

## МЕРЕЖЕВИЙ ПРИКЛАДНИЙ ПРОЦЕСОР ДЛЯ РОЗПОДІЛЕНИХ ВИМІРЮВАЛЬНО-КЕРУЮЧИХ СИСТЕМ

Тернопільський національний економічний університет

Представлено мережевий прикладний процесор для розподілених вимірювально-керуючих систем. Створений прототип мережевого прикладного процесору реалізовано як багато-процесорну архітектуру із роздільним виконанням на окремих мікроконтролерах (МК) процесів обробки та обміну даними. Це дозволило зменшити вимоги до продуктивності МК, а також відмовитись від використання операційної системи реального часу

### Вступ

Сучасні вимірювально-керуючі системи (ВКС) будуються як локальні мережі з розподіленими обчислювальними ресурсами, що дозволяє перенести обробку даних від центрального сервера безпосередньо у мережеві контролери, а також зменшити загальний об'єм інформаційних потоків і навантаження на мережу, незважаючи на загальне ускладнення алгоритмів обробки даних та керування. Поштовхом до цього стало зменшення вартості та зростання обчислювальної потужності МК разом із розширенням їх функціональних можливостей. Реалізують ВКС на базі поширених послідовних інтерфейсів *Ethernet*, *CAN*, *LIN*, *1-Wire* та ін.) [1, 2]. Однак на даний час жодна із промислових шин не домінує в даній області. Також достатньо гостро стоїть проблема уніфікації програмного забезпечення (ПЗ) ВКС для забезпечення сумісності обладнання різних виробників. Тому національним інститутом стандартів США розроблено серію стандартів *IEEE1451*, якою уніфіковано: 1) вимоги до послідовних інтерфейсів, що застосовуються у ВКС; 2) структуру ПЗ та типи даних, при

його розробці. Також означено термін МПП – вузла обробки даних для розподілених ВКС [3]. ВКС, що відповідає вимогам серії стандартів *IEEE-1451* (рис. 1), включає: 1) сервер – виконує загальний контроль за станом системи; 2) МПП – виконує обробку даних, що поступають від сенсорів, керування виконавчими механізмами, обмін даними із сервером та іншими МПП; 3) модулі давачів (МД) (в англійській літературі *Transducer Interface Module – TIM*) – взаємодіють із об'єктом керування через сенсори та виконавчі механізми. МПП звертається до МД через локальну мережу нижнього рівня. Кожен із МД має свою адресу та електронний паспорт, що дозволяє реалізувати технологію *Plug-and-Play* [4]. Основними функціями МПП є обробка даних на прикладному рівні та обмін даними із іншими вузлами мережі. Це вимагає від МПП: 1) достатньої обчислювальної потужності; 2) підтримки набору поширених послідовних інтерфейсів (*RS232*, *RS484*, *CAN*, *Ethernet* та ін.). Аналіз відомих рішень МПП [5 – 8] показав, що на даний час можна виділити два підходи до їх реалізації:

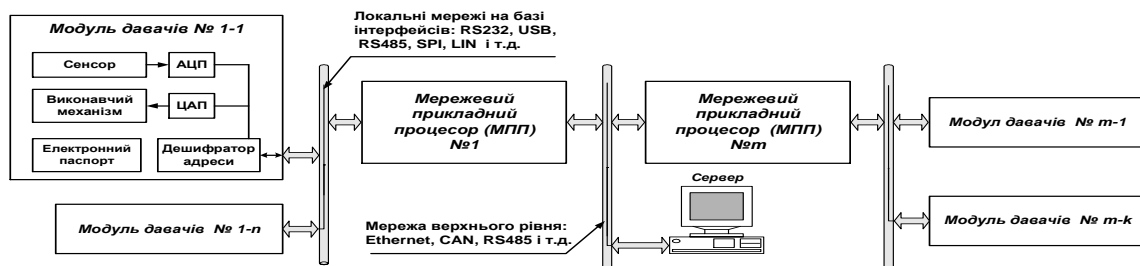


Рис. 1. Структурна ВКС згідно вимог серії стандартів *IEEE-1451*

1) високопродуктивні МПП на базі процесорів *Pentium*, що відповідають вище вка-

заним вимогам; 2) малопродуктивні МПП – на базі дешевих МК, які реалізують частину вище означених вимог. При цьому, вартість МПП, що відносяться до першої та другої групи відрізняється в десятки разів.

