

УДК 681.5.015

П.Б. Вовк, А.П. Усїйчук

Луцький інститут розвитку людини Університету «Україна»

ПРОБЛЕМИ ПРОЕКТУВАННЯ ВБУДОВАНИХ СИСТЕМ

У статті досліджуються шляхи вирішення проблем розробки, відлагодження та експлуатації вбудованих систем. Розуміння принципів, методів, проектування комп'ютерних систем забезпечить вірний вибір технічних рішень щодо розв'язання проблем керування процесами.

Ключові слова: вбудована система, мікроконтроллер, однокристальна мікропроцесорна система, програмне забезпечення, процесор.

Літ. 6.

Постановка проблеми. Вбудована система (вбудовувана система, англ. Embedded system) - спеціалізована мікропроцесорна система управління, концепція розробки якої полягає в тому, що така система буде працювати, будучи вбудованою безпосередньо у пристрій, яким вона керує [1].

Тобто пристрій будується на базі вбудованого комп'ютера, який в той же час не сприймається користувачем пристрою як комп'ютер, так як не має звичного монітора і клавіатури, немає звичної операційної системи (ОС) та іншого програмного забезпечення (ПЗ).

Областю застосування вбудованих систем є:

- засоби автоматичного регулювання та керування техпроцесами, наприклад авіоніка, контроль доступу.
- верстати з ЧПУ.
- банкомати, платіжні термінали.
- телекомунікаційне обладнання.

Під "високою продуктивністю" вбудованих систем розуміють обчислювальну потужність, яка вимірюється мільярдами операцій в секунду. Наприклад, в мобільних телефонах наступного покоління, основаних на використанні методів програмно-керованого радіозв'язку, будуть потрібні мікропроцесори, які здатні виконувати не менше 10 мільярдів операцій в секунду. Аналогічно, висока продуктивність вбудованої обчислювальної апаратури потрібна для мультимедійних додатків, причому висока продуктивність потрібна завжди, а не тільки при піковому навантаженні.

Однак проблема вбудованих високопродуктивних комп'ютерів є досить важливою не тільки тому, що потрібно досягти дійсно дуже високого рівня продуктивності, а й через те, що при цьому потрібно витримувати дві важливі вимоги - реальний час і низьке споживання електроенергії.

Аналіз останніх досліджень.

Комп'ютерна наука стає все більш зрілою. Дослідникам вдається вирішити багато фундаментальних проблем цієї дисципліни, але для вирішення інших проблем потрібні нові проривні результати, які неможливо передбачити. Багато сучасних дослідницьких проблем - в областях Semantic Web, нанотехнологій, обчислювальної біології, сенсорних мереж - змушують діяти на межі можливостей існуючої технології та застосовувати її в нових програмах. Багато талановитих студентів тепер не збираються присвячувати своє життя комп'ютерній науці, а вибирають для вивчення біологічні науки або наноінженерію. У той же час комп'ютерна технологія все більше проникає в повсякденне життя людей і вбудоване програмне забезпечення управляє комунікаційними, транспортними та медичними системами.

Можливості вбудованих систем видаються необмеженими, від "розумних" будинків до автоматизованих автомагістралей, хоча їх вартість часто виявляється надмірно високою, а надійність - недостатньою. Хорошим прикладом є автомобільна промисловість. У кожен автомобіль встановлюється все більше електронних керуючих блоків, складність програмного забезпечення досягає такого рівня, на якому сучасні процеси і засоби розробки не можуть гарантувати достатню надійність систем при допустимих витратах.

Певні проблеми щодо вбудованих систем поставили Томас Гейзінгера (Thomas A. Henzinger, EPFL in Lausanne, Switzerland) та Джозеф Сіфакіс (Joseph Sifakis, Verimag). Стаття називається "Дисципліна проектування вбудованих систем" ("The Discipline of Embedded Systems Design").

Альберто Санджіованні-Вінцентеллі і Марко Ді Натале (Alberto Sangiovanni-Vincentelli, University of California, Berkeley, Marco Di Natale, Scuola Superiore S. Anna of Pisa, Italy) у статті "Розробка вбудованих систем для автомобільних додатків" ("Embedded System Design for Automotive Applications") звернули увагу на проблеми стандартизації у сфері вбудованих систем, особливо в автомобілебудуванні.

