

БАГАТОРІВНЕВА СИСТЕМА УПРАВЛІННЯ ТЕХНОЛОГІЧНИМИ ПРОЦЕСАМИ

Розроблено структуру системи управління технологічними процесами з такими рівнями управління: збір даних та управління виконавчими механізмами; контроль та управління технологічними процесами; операторський контроль та формування управлінських рішень. Сформульовано вимоги до компонентів системи, вибрано методологію синтезу системи, яка ґрунтується на компонентно-орієнтованій технології та розглянуто основні апаратно-програмні компоненти багаторівневих систем управління технологічними процесами.

Ключові слова: багаторівневі системи управління, інтеграція систем, компонентно-орієнтована технологія, технологічні процеси, програмовані логічні контролери.

IVAN TSMOTS, SERGIY STRJAMETS AND DMYTRO ZERBINO

National University "Lviv Polytechnic"

THE MULTILEVEL SYSTEM FOR TECHNOLOGICAL PROCESSES CONTROL

The structure of the control system for intelligent manufacturing has been discussed, which has the following management levels: measurement data collection and control actuators level, self control and process control level, operator control and form-making level. The requirements to the system components have been formulated. The methodology of system's synthesis, based on component-oriented technology has been proposed. The basic hardware and software components of multi-process control systems have been chosen.

Keywords: Multi-management, systems integration, component-oriented technology, intelligent manufacturing, programmable logic controllers.

Постановка проблеми

Основними задачами сучасних багаторівневих систем управління (БСУТП) технологічними процесами є інтеграція функцій управління, створення єдиного інформаційного простору з достовірною, повною та оперативною інформацією. Інтеграція в БСУТП визначається як спосіб організації окремих компонентів в одну систему, яка забезпечує узгоджену і цілеспрямовану їх спільну взаємодію, що зумовлює високу ефективність функціонування всієї системи. Інтеграція в БСУТП здійснюється в таких напрямках: функціональному, організаційному, інформаційному, програмно-алгоритмічному, технічному та економічному.

Функціональна інтеграція забезпечує єдність цілей та узгодженість критеріїв і процедур виконання технологічних функцій, пов'язаних із досягненням поставлених цілей. Основою функціональної інтеграції є: оптимізація функціональної структури всієї системи, декомпозиція системи на локальні підсистеми, формалізований опис функцій кожної підсистеми і протоколів взаємодії підсистем.

Організаційна інтеграція передбачає формування узгоджених управлінських рішень шляхом раціональної взаємодії управлінського персоналу на різних рівнях системи.

Інформаційна інтеграція полягає в комплексному підході до створення єдиного інформаційного поля на основі об'єднання технологічного процесу збору, зберігання, передачі та опрацювання даних. Інформаційне забезпечення БСУ містить такі компоненти: систему класифікації та кодування, систему документації та інформаційну базу БСУТП, яка є розподіленою багаторівневою системою взаємопов'язаних інформаційних баз.

Програмно-алгоритмічна інтеграція передбачає присутність взаємопов'язаних комплексів моделей, алгоритмів, операційних систем, прикладних програм і їх спільне використання на всіх рівнях системи.

Технічна інтеграція полягає у використанні єдиного комплексу спільних апаратних засобів для управління агрегатами, апаратами, установками та виконавчими механізмами шляхом збору та опрацювання даних на всіх рівнях БСУТП.

Економічна інтеграція визначає стратегію витрат коштів на відновлення чи заміну компонентів системи, а також на її розвиток та модифікацію в цілому. Ця стратегія формулюється виходячи з функціонально-вартісного аналізу системи та з довгострокового прогнозу розвитку всієї галузі.

Синтез БСУТП технологічними процесами зводиться до рішення задачі інтеграції існуючих готових апаратних і програмних засобів, оскільки розробка та виготовлення нових вимагає значних коштів і часу. При розв'язанні задачі інтеграції необхідно враховувати багато факторів, а саме: інформацію про компоненти, які є на ринку, технічні характеристики готових компонентів, відповідність інтерфейсів компонентів стандартам, можливість їх покупки та інше.

Ефективна робота багаторівневих систем управління технологічними процесами вимагає опрацювання великих обсягів різнотипної інформації. Зменшення обсягів інформації досягається наближенням засобів опрацювання до джерел надходження інформації (давачів) та виконавчих механізмів шляхом використання розподілених мікроконтролерних систем.

У зв'язку з цим особливої актуальності набуває проблема розробки та синтезу апаратно-програмних компонент високоефективних БСУТП з використанням сучасної економічно обґрунтованої елементної бази.