Метою роботи є поєднання переваг обох підходів при забезпеченні низької вартості МПП.

### Підходи до реалізації МПП на базі мікроконтролерів

Серед розглянутих раніше двох підходів до реалізації МПП в роботі [9] показано, що доцільно реалізувати МПП як двопроцесорну систему із роздільним виконанням процесів обробки та обміну даними (рис. 2), що включає блок головного контролера (ГК) – виконує обробку даних, та блок інтерфейсного контролера (ІК) – реалізує функції обміну даними через мережу верхнього та нижнього рівнів. При цьому зменшуються вимоги до продуктивності МК, а вибір МК здійснюється відповідно до його функціональних задач, що полегшує пошук оптимального рішення. Однак, обмежений об'єм внутрішнього постійного запам'ятовуючого пристрою (ПЗП) МК, не дозволяє зберігати в МПП повний набір ПЗ. Це вимагає реалізації в МПП функції віддаленого перепрограмування через мережу верхнього рівня в режимі *on-line*, щоб, у випадку необхідності, замінити прикладне програмне забезпечення (ППЗ) МПП. Реалізація означеної функції в однопроцесорній системі можлива із застосуванням операційної системи реального часу (ОСРЧ), що вимагає додаткових обчислювальних ресурсів і економічно виправдано при масовому виробництві [9].

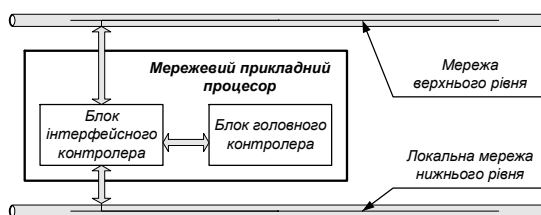


Рис. 2. Структура двопроцесорного МПП  
Також існує імовірність виникнення кон-

фліктів між ППЗ та ОСРЧ, що вимагає додаткових витрат на тестування в процесі розробки ППЗ. У МПП із роздільним виконанням функцій обробки та обміну даними, ППЗ виконується ГК і повністю використовує його апаратні ресурси. Процес перепрограмування ГК виконується апаратними засобами ІК, що дозволяє відмовитись від використання ОСРЧ.

### Реалізація МПП запропонованої структури

Структурна схема реалізованого МПП (рис. 3) розділена на блоки ІК та ГК. Блок ІК реалізовано на двох МК C8051F340 (U8), C8051F412 (U16), що характеризуються високою продуктивністю (до 50MIPS). МК C8051F340 забезпечує підтримку інтерфейсів верхнього рівня: *Ethernet*, *CAN*, *USB*, *RS232/RS485(0)*, а C8051F412 – інтерфейсів нижнього рівня: *RS232/RS485(1)*, *SPI*, *I<sup>2</sup>C*, а також *TII* [4]. Обмін даними між U8 та U16 відбувається через внутрішньосистемну шину *I<sup>2</sup>C*, що недоступна для зовнішніх користувачів. Інтерфейси, *RS232*, *RS485* та *I<sup>2</sup>C* реалізуються самими МК, які мають апаратно вбудовані контролери, а для підтримки інтерфейсів *CAN*, *Ethernet* відповідно використано мікросхеми (МС) контролерів інтерфейсів: *SJA1000* (U2), *CP2200* (U3), до яких ІК (U8) звертається по заданих адресах як до зовнішньої пам'яті даних. Обмін даними здійснюється через 8-и розрядну мультиплексну шину адресу-даних – *IRC\_AD* [7:0]. Інтерфейс *USB* (*host*) реалізується МС *VNC1L*, з якою ІК (U8) взаємодіє через *SPI* інтерфейс. Узгодження фізичних рівнів сигналів інтерфейсів *CAN*, *RS232* та *RS485* виконують МС драйвери інтерфейсів: *CAN* – *PCA82C250* (U1), *RS232* – *ADM3202* (U9, U12), *RS485* – *ADM3485* (U10, U13). Дані, що поступають при обміні, зберігаються у зовнішній оперативній пам'яті ІК (U8), реалізованій на МС статичної пам'яті *AS7C256A* (U7) об'ємом 32 Кбайт, що адресується у верхній області адресного простору U8.