Як це не парадоксально, саме недоліки сучасних процесів розробки, валідації та супроводу призводять до того, що програмне забезпечення є найбільш дорогою і найменш надійною частиною вбудованих додатків. В результаті індустрія не може отримати вигоду з величезного потенціалу технологій вбудованої комп'ютерної та комунікаційної апаратури.

Основну причину цієї ситуації автори бачать у відсутності суворих методів проектування вбудованих систем. З одного боку, вбудовані системи в основному ігноруються як об'єкт досліджень в області комп'ютерної науки, оскільки в дослідженнях зазвичай застосовуються абстракції, що не дозволяють брати до уваги фізичні обмеження. З іншого боку, для проектування вбудованих систем виявляється недостатньо традиційного досвіду фахівців в галузі електроніки, тому що складовими частинами вбудованих систем є обчислення і програмне забезпечення.

Мета. Нині з'явився шанс надихнути фахівців в області комп'ютерних наук до проведення досліджень, присвячених проектуванню вбудованих систем. Проблема проектування вбудованих систем, безумовно, породжує ряд питань, що відносяться до технології, але більш важливо те, що для її вирішення потрібно створення нового наукового фундаменту, який дозволяв би систематично інтегрувати обчислювальні та фізичні компоненти системи. Для створення такого фундаменту будуть потрібні збагачені комп'ютерні парадигми, що включають моделі і методи, які традиційно використовуються при розробці електронних приладів.

Паралельно цьому викладачам буде потрібно оновити програми комп'ютерних навчальних курсів. В індустрії підготовлені фахівці в галузі електроніки повсякденно проектують архітектури програмних систем і підготовлені комп'ютерні фахівці постійно стикаються з фізичними обмеженнями. Проте, проектування вбудованих систем займає периферійні позиції в навчальних програмах, що відносяться до обох цих напрямків. Різниця в культурах цих двох областей в чому пояснюється тим, що комп'ютерна наука в основному базується на дискретній математиці, а традиційні технічні дисципліни - на безперервній математиці. В індустрії відчайдушно потрібні інженери, які відчували б себе як вдома в обох світах. У дисципліні проектування вбудованих систем є потенціал для створення таких інтегрованих фахівців. Але для побудови наукового фундаменту цієї дисципліни буде потрібно узгоджені зусилля дослідників, викладачів і виробничих фахівців [2].

Виклад основного матеріалу. Еволюція поняття «Вбудовані системи»:

- Інформаційно-керуючі системи, керуючі обчислювальні комплекси (НБК), переважно централізовані системи віддалені від об'єкта управління, 60-і роки.
- Вбудовані обчислювальні системи (embedded systems), компактні обчислювальні системи вбудовані в об'єкт управління, кінець 70-х років.
- Розподілені вбудовані системи управління, контролерні мережі (networked embedded control systems) - NECS / PIUC, кінець 90-х років.
- Кібер-фізичні системи (Cyber Physical Systems), 2000-і роки.

Дослідження в сфері кібер-фізичних систем (CPS) передбачають отримання нових наукових і технологічних досягнень, які забезпечать швидке та надійне проектування та інтеграцію кібер- (обчислювально-) і інфоцентрованих фізичних та інженерних систем. Все, що пов'язано з вбудованими комп'ютерами, часто асоціюється з простими контролерами в дешевих продуктах. Але хоча невеликі контролери дійсно важливі (щорічно продаються мільярди 8-бітових мікроконтролерів), ними далеко не вичерпується дана область. Високопродуктивні мікроконтролери використовуються як поодинокі, так і пов'язані в мережу в найрізноманітніших додатках.