Аналіз останніх досліджень та публікацій

За останні роки було проведено багато досліджень, які опубліковані в працях, що так чи інакше стосуються питань розробки та синтезу апаратно-програмних компонент БСУТП [1–28]. У роботах [1–5] подано широкий та збалансований огляд технічних та інженерних аспектів розроблення багаторівневих

систем управління, розглянуто сучасні технології автоматизації і показано доцільність їх використання для створення сучасних автоматизованих систем управління технологічними процесами. Також там детально розглянуті промислові системи контролю, сенсори, виконавчі механізми, засоби збереження та автоматичного збору інформації та інші компоненти управління технологічними процесами. У [1–4] розглянуто використання розподілених мікроконтролерних систем для управління технологічними процесами, та пов'язані з цим питання організації взаємодії між рівнями управління в автоматизованих системах управління технологічними процесами. Розробка компонент для виконання збору даних, а також, питання управління виконавчими механізмами детально розглянуті [6–11]. З аналізу публікацій [6, 12] випливає, що для формування ефективних управлінських рішень у БСУТП необхідно розробити засоби для оцінювання протікання технологічних процесів та оперативного, аналітичного та інтелектуального опрацювання інформації та прогнозування.

Проте у розглянутих працях недостатньо уваги приділено питанням комплексного підходу до інтелектуалізації опрацювання даних, розроблення програмно-апаратних засобів та їх адаптації до задач управління на всіх рівнях БСУТП.

Завдання і мета дослідження

Метою статті є розроблення структури багаторівневої системи управління технологічними процесами, формулювання вимог до компонентів, вибір методології синтезу системи та розгляд основних апаратно-програмних компонент.

Виклад основного матеріалу

Структура багаторівневої системи управління технологічними процесами. Основними задачами сучасних БСУТП є дистанційне налаштування, контроль, управління виконавчими механізмами та процесами в цілому. Виконання таких задач передбачає збір даних, створення єдиного інформаційного простору з достовірною, повною та оперативною інформацією, опрацювання даних, формування сигналів управління виконавчими механізмами та управлінських рішень для операторів. В такій системі кожний рівень виконує свої задачі, які зв'язані з підвищенням ефективності системи в цілому. Особливістю роботи БСУТП повинна бути: висока гнучкість налаштування, можливість дистанційного контролю, а також використання компонентно-орієнтованих протоколів для взаємодії між компонентами системи.

Апаратно-програмні компоненти БСУТП повинні забезпечувати: збір даних; створення єдиного інформаційного простору з достовірною, повною та оперативною інформацією; опрацювання даних; формування сигналів управління та коректних управлінських рішень для операторів; контроль виконавчих механізмів. Для забезпечення перерахованих вимог розроблена базова структура БСУТП, яка наведена на рис. 1.



Рис. 1. Базова структура БСУТП

Розроблена базова структура БСУТП складається з трьох рівнів:

- збору даних та управління виконавчими механізмами;
- контролю та управління технологічними процесами;
- операторського контролю та формування управлінських рішень.

Специфіка кожного рівня системи управління технологічними процесами визначається апаратно-програмними компонентами [8]. На кожному ієрархічному рівні управління розв'язуються задачі відповідного рівня складності.

Рівень збору даних та управління виконавчими механізмами. На цьому рівні формується первинна

інформація, яка може попередньо опрацюватися, накопичуватися та надходити на засоби контролю. З використанням даної інформації формуються сигнали для управління виконавчими механізмами та технологічними процесами. Для попереднього опрацювання даних із давачів в умовах заводів і неповної інформації використовуються мікроконтролери сімейства STM8.

Рівень контролю та управління технологічним процесом. Цей рівень на управління передбачається як достатньо автономний, який при відсутності зв'язку з верхнім рівнем здатний тривалий час без втрати інформації працювати автономно. На даному рівні використовуються мікроконтролерні системи RaspberryPi та Arduino, програмовані логічні контролери Mitsubishi Melsec FX3U, засоби візуального контролю та управління технологічним процесом. Мікроконтролерні системи використовуються як для оперативного опрацювання даних, прогнозування поведінки процесів і об'єктів управління, так і для інтелектуалізації роботи компонентів системи управління. Опрацювання даних на даному рівні зменшує обсяги даних, які необхідно передавати на верхній рівень і тим самим забезпечує зниження вимог до пропускну здатності каналів зв'язку.

На даному рівні висуваються жорсткі вимоги до апаратно-програмних засобів за надійністю, часом реакції на дані, що надходять від давачів, на час формування управляючих сигналів для виконавчих механізмів. Загалом апаратно-програмні засоби повинні працювати у реальному часі, тобто гарантовано відкликатись на зовнішні події за час, визначений для кожної такої події.