Блок ГК реалізовано на МК *ADuC831* (U17). Пам'ять даних МК ГК

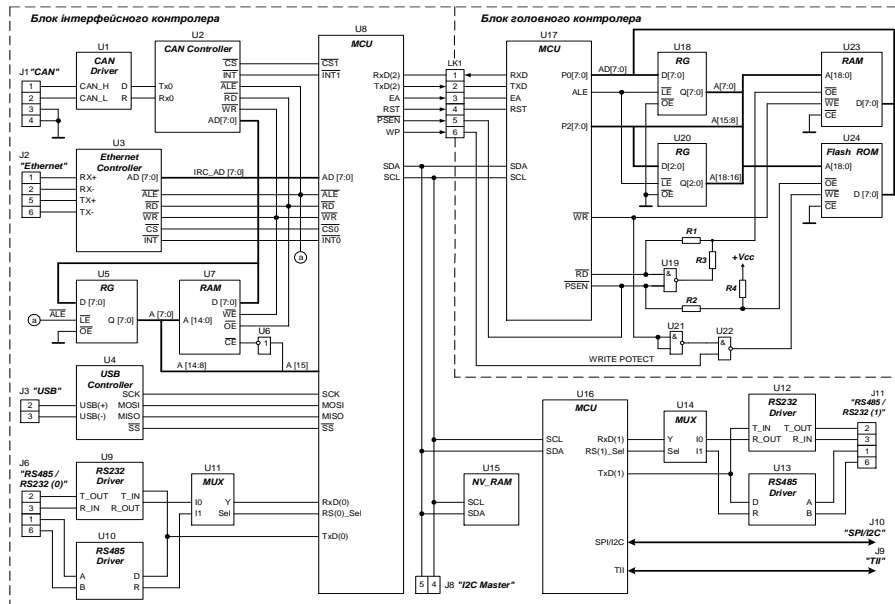


Рис. 3. Узагальнена структурна схема розробленого МПП

реалізовано на МК *BH62UV4000* (*U23*), а пам'ять програм – на МК *M29W040A* (*U24*). Обмін даними обома типами зовнішньої пам'яті здійснюється по 8-и розрядній двонаправлений мультиплексний шині адресу-даних  $AD[7:0]$ . Оскільки МК *ADuC831* може адресувати загальний об'єм зовнішньої пам'яті даних до 16 Мбайт, то для фіксації 16-и молодших розрядів адреси використано два регістри *74HC573* (*U18, U20*). Обмін даними між ГК та ІК (*U8*) здійснюється як через послідовний *UART* інтерфейс  $TxD(2)$ ,  $RxD(2)$ , так і внутрішньосистемну шину *I2C*. Резистори *R1-R4* та логічний елемент (ЛЕ) *2I* (*U19*) встановлюються при виготовленні МПП, в залежності від організації пам'яті ГК. У випадку, коли на платі встановлено МК пам'яті даних *U23* та пам'яті програм *U24* (Гарвардська архітектура), то на плату встановлюють резистори *R1, R2* та *R4*. В тому випадку, коли встановлено лише МК пам'яті даних *U23*, в якій в різних адресних областях зберігається як ПЗ так і дані (архітектура фон-Неймана), то на плату встановлюють резистор *R3* та ЛЕ *U19*. ЛЕ *U21* та *U22* керують режимами доступу ГК до пам'яті програм *U24*, в залежності від його режиму роботи.

Виділяються два режими роботи головного контролера: 1) перепрограмуван-

ня; 2) виконання прикладної програми. В режимі перепрограмування, ГК виконує програму завантажувача, яка знаходиться у його внутрішньому ПЗП, а *U24* розглядається як зовнішній оперативний запам'ятовуючий пристрій (ОЗП), в який записується програма, що буде виконуватись в подальшому. В режимі виконання прикладної програми *U24* виконує роль зовнішнього ПЗП для ГК. В цьому режимі, щоб не відбулось випадкового запису інформації в *U24*, сигналом *WRITE\_PROTECT*, що поступає від *U8* на вхід ЛЕ *U22*, блокується можливість проходження сигналу *WR* від ГК на одноім'яний вхід *U24*.