У зв'язку з тим, що система управління буде розміщуватися 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 відводу тепла (охолодження) забезпечують мінімум вимог теплових режимів. Якщо щільність теплового потоку (тепловий потік, що проходить через одиницю поверхні) не перевищує 0,5 мВт/см², перегрів поверхні пристрою щодо навколишнього середовища не перевищить 0,5 °С, така апаратура вважається нетеплонавантаженою і не вимагає спеціальних схем охолодження.
- мікропроцесор і системна логіка, а також ключові мікросхеми по можливості поєднані на одному кристалі.
- спеціальні військово-космічні вимоги з радіаційної та електромагнітної стійкості, працездатність у вакуумі, гарантований час напрацювання, термін доступності рішення на ринку і т. д.

Основою побудови простих вбудованих систем часто служать одноплатні (однокристальні) ЕОМ або мікроконтролери, спеціалізовані або універсальні мікропроцесори, ПЛІС. Для побудови деяких видів вбудованих систем широко використовують мікропроцесори архітектури ARM [3].

Деякі вбудовані системи використовуються в масових кількостях (наприклад, пристрої RFID). Вбудовані системи є привабливою ціллю для творців шкідливого коду через свою поширеність і відносній беззахисності. Поступово виникає шкідливий код для вбудованих систем (наприклад, RFID-вірус, Cabir). Цей процес поки що стримується різноманітністю вбудованих пристроїв, відсутністю домінуючого ПЗ і обмеженою функціональністю деяких видів пристроїв. З іншого боку, завдання антивірусних компаній і дослідників комп'ютерної безпеки також ускладнена цими обставинами, а також малопотужні вбудовані системи найчастіше не дозволяють скористатися поширеним антивірусним ПЗ.

Центральним процесорним пристроєм для вбудованої системи можуть служити сучасні мікропроцесори і мікроконтролери (МК). Конкретний вид визначається при проектуванні, виходячи з цілей і завдань виконуваних вбудованої системою.

Мікропроцесорні системи (МПС) на основі МК використовуються найчастіше як вбудовані системи для вирішення завдань управління деяким об'єктом. Важливою особливістю даного застосування є робота в реальному часі, тобто забезпечення реакції на зовнішні події протягом визначеного часового інтервалу. Такі пристрої отримали назву контролерів.

Однокристальні мікроконтролери (ОМК) є найбільш масовим видом пристроїв сучасної мікропроцесорної техніки, річний обсяг випуску яких становить понад 2,5 млрд. штук. Інтегруючи на одному кристалі високопродуктивний процесор, пам'ять і набір периферійних схем, ОМК дозволяють з мінімальними витратами реалізувати високоефективні системи та пристрої керування різними об'єктами (процесами). На відміну від звичайних мікропроцесорів, для роботи яких необхідні зовнішні інтерфейсні схеми, в корпусі ОМК поряд з основними функціональними вузлами розміщені такі допоміжні вузли, як тактовий генератор, таймер, контролер переривань, цифро-аналоговий і аналого-цифровий перетворювачі, порти вводу-виводу.

Світова промисловість випускає величезну номенклатуру мікроконтролерів. За областю застосування їх можна поділити на два класи: спеціалізовані, призначені для застосування в якій-небудь одній конкретній області (контролер для телевізора, контролер для модему, контролер для комп'ютерної мишки) і універсальні, які не мають конкретної спеціалізації і можуть застосовуватися в різних областях мікроелектроніки, за допомогою яких можна створити як будь-який з названих вище пристроїв, так і принципово нові пристрої.

Завдяки цим якостям ОМК знаходять широке застосування в системах промислової автоматики, контрольно-вимірювальних приладах і системах, апаратурі зв'язку, автомобільної електроніки, медичному обладнанні, побутовій техніці та багатьох інших областях.

Застосування однокристальних мікроконтролерів дозволяє перенести основні витрати, пов'язані з розробкою вбудованих систем управління, з апаратної в програмну область. Це неминуче тягне за собою збільшення складності програмного забезпечення мікроконтролерів.

Особливістю розробки ПЗ для ОМК є використання мови низького рівня - мови асемблера. Це пов'язано з тим, що при реалізації вбудованих систем критичними є час реакції на зовнішні впливи, час виконання заданих процедур обробки даних, розмір програмного коду і області даних.