Рівень операторського контролю та формування управлінських рішень. На цьому рівні операторський контроль представлений автоматизованим робочим місцем оператора. Апаратні засоби, які можуть використовуватись на цьому рівні, визначається інтенсивністю надходження даних, складністю алгоритмів опрацювання та вимогами за надійністю, які висуваються до системи управління. В якості апаратних засобів можуть використовуватись робочі станції оператора на RISC- або Intel-платформах. Задачі цього рівня такі: збір даних з програмованих логічних контролерів і мікроконтролерних систем; збереження даних; опрацювання даних; опрацювання відео-потоків, розпізнавання зображень і сцен в системах технічного зору; формування управлінських рішень; синхронізація єдиного часу в розподіленій системі; синхронізація роботи розподілених підсистем; візуалізація та відображення ходу виконання технологічного процесу.

Для розв'язання даних задач використовується системи SCADA, основною функцією якої є створення інтерфейсу оператора та збір даних про технологічний процес. Для контролю та управління технологічним процесом використовується інструментальне забезпечення, яке є сукупністю апаратних засобів, каналів зв'язку та алгоритмічно-програмних засобів.

Особливостями задач, які розв'язуються на рівні операторського контролю та формування управлінських рішень є:

- великий обсяг і різноманітність даних;
- суперечливість та неповнота даних;
- постійність і висока інтенсивність надходження вхідних даних;
- великий обсяг обчислень з переважанням обчислювальних операцій над логічними при опрацюванні відео-потоків, розпізнавання зображень і сцен в системах технічного зору;
- постійне ускладнення алгоритмів опрацювання та підвищення вимог до точності результатів;
- можливість розпаралелення опрацювання даних як у часі, так і у просторі.

Вимоги до компонентів багаторівневої системи управління. Однією з найбільш широко розповсюджених вимог, що ставиться до апаратно-програмних компонентів багаторівневої систем управління є забезпечення високої швидкодії.

Подібна проблема виникає, як правило, при використанні компонентів для розв'язання задач у реальному часі, який накладає певні обмеження на процес обробки інформації [8]. Для забезпечення опрацювання потоків даних у реальному часі продуктивність (тобто, кількість операцій на секунду) апаратно-програмних компонентів повинна бути:

$$П \geq \frac{R\beta F_d}{n_d},$$

де R – складність алгоритму розв'язання задачі (кількість операцій); $\beta > 1$ – коефіцієнт врахування особливостей засобів реалізації алгоритму; F_d – частота надходження даних (кількість разів виконання алгоритму за секунду); n_d – кількість компонент розпаралелення.

Використання апаратно-програмних компонентів на нижніх рівнях управління, де апаратура розміщена безпосередньо біля давачів і виконавчих механізмів, накладає жорсткі обмеження на їхні масогабаритні характеристики. Одночасно до таких компонентів висуваються жорсткі вимоги за потужністю енергоспоживання, яка впливає на габарити джерел живлення та засобів відведення тепла. Забезпечити такі вимоги можна шляхом зменшення довжини розрядної сітки, використовувати фіксовану кому для представлення операндів, скорочення переліку команд і числа ліній адресної шини, що визначають доступну користувачу ємність пам'яті.

Окрім того, апаратно-програмні компоненти БСУП мають забезпечувати високу живучість, надійність, перевірку працездатності та швидку локалізацію несправності. Проблема високої живучості апаратно-програмних компонентів виникає при використанні його в системах управління особливо важливими об'єктами, розміщеними на великій відстані від людини або об'єктами, що мають великий зовнішній вплив. Щоб забезпечити високу живучість компонентів, необхідна взаємозаміна його структурних частин.

Зменшення масогабаритних характеристик, енергоспоживання, підвищення надійності компонентів

та забезпечення режиму реального часу може бути досягнуто шляхом використання сучасних інтегральних технологій та надвеликих інтегральних схем (НВІС).

Компонентно-орієнтована технологія. На сучасному етапі розвитку інформаційних технологій розробка БСУТП зводиться до рішення задачі інтеграції існуючих готових апаратних і програмних засобів, оскільки розробка та виготовлення нових вимагає значних коштів і часу. При розв'язанні задачі інтеграції необхідно враховувати множину факторів, а саме: інформацію про готові апаратно-програмні компоненти, їх технічні характеристики, відповідність інтерфейсів стандартам, можливості їх покупки та інше.

