

Лекція 2

СММП: вбудовані системи

К.т.н. Талалаєв В.О.

План лекції

1. Призначення, функції і основні типи вбудованих СММП
2. Системи управління технологічними процесами (SCADA системи).
3. Програмне забезпечення вбудованих систем.

Приклади застосування вбудованих систем



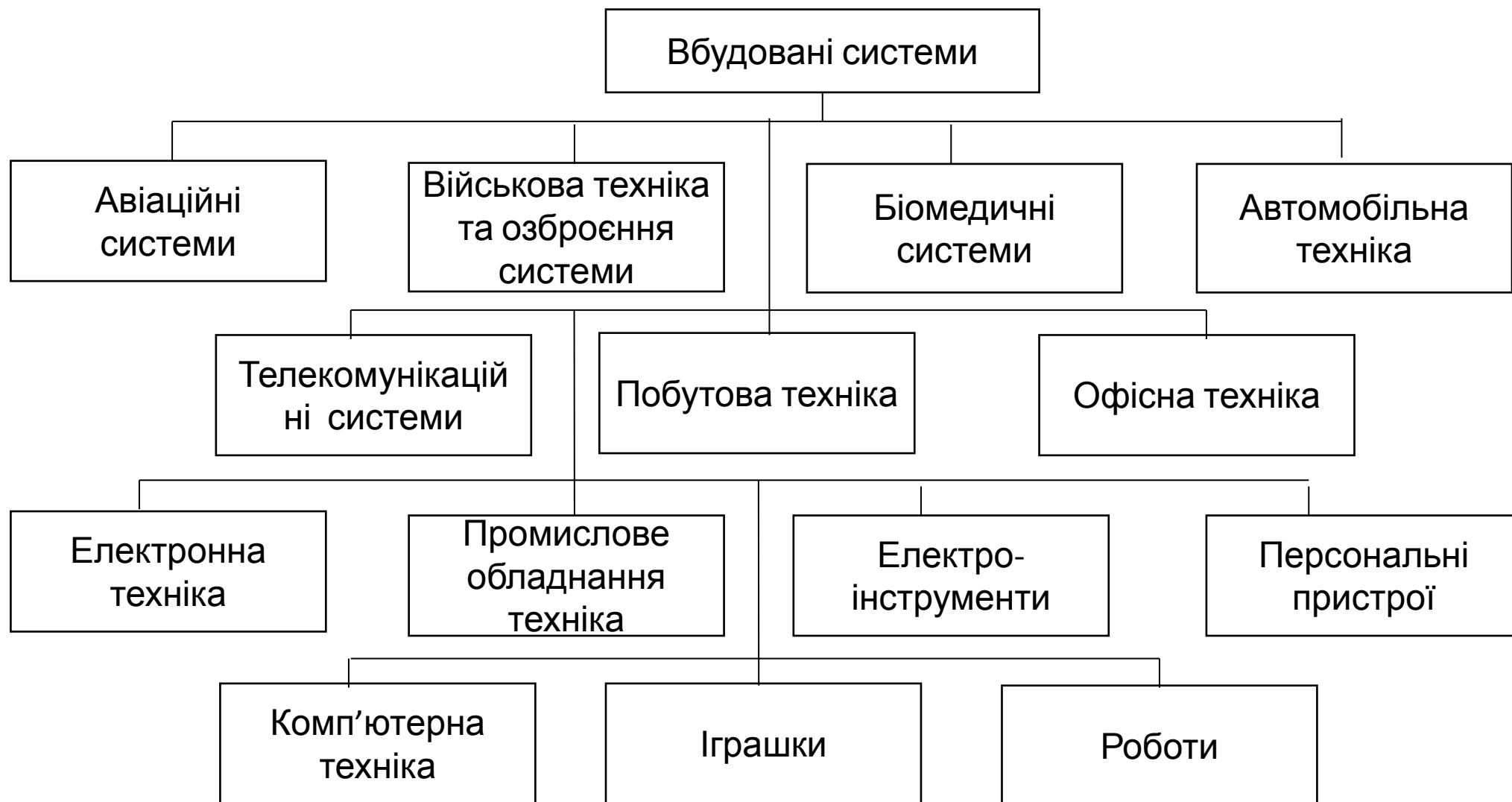
Вбудовані інформаційні системи: поняття і визначення

Вбудована інформаційна система (ВІС) – це інформаційна система, яка забезпечує управління певним об'єктами (технічні засоби, технологічні процеси) і конструктивно розміщується в межах об'єкта.