Перед розробником вбудованих МПС стоїть завдання реалізації повного циклу проектування, починаючи від розробки алгоритму функціонування і закінчуючи комплексними випробуваннями в складі виробу, а, можливо, і супроводом при виробництві.

Нові методи аналізу системного рівня і моделювання потрібні не тільки для забезпечення передбачуваності і можливості компонування при декомпозиції наскрізних функцій під час

проектування (і пізніше, під час системної інтеграції), але також і для забезпечення підтримки проєктувальників на дуже ранній стадії, коли виробляються оцінка та вибір електроніки та програмної архітектури для лінійки продуктів. Ця важлива стадія робить сильний вплив на вартість, експлуатаційні характеристики і якість лінійки продуктів.

Вибір архітектури зазвичай проводиться за роки до реальної розробки та інтеграції підсистем. У цьому процесі мають бути визначені моделі функцій та можливі рішення для реалізації фізичної архітектури, і повинні бути зроблені оцінка якості та вибір найкращої апаратної платформи по відношенню до продуктивності, надійності і цінових показників.

Враховуючи високу вартість досліджень, підготовки фахівців і, можливо, придбання ліцензій для проєктування на системному рівні, бажаним є використання узгодженого набору моделей, методів та інструментальних засобів у всьому життєвому циклі продукту або платформи. Ця рекомендація поширюється зі стадії аналізу архітектури на стадію декомпозиції і проєктування системи і включає кроки розробки на основі моделей із застосуванням відповідного проміжного програмного забезпечення і автоматичної генерації коду додатків, остаточної інтеграції, тестування і валідації.

Суттєвою технічною проблемою є формування стандартів, що забезпечують сумісність між програмними і апаратними рішеннями, ліцензованим у власників інтелектуальної власності, та інструментальними засобами інтеграції. Міжнародний консорціум AUTOSAR ставить в якості однієї з основних своїх цілей рішення цієї проблеми [4].

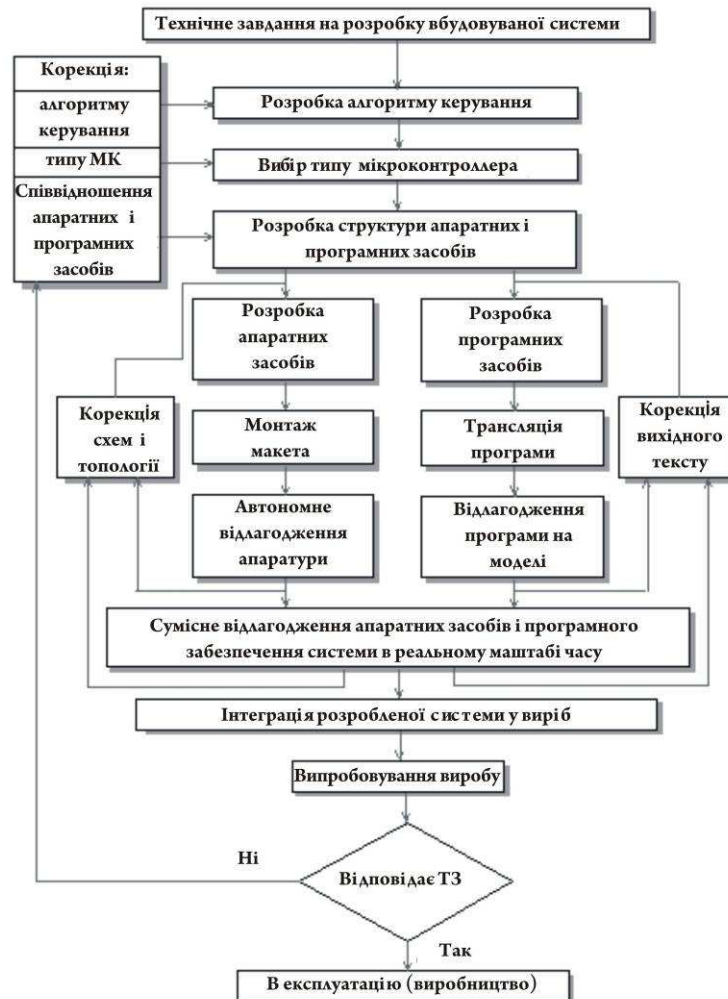


Рис 1. Основні етапи розробки вбудованої системи
 Авторська розробка

Якщо програмне забезпечення, яке надходить з різних джерел, має інтегруватися на загальній апаратній платформі за відсутності правил композиції та формальної верифікації властивостей складовою системи, то хто повинен відповідати за коректність функціонування остаточного продукту? Хто б не прийняв на себе відповідальність за специфікацію підсистем та їх