В основу проектування сучасних БСУТП покладена системна інтеграція, яка ґрунтується на системному підході, який охоплює всі рівні інтеграції процесів, об'єктів, суб'єктів та інфраструктури з врахуванням ефективності їх застосування та вимог конкретного застосування.

Системну інтеграцію пропонується здійснювати на основі компонентно-орієнтованої технології, яка передбачає поділ процесу розробки на ієрархічні рівні та види забезпечення (алгоритмічне, апаратне та програмне). Для реалізації даної технології використовується метод декомпозиції, що передбачає розбиття багаторівневих систем управління на рівні та апаратно-програмні компоненти. На кожному рівні ієрархії використовуються апаратно-програмні компоненти для розв'язання задач відповідної складності, що характеризуються як одиницями інформації, так і алгоритмами обробки.

Збільшення номера рівня ієрархії відповідає збільшенню деталізації алгоритмічних, апаратних і програмних компонентів. При цьому на вищих рівнях ієрархії одиниці інформації, алгоритми, програмні та апаратні засоби представляють собою впорядковані сукупності одиниць інформації та композиції алгоритмів, програмних і апаратних засобів нижчих рівнів ієрархії. Методологія послідовної декомпозиції, яка використовується при розробці БСУТП, відображає процес розробки «зверху вниз». Процес синтезу БСУТП з використанням компонентно-орієнтованої технології описується за допомогою такого виразу:

$$C_{БСУТП}^1 = \bigcup_{i=1}^n C_{БСУТП}^{2i} \bigcup_{j=1}^m C_{БСУТП}^{3j},$$

де $C_{БСУТП}^{2i}$, $C_{БСУТП}^{3j}$ – відповідно апаратно-програмні компоненти другого та третього рівнів управління; n – кількість апаратно-програмних компонентів другого рівня; m – кількість апаратно-програмних компонентів третього рівня.

При проектуванні багаторівневих систем управління використовується як горизонтальну, так і вертикальну інтеграцію. Горизонтальна інтеграція передбачає об'єднання багаторівневих систем управління на одному рівні ієрархії, а вертикальна – об'єднує засоби управління сусідніх рівнів ієрархії.

Апаратні засоби БСУТП. Більшість апаратних засобів, які використовуються в БСУТП є вбудованими комп'ютерними системами, концепція розробки яких полягає в тому, що такі системи працюють, будучи вбудованими безпосередньо у пристрої, якими вони управляють. Особливістю таких систем є оптимізація їх архітектури для виконання визначених функцій у реальному часі. Основною елементною базою, яка використовується для реалізації таких систем є програмовані логічні контролери (ПЛК). Архітектура сучасних ПЛК орієнтована на обробку сигналів, контроль стану пристроїв і управління роботою виконавчих механізмів [9]. Основна задача ПЛК – це збір даних шляхом опитування вимикачів, давачів і в залежності від їх стану формувати управляючі сигнали для виконавчих механізмів. Сучасні ПЛК мають широкі можливості для програмування і внесення змін в уже існуюче програмне забезпечення. З використанням ПЛК розробляється для БСУТП апаратно-програмний компонент у вигляді модуля, який об'єднує реле, таймери, лічильники та перетворювачі. Можливість перепрограмування такої апаратно-програмної компоненти збільшила гнучкість і забезпечила швидко адаптацію БСУТП до змін. Апаратно-програмні компоненти, що реалізуються на базі ПЛК мають можливість дистанційного налаштування та управління, що значно розширили сферу їх застосування [9].

На рівні операторського контролю та формування управлінських рішень для розроблення апаратно-програмних компонентів використовуються промислові комп'ютери такі як Mitsubishi Electric, BECKHOFF, Eaton, АХІОМТЕК, які відповідають промисловим умовам експлуатації з підвищеними вимогами до міцності та надійності. Апаратно-програмні компоненти, які використовуються для збору даних, контролю та управління виконавчими механізмами і технологічними процесами, реалізуються з використанням ПЛК сімейства STM, Modicon, LOGIC, MAXIM. Крім того як апаратні компоненти у БСУТП широко використовуються програмовані реле ZelioLogic, ZEN та Easy, які за своїми функціональними можливостями та габаритами повністю замінюють старі реле.

Мови програмування мікроконтролерних систем.

За останнє десятиліття з'явилося кілька технологічних мов програмування ПЛК, для яких розроблений стандарт МЕК 61131. Даний стандарт регламентує зміни в галузі мов програмування для систем управління технологічними процесами. Цей стандарт охоплює вимоги до апаратних засобів, монтажу, тестування, документації, зв'язку і програмування ПЛК. Стандарт визначає 5 мов програмування, які діляться на текстові та графічні. До текстових мов ПЛК відносяться:

- Instruction List (IL) – мова інструкцій;
- Structured Text (ST) – мова структурованого тексту.