У зв'язку з тим, що ВІС розміщується всередині більш складного пристрою, при її розробці ключову роль відіграють такі чинники:

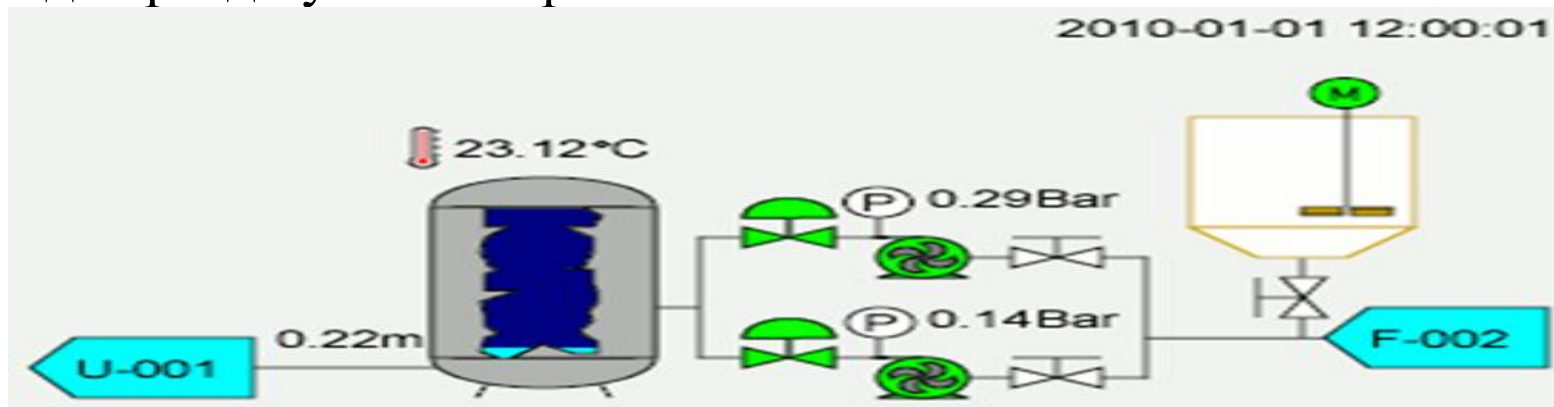
- мінімальне власне енергоспоживання (можливо, автономне живлення);
- мінімальні власні габарити і вага;
- власний захист (корпус) мінімальний і забезпечується міцністю і жорсткістю конструкції і застосованими елементами;
- функції відведення тепла (охолодження) забезпечують мінімум вимог теплових режимів. Якщо щільність теплового потоку (тепловий потік, що проходить через одиницю поверхні) не перевищує $0,5 \text{ мВт} / \text{см}^2$, перегрів поверхні пристрою щодо навколишнього середовища не перевищить $0,5 \text{ }^\circ\text{C}$, така апаратура вважається нетеплонагруженою і не вимагає спеціальних схем охолодження.
- процесор і системна логіка, а також ключові мікросхеми по можливості поєднані на одному кристалі
- спеціальні військово-космічні вимоги з радіаційної та електромагнітної стійкості, працездатність в вакуумі, гарантований час напрацювання, термін доступності рішення на ринку і т. Д.