інтеграцію, йому буде потрібно сувора методологія і залізна воля, щоб змусити постачальників відповідати їй. Цього може виявитися недостатньо, оскільки характеристики програмного забезпечення важко точно визначити. У розробників може просто відсутня можливість гарантувати функціональну поведінку системи, її часові характеристики і надійність при наявності зовнішніх компонентів.

Постійне зростання складності проектування вбудованих систем робить непрактичними і схильними до помилок аналіз і проектування, вироблені вручну. Ідеальний підхід мав би полягати в автоматичному відображенні набору завдань на платформу з забезпеченням гарантій коректного функціонування та витримування тимчасових співвідношень при оптимальному використанні ресурсів.

На теперішній час сформована методологія проектування вбудованих систем, що може бути представлена так, як показано на рисунку 1.

У технічному завданні формулюються вимоги до системи з погляду реалізації певної функції управління. Технічне завдання включає в себе набір вимог, який визначає, що користувач хоче від вбудованої системи і що розроблювальний прилад повинен робити. Технічне завдання може мати вигляд текстового опису, що в загальному випадку не є вільним від внутрішніх протиріч.

На підставі вимог користувача складається функціональна специфікація, яка визначає функції, які контролер реалізовує для користувача після завершення проектування, уточнюючи тим самим, наскільки пристрій відповідає пропонованим вимогам. Вона включає в себе описи форматів даних, як на вході, так і на виході, а також зовнішні умови, що керують діями контролера.

Етап розробки алгоритму управління є найбільш відповідальним, оскільки помилки даного етапу зазвичай виявляються тільки при випробуваннях закінченого виробу і призводять до необхідності переробки всього пристрою. Розробка алгоритму звичайно зводиться до вибору одного з кількох можливих варіантів алгоритмів, що відрізняються співвідношенням обсягу програмного забезпечення та апаратних засобів.

При цьому необхідно виходити з того, що максимальне використання апаратних засобів [5] спрощує розробку і забезпечує високу швидкодію контролера в цілому, але супроводжується, як правило, збільшенням вартості і споживаної потужності. При виборі типу МК враховуються такі основні характеристики:

- розрядність;
- швидкодія;
- набір команд і способів адресації;
- вимоги до джерела живлення і споживана потужність в різних режимах;
- обсяг ПЗУ програм і ОЗУ даних;
- можливості розширення пам'яті програм і даних;
- наявність і можливості периферійних пристроїв, включаючи засоби підтримки роботи в реальному часі (таймери, процесори подій тощо);
- можливість перепрограмування в складі пристрою;
- наявність і надійність засобів захисту внутрішньої інформації;
- можливість постачання в різних варіантах конструктивного виконання;
- вартість у різних варіантах виконання;
- наявність повної документації;
- наявність і доступність ефективних засобів програмування і налагодження МК;
- кількість і доступність каналів постачання, можливість заміни виробами інших фірм.

Список цей не є вичерпним, оскільки специфіка проектуваного пристрою може перенести акцент вимог на інші параметри МК.

Номенклатура МК, що випускаються в даний час, обчислюється тисячами типів виробів різних фірм. Сучасна стратегія модульного проектування забезпечує споживача різноманітністю моделей МК з одним і тим же процесорним ядром. Така структурна розмаїтість відкриває перед розробником можливість вибору оптимального МК, що не має функціональної надмірності, що мінімізує вартість комплектуючих елементів.