Мова IL – типовий асемблер з акумулятором та переходами за позначками. Набір інструкцій стандартизовано, він не залежить від конкретної цільової платформи. Дана мова забезпечує роботу з будь-якими типами даних, викликами функцій та функціональні блоки, реалізовані іншими мовами. За допомогою IL реалізуються алгоритми будь-якої складності.

Мова ST – це мова високого рівня, яка синтаксично подібна до мови Паскаль. Замість процедур мови Паскаль в ST використовуються компоненти програм стандарту МЕК.

До графічних мов ПЛК відносяться:

- Sequential Function Chart (SFC) – мова послідовних функціональних діаграм;
- Function Block Diagram (FBD) – мова функціонально-блокових діаграм;
- Ladder Diagrams (LD) – мова релейно-контактних схем;

Мова SFC – високорівневий графічний інструмент, у якій графічна діаграма складається із кроків і переходів між ними. Дозвіл переходу визначається умовою, а з кроком пов'язані певні дії.

Мова FBD – орієнтована на опис принципів схем електронного пристрою на мікросхемах. Провідники в FBD використовуються для передачі сигналів будь-якого типу (логічний, аналоговий, час тощо). Сигнали з виходів блоків можуть подаватися на входи інших блоків або безпосередньо на виходи ПЛК. Самі блоки, подані на схемі як «чорні ящики» можуть виконувати будь-які функції. За допомогою мови FBD забезпечується опис взаємозв'язку входів і виходів діаграми. Якщо алгоритм добре описується з позиції сигналів, то його FBD-подання завжди виходить наочніше, ніж у текстових мовах.

Мова LD – орієнтована на реалізацію структури електричних ланцюгів. Графічно LD-діаграма подається у вигляді двох вертикальних шин живлення. Між ними розмішені ланцюги, утворені з'єднанням контактів. Навантаженням кожному ланцюгу служить реле. Кожне реле має контакти, які можна використати в інших ланцюгах. Послідовне (I), паралельне (АБО) з'єднання контактів та інверсія (НЕ) утворюють базис Буля. У результаті мова LD ідеально підходить не тільки для побудови релейних автоматів, але й для програмної реалізації комбінаційних логічних схем. Завдяки можливості використання в LD функцій і функціональних блоків, виконаних іншими мовами, галузь її застосування збільшується.

Мови із стандарту МЕК 61131 розроблені на основі найпопулярніших мов програмування для сучасних контролерів. Програми, які написані для сучасних контролерів, можна перенести в середовище МЕК 61131-3 із мінімальними затратами. Особливістю стандарту МЕК 61131 є можливість створення апаратно-незалежних бібліотек для реалізації регуляторів, фільтрів, керування приводами, модулі з нечіткою логікою тощо.

Для програмування контролерів мовами стандарту МЕК 61131-3 розроблено середовище програмування CoDeSys (Controllers Development System), редактори та налагоджувальні засоби якого ґрунтуються на принципах популярних середовищах професійного програмування (VisualC++ тощо). Особливістю середовища програмування CoDeSys є те, воно не прив'язане до конкретної апаратної платформи. Під різні ПЛК здійснено модифікацію середовища програмування шляхом адаптації програми до низькорівневих ресурсів – розподілу пам'яті, інтерфейсів зв'язку та інтерфейсів введення-виведення. Середовища CoDeSys забезпечує: пряму генерацію машинного коду; повноцінну реалізацію мов стандарту МЕК 61131-3; інтелектуальні редактори мов, які виправляють типові для початківців помилки; вбудований емулятор контролера, що забезпечує налагодження проекту без додаткових апаратних засобів; вбудовані елементи візуалізації забезпечують створення моделі об'єкту управління та налагодження проекту без виготовлення засобів імітації; використання готових бібліотек і сервісних функцій.

Технологія Device Hive. Для об'єднання між собою провідними та безпровідними каналами та під'єднання до мережі Інтернет давачів, мікроконтролерних систем і виконавчих механізмів у БСУТП використовується технологія Device Hive. Дана технологія гнучка, масштабована та проста в застосуванні, яка забезпечує обмін даними між апаратними компонентами БСУТП за принципом М2М. Застосування технології Device Hive забезпечує формування комунікаційного середовища, програмний контроль і використання мультитиплатформових бібліотек для розроблення засобів віддаленого управління та моніторингу, телеметрії, дистанційного управління та контролю. З технологією Device Hive можна працювати використовуючи широкий спектр технологій, наприклад “embedded Linux”, Python, бібліотеки C++, протокол JSON, або підключати AVR, Microchip мікроконтролери. Особливістю роботи з Device Hive є організація в першу доступу до мережі, а потім програмування конкретних застосувань, що зменшує час на проектування способів передачі даних.