Сфери застосування вбудованих систем



SCADA СИСТЕМИ

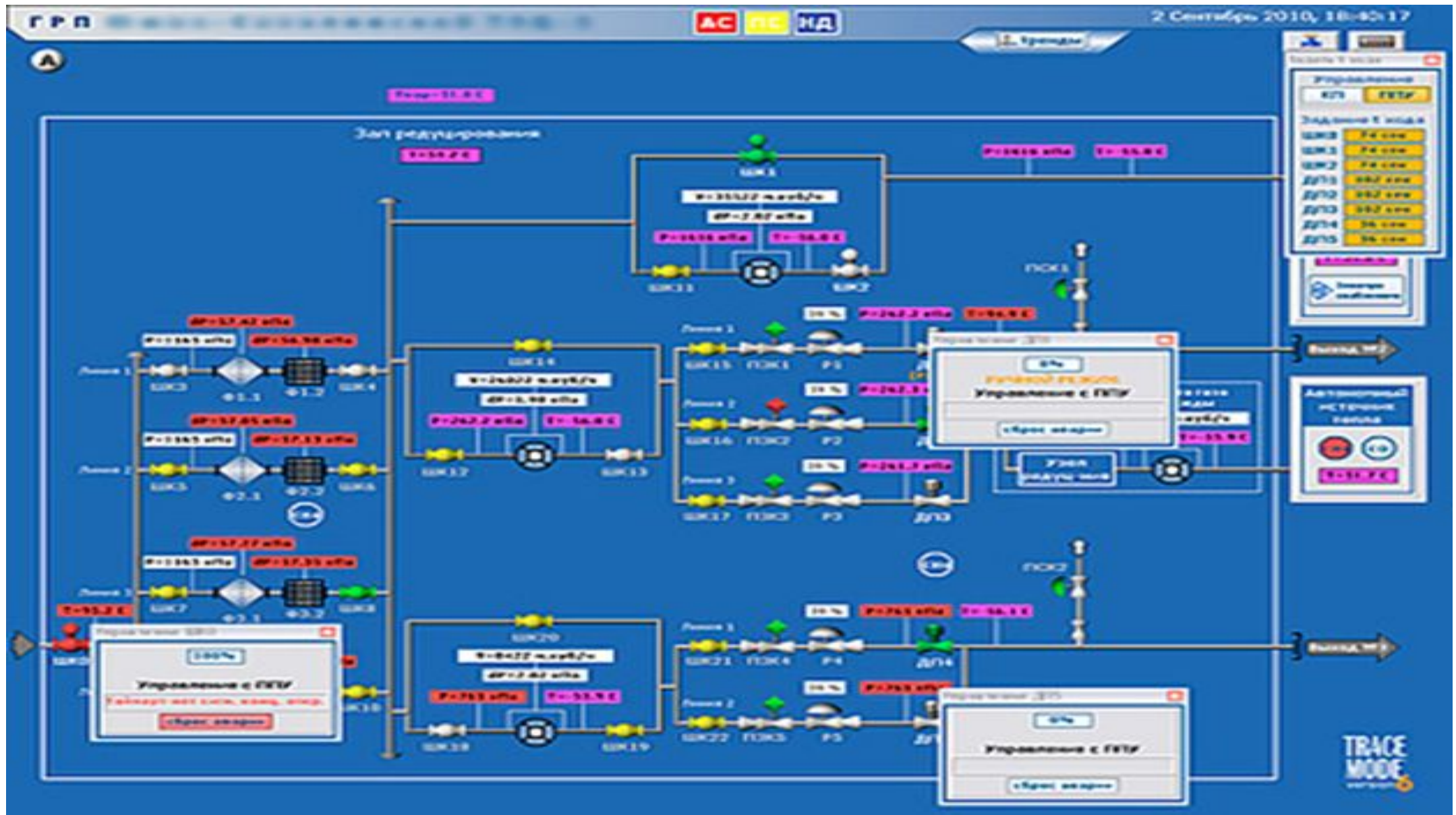
- **SCADA** (supervisory control and data acquisition, диспетчерське управління і збір даних) - програмний пакет, призначений для розробки або забезпечення роботи в реальному часі систем збору, обробки, відображення та архівування інформації про об'єкт моніторингу або управління.
- Термін "SCADA" має двояке тлумачення. Найбільш широко поширене розуміння SCADA як додатку, тобто програмного комплексу, що забезпечує виконання визначених функцій, а також інструментальних засобів для розробки цього програмного забезпечення. Однак, часто під SCADA-системою мають на увазі програмно-апаратний комплекс. Подібне розуміння терміна SCADA більш характерно для розділу телеметрія.



