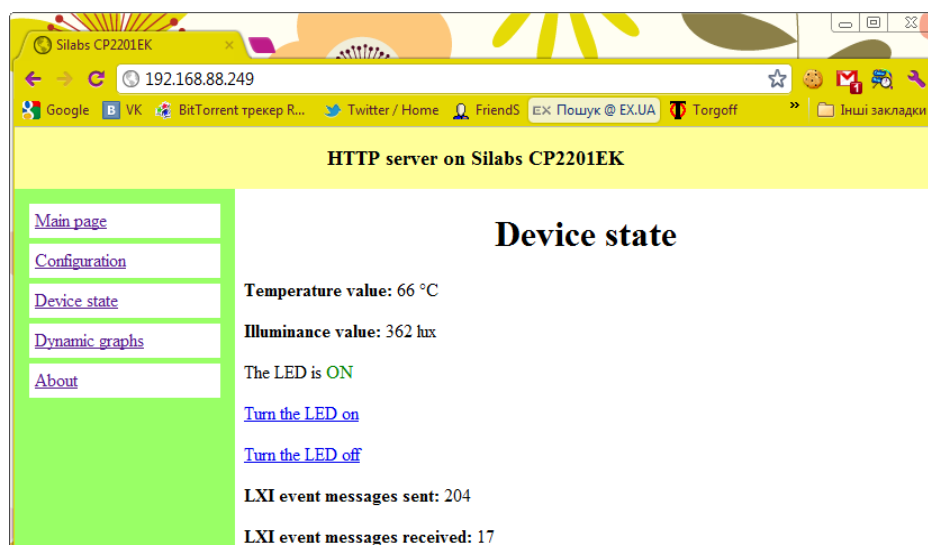


Рисунок 28. Сторінка стану LXI-модуля

CGI-функція для відображення головної сторінки виводить наступну інформацію: хост пристрою, IP-адресу, маску підмережі, шлюз за замовчуванням та MAC-адресу. Головну сторінку веб-інтерфейсу для мікроконтролерного LXI-модуля наведено на рис.29.

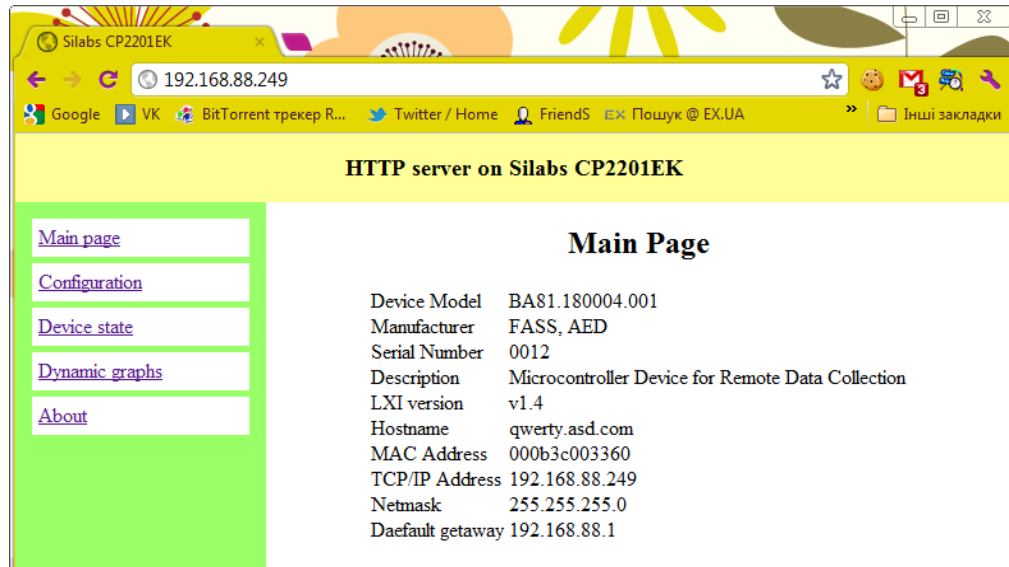


Рисунок 29. Головна сторінка опису мікроконтролерного LXI-модуля

CGI-функція для виведення форми редагування параметрів протоколу TCP/IP виводить частини веб-форми послідовно. Частина функції, яка формує поля для введення IP-адреси, наведена на рис.30, інші частини форми генеруються аналогічно.

```
case 1:
    uip_send(uip_appdata, sprintf((char *)uip_appdata,
    "<Table width=\"500\"><TR><td width=\"200\" valign=\"middle\" \"
    \"align=\"left\">IP Address:</td>\"
    \"<td><input size=\"3\" name=\"T20\" value=\"%i\" /><B>.</B>\"
    \"<input size=\"3\" name=\"T21\" value=\"%i\" /><B>.</B>\"
    \"<input size=\"3\" name=\"T22\" value=\"%i\" /><B>.</B>\"
    \"<input size=\"3\" name=\"T23\" value=\"%i\" /></td></tr></table>\",
        (htons(uip_hostaddr[0]) >> 8),
        (htons(uip_hostaddr[0]) & 0xff),
        (htons(uip_hostaddr[1]) >> 8),
        (htons(uip_hostaddr[1]) & 0xff));
    break;
```

Рисунок 30. Формування полів для введення IP-адреси

Сторінку налаштування мікроконтролерного LXI-модуля наведено на рис.31.