Мікроконтролерні системи. Програмовані контролери Mitsubishi FX3U є найпотужнішими і високопродуктивними в лінійці контролерів MELSECFX (FX1S, FX3S, FX1N, FX3G). Архітектура контролера є двоштинною, що збільшує його можливості. Програмований контролер Mitsubishi забезпечує підключення як розширювальних модулів попереднього покоління, так і модулів нового покоління серії FX3U. При підключенні модулів FX3U контролер автоматично перемикає свою комутаційну шину на високошвидкісний режим і обмін даними відбувається на підвищеній швидкості. Модулі FX0N, FX2N працюють з контролером на звичайних швидкостях. Таким чином, використання розширювальних модулів забезпечує збільшення кількості входів/виходів до 256 (при прямій адресації), а через станції децентралізованого введення/виведення до 384.

Крім того, програмований контролер Mitsubishi забезпечує підключення високошвидкісних адаптерних модулів FX3U-XXX-ADP, які розширюють можливості контролера при роботі з аналоговими сигналами та збільшують кількість додаткових комунікаційних інтерфейсів (RS232 / 422/485) [11].

Одноплатний мікрокомп'ютер Raspberry Pi є перспективною платформою для систем автоматизації, який характеризується низькою ціною, своєю операційною системою та відкритістю архітектури. Мікрокомп'ютер Raspberry Pi побудований на системі (SoC) Broadcom BCM2835, яка включає в себе процесор ARM із тактовою частотою 700 МГц, графічний процесор VideoCore IV, і 512 або 256 мегабайт оперативної пам'яті, в якості додаткової пам'яті використовується SD картка [12]. Даний

мікрокомп'ютер випускається у двох версіях: молодша (А) (256 Мб ОЗП, один USB-порт) і старша (В) (з Ethernet, 512 Мб ОЗП, два USB-порти)

Для Raspberry Pi випущені спеціалізований дистрибутив Linux, Raspbian OS (він заснований на дистрибутиві Debian) і ряд застосунків Pi-Store, де є як платні, так і безплатні програми. Raspberry Pi має власні порти GPIO, які можна задіяти під різні функції.

Мікроконтролери сімейства STM8 – це відносно нове сімейство мікроконтролерів від компанії ST Microelectronics, в якому втілено мало не все, чого можна очікувати від 8 бітових програмованих контролерів. До переваг контролерів STM8 слід віднести в першу чергу низьку вартість, причому не тільки самого контролера, а і відлагоджувальних плат, програматорів та відладчиків. Вбудований bootloader дозволяє завантажувати програми з UART, SPI, CAN, або I2C. Надається безкоштовна бібліотека драйверів периферії. В компанії STM розробили бібліотеку модулів STM8x Firmware Library, яка є фреймворком для мікроконтролерів. Взагалі, у всьому ряді STM8 простежується підвищені вимоги до електромагнітної сумісності та надійності, що також є привабливим для їх використання [14].

До недоліків слід віднести відсутність контролерів з USB і в маловивідних корпусах (типу SO8). Наразі існує три сімейства STM8:

- STM8S - "standard" контролери загального застосування, зазвичай 10 бітна аналогова периферія, середнє за сучасними стандартами енергоспоживання та діапазон живлення – 2.95–5.5В.

- STM8L - "low-power" контролери з низьким споживанням, 12 бітова аналогова периферія, покращена електромагнітна сумісність і діапазон живлення – 1.8–3.6В. У порівнянні зі стандартними контролерами, в цьому сімействі додається деяка кількість периферії, зокрема, DMA.

- STM8A - "automobile" – сімейство зосереджено на безпеці і надійності, витримують більше навантажень на ніжки, діапазон живлення – 2.95–5.5В і працюють при 145 градусах [15].

STM8 має 8-бітове CISC ядро і містить 6 регістрів: А – *Accumulator* – однобайтний акумулятор, який містить результат виконання будь-якої арифметико-логічної команди; X, Y – *Index* – двобайтні індексні регістри, які можуть містити відносну адресу комірок пам'яті з якими можна виконувати операції; PC – *Program counter* – трибайтний програмний лічильник, який містить абсолютну адресу комірки пам'яті, що містить команду, яка на даний момент виконується; SP – *Stack Pointer* – двобайтний вказівник верхівки стеку, який має свою пам'ять; CC – *Condition Codes* – однобайтний регістр прапорців (Flags), що містить бітові ознаки результату арифметико-логічної команди, які можна використовувати для розгалужень в програмі [8].