Основні задачі, які вирішує SCADA-системи

- Обмін даними з "пристроями зв'язку з об'єктом", (тобто з промисловими контролерами і платами вводу / виводу) в реальному часі через драйвери.
- Обробка інформації в реальному часі.
- Логічне управління.
- Відображення інформації на екрані монітора в зручній і зрозумілій для людини формі.
- Ведення бази даних реального часу з технологічною інформацією.
- Аварійна сигналізація і управління тривожними повідомленнями.
- Підготовка та генерування звітів про хід технологічного процесу.
- Здійснення мережевої взаємодії між SCADA-станціями (комп'ютерами).
- Забезпечення зв'язку з зовнішніми додатками (СУБД, електронні таблиці, текстові процесори і т. Д.). В системі управління підприємством такими додатками найчастіше є додатки, які відносять до рівня MES.

Операторський інтерфейс системи SCADA (приклад 0)



Типовий склад SCADA-системи

1. *Драйвери або сервери вводу-виводу* - програми, що забезпечують зв'язок SCADA з промисловими контролерами, лічильниками, АЦП і іншими пристроями введення і виведення інформації.
2. *Система реального часу* - програма, що забезпечує обробку даних в межах заданого тимчасового циклу з урахуванням пріоритетів.
3. *Людино-машинний інтерфейс* (НМІ, англ. Human Machine Interface - інструмент, який представляє дані про хід процесу людині оператору, що дозволяє оператору контролювати процес і керувати ним. *Програма-редактор* для розробки людино-машинного інтерфейсу.
4. *Система логічного управління* - програма, що забезпечує виконання призначених для користувача програм (скриптів) логічного управління в SCADA-системі. Набір редакторів для їх розробки.
5. *База даних реального часу* - програма, що забезпечує збереження історії процесу в режимі реального часу.
6. *Система управління тривогами* - програма, що забезпечує автоматичний контроль технологічних подій, віднесення їх до категорії нормальних, що попереджають або аварійних, а також обробку подій оператором або комп'ютером.
7. *Генератор звітів* - програма, що забезпечує створення призначених для користувача звітів про технологічні події. Набір редакторів для їх розробки.
8. *Зовнішні інтерфейси* - стандартні інтерфейси обміну даними між SCADA та іншими додатками. Зазвичай OPC, DDE, ODBC, DLL і т. Д.

Огляд ринку SCADA -систем

1. Повністю безкоштовні SCADA: *OpenSCADA, Rapid SCADA, FreeSCADA, scada-ГІНЕС, Inductive Automation Ignition.*

2. Умовно-безкоштовні SCADA:

- *Simp Light Free* - обмеження 8 тегів;
- *MasterSCADA* - обмеження 32 тега для MasterSCADA RT32 без додаткових можливостей або 1 годину повнофункціональної роботи для MasterSCADA Demo;
- *IGSS* - обмеження 50 об'єктів (орієнтовно 150 тегів) і вибір одного протоколу передачі даних (*IGSS FREE50*), по іншому типу ліцензії обмежено час роботи без перезапуску на 1 годину і 1000 об'єктів (*DEMO Mode*);
- *IntegraXor* вільна для 128 Modbus I / O;
- *Каскад*. Демо-версія має обмеження на 32 фізичних каналу введення / виводу і 2 години безперервної роботи, включає себе повну довідкову систему, SQL-сервер Firebird 2.5, WEB-модуль (реалізація WebSCADA) і ряд проектів, які демонструють можливості системи.;
- *Vijeo Citect* - дозволяє створити і налагодити повноцінний проект без придбання ліцензій, Ліцензується тільки середовище виконання (Runtime), середовище розробки поширюється вільно. У комплекті йде стандартний набір драйверів для підключення до контролерів і RTU різних виробників. Всього налічується більше 100 типів пристроїв, без урахування підтримки стандартних протоколів, таких як OPC, Modbus і т.д.