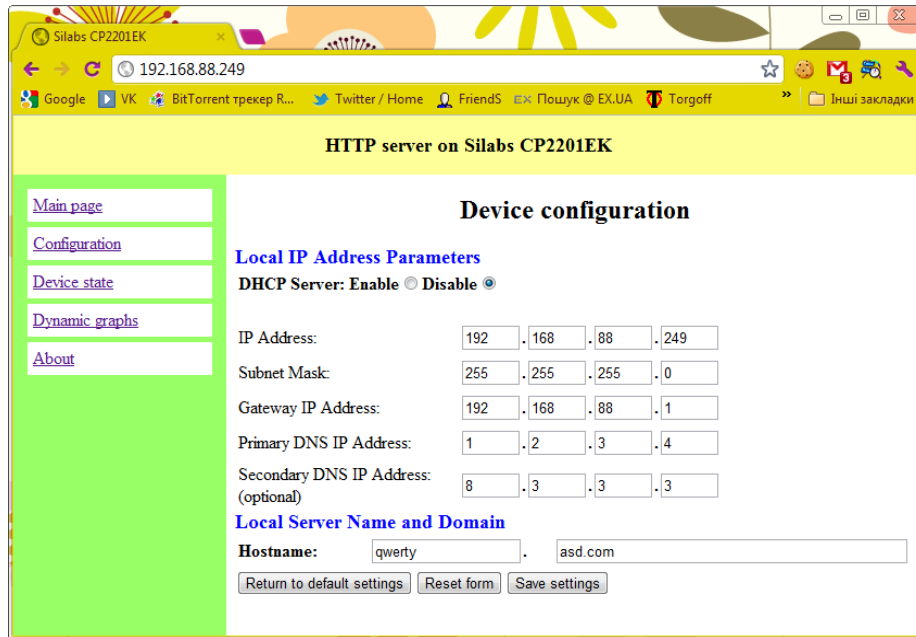


Рисунок 31 – Сторінка налаштування LXI-модуля

Для реалізації динамічного виведення графіку даних використовувався формат SVG – формат файлів для двовимірної векторної графіки. Оновлення даних на SVG-графіку реалізовано за допомогою скрипта, розробленого мовою JavaScript з використанням технології AJAX. Веб-сторінка з графічним відображенням результатів вимірювання температури наведена на рис.32.

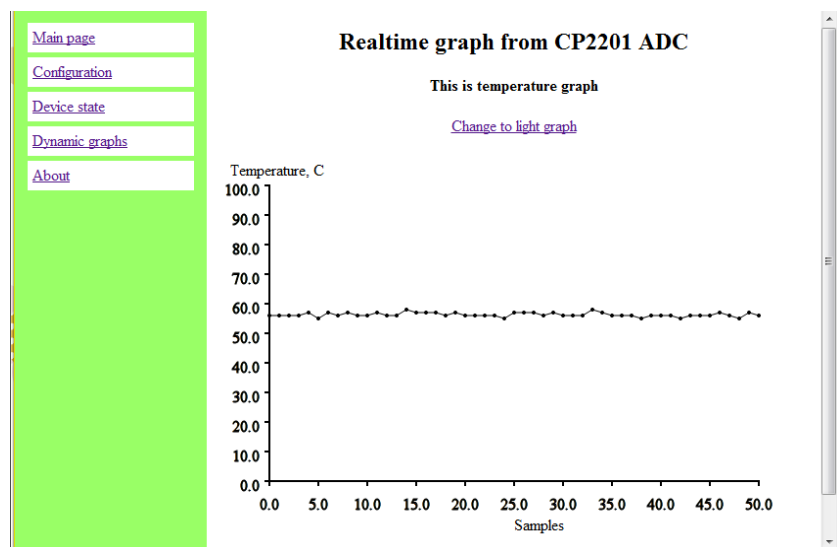


Рисунок 32. Результати вимірювання температури

Висновки. Розроблено типові рішення для реалізації апаратно-програмного забезпечення LXI-сумісних вимірювальних модулів на основі мікроконтролерних технологій. Їх використання дозволяє суттєво спростити організацію мережових Ethernet-систем збору та аналізу даних, розробку систем на основі єдиного відкритого стандарту, забезпечує можливості інтеграції систем з виробами відомих світових виробників, їх апаратну та програмну сумісність.