Мікроконтролер STM8 має 32 вектори переривань, кожен з яких являє собою 4-х байтну (старший байт містить зарезервоване значення 82h) адресу початку підпрограми-обробника відповідної події, що спричинила переривання. Вихід з обробника переривань здійснюється командою IRET.

У мікроконтролері STM8 частоту тактових імпульсів можна змінювати в процесі роботи, на відміну від AVR, де тактування жорстко прошивається. У застосунках з низьким споживанням це інколи застосовується, при цьому, при збої генератора, контролер перемикається на RC і переходить в режим переривання, в результаті система не залишиться без управління.

Пам'ять мікроконтролера умовно розділена на секції по 64К, а кожна секція – на 256 сторінок по 256 байт кожна. Мікроконтролер STM8 дозволяє наступні способи адресації: безпосередню, індексну, пряму, стекову, непряму, бітову та відносну. В залежності від довжини операнду режим адресації ділиться на короткий – 1 байт, довгий – 2 байти та розширений – 3 байти.

З практичної точки зору, в першу чергу слід звернути увагу на єдиний 24-бітовий адресний простір, в якому розміщено Flash-пам'ять, ОЗУ, Еeprom, а також регістри периферії. Це істотно спрощує написання коду, наприклад, функції для роботи з масивами з ОЗУ і Flash-пам'яті не потрібно писати в декількох примірниках. Коди різних стеків і бібліотек легко перетворюються, так як в основному розраховані на архітектуру фон Неймана (в сенсі адресного простору). При цьому шини для доступу до різних типів пам'ятей розділені, що вказує на наявність Гарвардської архітектури [16].

Висновки

1. Проектування сучасних багаторівневих систем управління технологічними процесами здійснюється на основі системної інтеграції, яка ґрунтується на системному підході, що охоплює всі рівні інтеграції процесів, об'єктів, суб'єктів та інфраструктури з врахуванням ефективності їх застосування та вимог конкретного застосування.

2. Системну інтеграцію пропонується здійснювати на основі компонентно-орієнтованої технології, яка передбачає поділ процесу розробки на ієрархічні рівні та види забезпечення: алгоритмічне, апаратне та програмне.

3. Розроблення швидкодіючих, малогабаритних та з малою потужністю енергоспоживання апаратних компонент багаторівневої систем управління здійснюється на базі мікроконтролерів з фіксованою комою, невеликою розрядною сіткою, усіченим інтерфейсом і скороченою системою команд.

4. Для програмування мікроконтролерів доцільно використовувати середовище програмування CoDeSys (Controllers Development System), редактори та налагоджувальні засоби якого ґрунтуються на принципах середовищ професійного програмування (VisualC++ тощо).

5. У сучасних багаторівневих систем управління технологічними процесами для об'єднання давачів, мікроконтролерних систем і виконавчих механізмів між собою провідними та безпроводними каналами та під'єднання їх до мережі Інтернет доцільно використовувати технологію Device Hive.