3. Інші SCADA: *Simatic WinCC, Intouch Wonderware, Trace mode, Genesis, SCADA Infinity, PcVue Solutions, RSView, ClearSCADA, DATARate, Контур, Коло-2000, ZenOn, Winlog, iFix, InduSoft Web Studio SCADA, Wizcon, Vijeo Citect, Статус-4, Каскад, Ентек, Sitex, Elipse E3, Elvis, Realflex RealWin SCADA, Broadwin (Advantech) WebAccess, General Electric Proficy Simplicity, WellinTech SCADA, Factory Link (з 2012 року не підтримується розробником), Monitor Pro (базувалася на Factory Link, рекомендована заміна на Vijeo Citect), Vijeo Look .*

WebSCADA

Під терміном WebSCADA, як правило, розуміється реалізація людино-машинного інтерфейсу (HMI) SCADA-систем на основі web-технологій.

Це дозволяє здійснювати контроль і управління SCADA-системою через стандартний браузер, який виступає в цьому випадку в ролі тонкого клієнта.

Архітектура таких систем включає в себе WebSCADA-сервер і клієнтські термінали - ПК, КПК або мобільні телефони з Web-браузером. Підключення клієнтів до WebSCADA-сервера через Internet / Intranet дозволяє їм взаємодіяти з прикладною задачею автоматизації як з простої web або WAP-сторінкою. Однак на даному етапі розвитку WebSCADA ще не досягло рівня широкого промислового впровадження, так як існують складності з захистом інформації, що передається. Крім цього, реалізація функцій управління через незахищені канали зв'язку суперечить міркувань безпеки будь-якого промислового об'єкта. У зв'язку з цим, в більшості випадків Web-інтерфейси використовуються в якості віддалених клієнтів для контролю і збору даних.

Особливості програмного забезпечення ВІС

1. Значне функціональне навантаження, яка обумовлене складністю об'єктів управління і завдань управління.
2. Налагодження програмного забезпечення при цьому представляє собою непросту задачу, оскільки пов'язана з необхідністю враховувати взаємодію всіх компонентів системи.
3. Доступ до вбудованої системи в робочих умовах отримати складно, особливо це стосується авіаційних, космічних та інших, що працюють в особливих умовах навколишнього середовища.
4. Високі вимоги до надійності і коректності роботи ВІС, особливо в нештатних ситуаціях.

VxWorks

VxWorks - вбудована операційна система «жорсткого» реального часу, що застосовується в пристроях з підвищеними вимогами до продуктивності і безпеки. Для спрощення рішення типових задач також існують готові пакети технологій (платформи) на базі VxWorks, орієнтовані на конкретні галузі промисловості (авіація / космонавтика, мережі / телекомунікації і т.п.).

Гібридне ядро, багатопоточність POSIX

«Плоска» модель пам'яті з налаштовуваними доменами захисту

«Жорсткий» реальний час: перемикання контексту / реакція на переривання - одиниці мкс,

Планувальник: витісняє, 256 рівнів пріоритету, підтримка ARINC 653

Дисципліни планування: карусельна

API: POSIX з розширеннями

Інструментарій: штатне інтегроване крос-середовище на базі Eclipse (Windows, Linux, Solaris)

Мови: C / C ++, Java, UML (IBMRational Rhapsody), Ada (AdaCore), Fortran

Компілятор GNU C / C ++ або Wind River Compiler (Diab)

Символьний крос-відладчик

Вбудована система управління версіями

Засоби верифікації та діагностики: аналізатор покриття коду, профайлер, монітор даних, аналізатор ОЗУ, системний профайлер

Візуальний конструктор вбудованих конфігурацій

Засоби управління тестовими сценаріями (модульне і інтеграційне тестування)

Конструктор ОС з вихідних текстів

Wind River Linux

Wind River Linux - комерційний дистрибутив ОС Linux, призначений для розробки вбудованих додатків. До складу дистрибутива входять більше 500 готових програмних пакетів, включаючи розширення «жорсткого» реального часу і пакети мережевих протоколів, а також інтегроване середовище розробки Wind River Workbench.