Однак для реалізації на практиці можливості вибору оптимального МК необхідне досить глибоке опрацювання алгоритму управління, оцінка обсягу виконуваної програми і числа ліній сполучення з об'єктом на етапі вибору МК. Допущені на даному етапі прорахунки можуть згодом

привести до необхідності зміни моделі МК і повторного розведення друкованої плати макета контролера. У таких умовах доцільно виконувати попереднє моделювання основних елементів прикладної програми з використанням програмно-логічної моделі обраного МК.

На етапі розробки структури контролера остаточно визначається склад апаратних модулів, протоколи обміну між модулями, типи роз'ємів. Виконується попереднє опрацювання конструкції контролера. У частині програмного забезпечення визначаються склад і зв'язки програмних модулів, мова програмування. На цьому ж етапі здійснюється вибір засобів проектування і налагодження.

Висновки. Вважайте, що всі системи - системи жорсткого реального часу, де відгук на зовнішній вплив повинен бути не пізніше n -ної кількості часу. Бажано ще й мінімального;

Функція `main ()` повинна закінчуватися безскінченним циклом. Без `return` взагалі. Передбачайте стани системи, в якому вона принесе мінімальної шкоди! Так звана захисна відмова. Якщо раптом у робота відвалюється датчик положення виконавчого органу - він повинен зупинитися, а не намагатися перейти в безпечне положення. Потрібно, щоб і користувач дізнався, що пристрій знаходиться у такому стані. Наприклад, перевести годинник в 88:88 - краще, ніж залишити дисплей незмінним. Дуже корисно перевіряти обладнання, підключене до мікроконтролера, хоча б при ініціалізації. Не всяке можна перевірити, але те, що можна - обов'язково [6]!

Програма для мікроконтролера - багатопотокова в більшості випадків, і `main ()` повинна бути потоком з мінімальним пріоритетом. Потоки з більш високим пріоритетом - обробники переривань. Переривання відключати без крайньої необхідності (наприклад, переходу в стан захисного відмови) не слід. Обробник переривання не повинен виконуватися довше, ніж мінімальний період між перериваннями, а то не залишиться системі часу на менш пріоритетні завдання. А як час оцінювати - так це просто. Один вивід мікроконтролера повинен бути призначений для звукової чи світлової сигналізації. Ідеально на нього підключити світлодіод і сигналізувати ним про все, що тільки можна.

Грамотно розставляйте пріоритет переривань! У годиннику пріоритет - за кнопками управління. Якщо є кнопка аварійної зупинки - вона повинна мати вищий пріоритет: невідомо що в пристрої відбудеться і де програма зациклиться, але повинна бути можливість його зупинити. Вищий пріоритет - у аварійних сигналів, далі - за важливістю функцій пристрою.

Дуже важливо враховувати час виконання обробника і періодичність виникнення переривань для задач комунікації.

Знай і любі обладнання! Проблеми на шині можуть вилитися в довгі години налагодження та невідтворювані помилки. Використання асемблера далеко не завжди виправдане: виграш у долі часу може призвести до поганого читання коду.

1. Матеріал из Википедии — свободной энциклопедии
http://ru.wikipedia.org/wiki/Встраиваемая_система
2. Кузнецов С. Высокопроизводительные встроенные системы. Електронне видання. Режим доступу: <http://citforum.ru/computer/2007-10/>
3. Уилмсхерст Т. Разработка встроенных систем с помощью микроконтроллеров PIC. -М.: МК-Пресс. 2008, 544с.
4. Барретт С.Ф., Пак Д. Дж. «Встраиваемые системы. Проектирование приложений на микроконтроллерах семейства 68HC12/HCS12 с применением языка C» (электронная книга) Режим доступу: <http://www.rulit.net/books/vstraivaemye-sistemy-proektirovanie-prilozhenij-na-mikrokontrollerah-semejstva-68hc12-hcs12-s-primen-read-237830-1.html>
5. Arnold S. Berger. Embedded Systems Design: An Introduction to Processes, Tools, and Techniques. CMP Books © 2002. ISBN: 1-57820-073-3
6. Роберт Гласс. Факты и заблуждения профессионального программирования. "Символ", 2008. ISBN 13: 978-5-93286-092-2