Література

1. Mikell P. Groover. Automation, production systems, and computer-integrated manufacturing / 4th Edition by G. Jayaprakash. – Pearson Education Limited, 2016. – 798 p. – ISBN 10: 1-292-07611-9.
2. James A. Rehg. Computer-Integrated Manufacturing / James A. Rehg, Henry W. Kraebber. – 3rd Edition. – Prentice Hall, 2012. – 592 p. – ISBN 10: 0131134132.
3. Paulo Leitao. Agent-based distributed manufacturing control: A state-of-the-art survey / Engineering Applications of Artificial Intelligence. – Volume 22, Issue 7 (2009). – 979–991.
4. Herve Panetto. Enterprise integration and interoperability in manufacturing systems: Trends and issues / Herve Panetto, Arturo Molina. – Computers in industry Volume 59, Issue 7 (2008). – 641–646.
5. Artificial neural networks for intelligent manufacturing / Edited by Cihan H. Dagli. – Intelligent Manufacturing Series, Springer Science & Business Media, 2012. – ISBN 978-94-010-4307-6.
6. Інтелектуальні компоненти інтегрованих автоматизованих систем управління : монографія / Медиковський М.О., Ткаченко Р.О., Цмоць І.Г., Цимбал Ю.В., Дорошенко А.В., Скорохода О.В. – Львів : Видавництво Львівської політехніки, 2015.–280 с.
7. Цмоць І.Г. Визначення задач і формування вимог до інтелектуальних компонентів інтегрованих АСУ / І.Г. Цмоць, О.В. Скорохода // Технічні вісті : науково-технічний журнал. – 2014. – 2(40), 2(34). – С. 53–54.
8. Цмоць І.Г. Багатоканальний пристрій з буферизацією даних для автоматизованих систем управління технологічним процесом з промисловою мережею / І.Г. Цмоць, Б.А. Демида, Х.Г. Гульовата // Науковий вісник національного лісотехнічного університету України : збірник науково-технічних праць. – Львів : РВВ НЛТУ України. –2009. – Вип. 19.5. – С. 275–284.
9. Цмоць І.Г. Проектування багатоканального пристрою обміну даними для АСУ ТП з промисловою мережею / І.Г. Цмоць, М.Р. Подольський // Системні технології : регіональний міжвузівський збірник наукових праць. – Дніпропетровськ, 2009. – Випуск 6 (65). – С. 131–140.
10. Харазов В.Г. Интегрированные системы управления технологическими процессами / Харазов В.Г. – СПб : Профессия, 2009. – 592 с.
11. Пупена О.М. Промислові мережі та інтеграційні технології в автоматизованих системах : навчальний посібник / Пупена О.М., Ельперін І.В., Луцька Н.М., Ладанюк А.П. – К. : Вид-во «Ліра-к», 2011. – 552 с.
12. Пьявченко Т.А. Проектирование АСУТП в SCADA-системе : учебное пособие / Пьявченко Т.А. – Таганрог : Изд-во Технологического института ЮФУ, 2007. – 84 с.
13. Bjorn Solvang, Gabor Sziebig, and Peter Korondi. Multilevel Control of Flexible Manufacturing Systems / Int. Conf. on Flexible Manufacturing Systems, Krakow, Poland, May 25–27, 2008, pp. 785–790.
14. S.H. Suh, B.E. Lee, D.H. Chung, S.U. Cheon. Architecture and implementation of a shop-floor programming system for STEP-compliant CNC / Computer-Aided Design, vol. 35, Elsevier, 2002, pp. 1069–1083.
15. X.W. Wu. Realisation of STEP-NC enabled machining / Robotics and Computer-Integrated Manufacturing, Volume 22, Elsevier, 2006. – 144–153.
16. Tancheng Xie, Yuanzhou Xu, Xiang Nan, Hongru Wang, Yuanwei Xu. R&D of General Machine-Oriented Manufacturing Unit / International Conference on Mechatronics and Automation, 2006. – 457–462.
17. Nuno Lopes, Pedro Lima. OpenSDK – An Open-source Implementation of OPEN-R / Proc. of 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2008), Estoril, Portugal, 2008.
18. Noriaki Ando, Geoffrey Biggs, Tetsuo Kotoku. Robotic Component Specification / Document 12-09-01.pdf of AIST-National Institute of Advanced Industrial Science and Technology, 2012.– 22 p.
19. D. Y. Chao. Enumeration of lost states of a suboptimal control model of a well-known S3PR / IET Control Theory & Applications, Volume 5, Issue 11, (2011). – 1277–1286.
20. Шишов О.В. Современные технологии промышленной автоматизации [Электронный ресурс] : учебник / О. В. Шишов. – Саранск : Изд-во Мордов. ун-та, 2007. – 273 с. – Режим доступа : <http://chem.net/promelectr/promelectr5.php>
21. Минаев И.Г. Програмируемые логические контроллеры : практическое руководство для начинающего инженера / И.Г. Минаев, В.В. Самойленко. – Ставрополь : АГРУС, 2009. – 100 с.
22. Яковлев О.Г. Программирование ПЛК / Яковлев О.Г. – Москва, 2001.
23. Програмируемые логические контроллеры Mitsubishi Electric [Электронный ресурс]. – Режим доступа : <http://www.sovras.com/fx.php>
24. <http://www.raspberrypi.org/archives/2180>
25. <http://raspberrypi.ru/page/doc/>
26. <http://eu.mouser.com/new/stmicroelectronics/stm8s-mcus/>
27. http://www2.st.com/content/st_com/en.html
28. <http://www.compel.ru/lib/ne/2013/8/8-stm8-dlya-chaynikov-pervyye-shagi-v-srede-iar/>

Рецензія/Peer review : 4.8.2016 р. Надрукована/Printed : 25.8.2016 р.

Рецензент: д.т.н. Пелешко Д. Д.