Архітектура і основні характеристики

Монолітне ядро з завантажуваними модулями, захист пам'яті процесів, многопоточність POSIX

Розширення «м'якого» і «жорсткого» реального часу

Планувальник: витісняє, 140 рівнів пріоритету

Дисципліни планування: FIFO, карусельна, адаптивна

API: POSIX з розширеннями

Ресурсоємність: ОЗУ / ПЗУ - одиниці-десятки Мбайт, підтримує виконання з ПЗУ (XIP)

Інструментарій розробника

Штатне інтегроване крос-середовище на базі Eclipse (Windows, Linux, Solaris)

Мови: C / C ++, Java, UML (IBM Rational Rhapsody), Python, Ruby, Fortran

Компілятор GNU C / C ++

Символьний крос-відладчик

Вбудована система управління версіями

Засоби верифікації та діагностики: аналізатор покриття коду, профайлер, монітор даних, аналізатор ОЗУ

Візуальний конструктор вбудованих конфігурацій з набором готових профілів ядра

Засоби управління тестовими сценаріями (модульне і інтеграційне тестування)

Конструктор ОС з вихідних текстів

QNX

QNX - вбудована ОС «жорсткого» реального часу для відповідальних застосувань. Завдяки архітектурі на основі мікроядра і вбудованим механізмам відмовостійкості QNX забезпечує міцний фундамент для високонадійних систем, що вимагають безвідмовної роботи в безперервному режимі. Підтримка стандартів POSIX забезпечує QNX гнучку розширюваність; в поєднанні з надійною архітектурою це дозволяє оперативно створювати на базі QNX системи широкої функціональності.

Архітектура і основні характеристики

Мікроядро, захист пам'яті процесів, багатопоточність POSIX

«Жорсткий» реальний час: перемикання контексту / реакція на переривання - одиниці мкс.

Планувальник: витісняє, 64 рівня пріоритету, адаптивне квотування

Дисципліни планування: FIFO, карусельна, спорадична

API: POSIX з розширеннями ресурсоємних: ОЗУ / ПЗУ - одиниці Мбайт,

Вбудована відмовостійкість, динамічне відновлення компонентів ($K_f \geq 0,99999$)

Інструментарій розробника

Інструментарій: штатне інтегроване крос-середовище на базі Eclipse (Windows)

Мови: C / C ++, Java (IBM WebSphere Studio Device Developer), UML (IBM Rational Rhapsody), Python, Ruby, Fortran

Компілятор GNU C / C ++ або Intel C ++ Compiler, бібліотеки GNU і Dinkum, вбудований C ++

Символьний крос-відладчик

Вбудована система управління версіями

Засоби верифікації та діагностики: аналізатор покриття коду, профайлер, монітор цільових систем, аналізатор ОЗУ

Візуальний конструктор вбудованих конфігурацій

RTOS-32

RTOS-32 - операційна система «жорсткого» реального часу для спеціалізованих вбудованих пристроїв на базі архітектури x86 / IA. Ядро RTOS-32 реалізує підмножина Win32 API з розширеннями реального часу, що дозволяє переносити в неї прикладний код, розроблений для Windows, і одночасно забезпечувати передбачуваність виконання і високу швидкість реакції. У поєднанні з високою компактністю це робить RTOS-32 ідеальним вибором для «глибоко вбудованих» систем реального часу - наприклад, промислових контролерів.

Архітектура і основні характеристики

Монолітне ядро, многопоточність Win32

«Плоска» модель пам'яті з підтримкою захисту сторінок

«Жорсткий» реальний час: перемикання контексту / реакція на переривання - від сотень нс,

Планувальник: кооперативний або витісняє, 64 рівня пріоритету

Дисципліни планування: карусельна, балансування навантаження по типу HPRN

API: підмножина Win32 з розширеннями

Інструментарій розробника

Інструментарій: Microsoft Visual Studio, Microsoft Visual C ++, Borland C / C ++, Borland C ++ Builder, Borland Delphi) і Ada (Aonix ObjectAda) в середовищі Windows

Мови: C / C ++ (з підтримкою компілятора Intel C ++ Compiler)

Засоби управління тестовими сценаріями (модульне і інтеграційне тестування)

Windows Embedded Standard

Windows Embedded Standard - вбудована ОС загального призначення, сумісна з настільною ОС Windows. Вона забезпечує багато- функціональність, знайомий інтерфейс і надійність Windows в формі, зручній для розробників сучасних комерційних і споживчих пристроїв. Завдяки бінарної сумісності програми Windows можуть виконуватися в середовищі Windows Embedded Standard без змін, а можливість вибору тільки необхідних компонентів ОС дозволяє зменшити обсяги споживаних ресурсів і знизити витрати на обладнання.