Управління режимами роботи головного контролера здійснює ІК *U8* за допомогою керуючих сигналів, які поступають на входи *RST, PSEN, EA* головного контролера. Вибір режиму задає рівень сигналу на вході *EA* ГК, при активному сигналі *RESET*. Якщо  $EA=1$ , то ГК переходить в режим перепрограмування, виконуючи програму завантажувача, що розміщена в його внутрішньому ПЗП. В цьому режимі ІК *U8* отримує від сервера ВКС програму роботи ГК, яка через *UART* інтерфейс передається ГК. Останній записує програму, яку він в майбутньому буде виконувати, в МК *U24*.

Після того, як процес перепрограмування завершився, ІК U8 повторно формує сигнал *RESET* і одночасно встановлює на вході EA рівень лог.0. Після завершення сигналу *RESET* ГК переходить виконання прикладної програми, яка раніше була завантажена в ПЗП (U24).

Загальний вигляд розробленого МПП представлено на рис. 4.

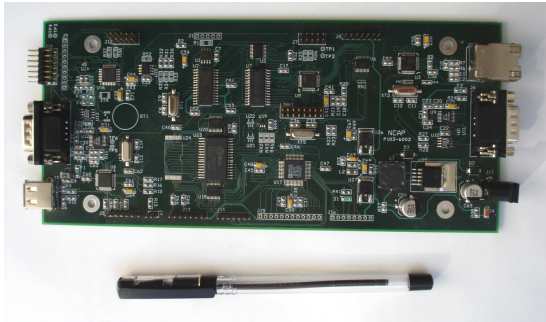


Рис. 4. Зовнішній вигляд розробленого МПП

### Висновки

В роботі представлено двопроцесорний МПП із роздільною реалізацією процесів обміну та обробки даних. Запропонований МПП, на відміну від однопроцесорного, має ряд переваг:

1) зменшені вимоги до продуктивності МК, оскільки кожен із МК спеціалізується на виконанні задач певного характеру;

2) спрощена структура ПЗ;

3) можливість дистанційного перепрограмування в режимі *on-line*, без використання ОСРЧ;

4) можливість нарощування продуктивності МПП, шляхом створення множини вузлів ГК із різними функціональними можливостями, які можуть підключатись до внутрішньосистемного інтерфейсу I2C.

Вище означені переваги реалізованого МПП вказують на можливість створення на базі МК серії не дорогих МПП із функціонально-вартісними показниками, оптимізованими під конкретну задачу.

### Список літератури

1. Кругляк К. Промышленные сети: цели и средства // Современные техноло-

гии автоматизации. – 2002, № 4. – С. 6–17.

2. Киселев В. Промышленный Ethernet в стиле Hirschman / Современные технологии автоматизации, №2, 2005. – С. 6–12.

3. IEEE Std 1451.1-1999 Standard for a Smart Transducer Interface for Sensors and Actuators - Network Capable Application Processor (NCAP) Information Model // Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey 08855, 25 June, 1999.

4. IEEE Std 1451.2-1997 Standard for a Smart Transducer Interface for Sensors and Actuators. Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats // Institute of Electrical and Electronics Engineers, Inc., Piscataway, New Jersey 08855, 26 September, 1997.

5. T. Brooks, S. Chen, K. Lee IEEE 1451 Smart Wireless Machinery Monitoring and Control for Naval Vessels // Proceedings of the Thirteenth International Ship Control Systems Symposium, 7-9 April 2003, Orlando, Florida, USA. – P. 255–259.

6. Перевалов А. Время думающих контроллеров уже наступило // CHIPnews-Україна, 2005. – №8 (48). – С. 52–56.

7. D. Wobschall An Implementation of IEEE 1451 NCAP for Internet Access of Serial Port-Based Sensors // Proceedings of second Sensor for Industry Conference (SIcon'02), 19-21 November 2002, Houston, Texas, USA. – P. 157–160.

8. R. Kochan, K. Lee, V. Kochan, A. Sachenko Development of a Dynamically Reprogrammable NCAP // Proceedings of the IEEE Instrumentation and Measurement Technology Conference (IMTC'2004), 18-20 May 2004. – P. 1188–1193.

9. Maykiv, A. Stepanenko, D. Wobshal, V. Kochan, R. Kochan, A. Sachenko Remote Reprogrammable NCAPs: Issues and Approaches // Proceedings of fourth IEEE international workshop on Intelligent Data Acquisition and Advancing Computing Systems (IDAACS'2007), 6-8 September, 2007, Dortmund, Germany. – P. 109–113.