Архітектура і основні характеристики

Гібридне ядро, многопоточність Win32, захист пам'яті процесів

Планувальник: витісняє, 32 рівня пріоритету

Дисципліни планування: карусельна, адаптивна; балансування навантаження по типу HPRN

API: native NT, Win32, підмножина POSIX з розширеннями

ОЗУ / ПЗУ - сотні Мбайт

Інструментарій розробника

Інструментарій: Microsoft Visual Studio / Visual Studio Express в середовищі Windows

Мови: C / C ++, C #, Visual Basic .NET, F #, Ruby, Java

Засоби управління тестовими сценаріями (модульне і інтеграційне тестування)

Діагностичний інструментарій: профайлер, аналізатор динамічної пам'яті, монітор продуктивності

Інструментарій верифікації: статичний аналіз, аналіз покриття коду, засоби навантажувального тестування

Візуальний конструктор вбудованих конфігурацій

Особливості розробки програмного забезпечення для ВІС

Значна функціональне навантаження і підвищені вимоги до надійності роблять розробку програмного забезпечення вбудованих систем непростим завданням.

На етапі системного проектування необхідно, перш за все:

- правильно декомпонувати програмне забезпечення. Структура, сформована на цьому етапі, може бути природним чином застосована в кінцевому продукті при наявності зручних пропрограмих системних примітивів;
- слід передбачити налагоджувальні інтерфейси. Дане питання має вирішуватися на етапі розробки протоколів взаємодії між модулями програмного забезпечення.
- використовувати програмних імітаторів на різному рівні ієрархічної структури програмного забезпечення. Можливість написання таких імітаторів повинна забезпечуватися зручною архітектурою, що забезпечує обмін даними між модулями програмного забезпечення в реальному і модельному часі.
- використовувати стенди напівнатурного моделювання, в які закладена по можливості ідображення моделі навколишнього середовища. Це дозволяє перекопатися в працездатності програмного забезпечення в тому числі в позаштатних ситуаціях, що представляє особливий інтерес.
- предметно-орієнтований код модулів програмного забезпечення ВІС необхідно робити по можливості переміщуваним. Це дозволяє використовувати модульні тести, які суттєво спрощують налагодження і дозволяють уникнути пропрограмих помилок, а також робить можливим виконання обчислювальної частини програм в модельному часі (зокрема, для прискорення перевірки).

Стандарти в галузі ПЗ ВІС

- Стандарт **POSIX** був створений як стандартний інтерфейс сервісів операційних систем. Цей стандарт дає можливість створювати критичні додатки. Згодом цей стандарт був розширений особливостями режиму реального часу [POSIX].
- Стандарт **DO178B**, створений радіотехнічною комісією з авіації (RTCA, Radio Technical Commission for Aeronautics) для розробки ПО бортових авіаційних систем [DO178B]. Перша його версія була прийнята в 1982 р, друга (DO-178A) - в 1985-м, поточна DO-178B - в 1992 р, нова версія - DO-178C. Стандартом передбачено п'ять рівнів серйозності відмови, і для кожного з них визначено набір вимог до програмного забезпечення, які повинні гарантувати працездатність всієї системи в цілому при виникненні відмов даного рівня серйозності
- Стандарт **ARINC-653 (Avionics Application Software Standard Interface)** розроблений компанією ARINC в 1997 р Цей стандарт визначає універсальний програмний інтерфейс APEX (Application / Executive) між ОС авіаційного комп'ютера і прикладним ПО.
- Стандарт **OSEK / VDX** є комбінацією стандартів, які спочатку розроблялися в двох окремих консорціумах, згодом злилися. OSEK бере свою назву від німецького акроніма консорціуму, до складу якого входили провідні німецькі виробники автомобілів - BMW, Bosch, Daimler Benz (тепер Daimler Chrysler), Opel, Siemens і Volkswagen, а також університет в Карлсруе (Німеччина).