

I.V. ГУЛА, Л.Є. БАЙДИЧ
Хмельницький національний університет
А.В. КЛЕПІКОВСЬКИЙ, Є.Г. МАХРОВА
Буковинський державний медичний університет

ЗАСТОСУВАННЯ ОПЕРАЦІЙНИХ СИСТЕМ РЕАЛЬНОГО ЧАСУ В ЗАДАЧАХ ВИМІРЮВАННЯ ТА ОБРОБКИ ВИМІРЮВАЛЬНОЇ ІНФОРМАЦІЇ

Застосування сучасних технічних рішень у вигляді вбудованих електронно-вимірювальних систем на базі сучасних мікроконтролерів із ядром ARM надає потенційному споживачу широкі можливості у виконанні задач вимірювання. Великий обсяг вбудованої пам'яті та доступ до швидкої зовнішньої пам'яті надає можливість виконання широкого спектру задач. Проте, створення програмного забезпечення, що здатна керувати такими програмними додатками як модулі вимірювання, опрацювання отриманого сигналу, взаємодія з дисплеєм та модулями вводу-виводу частото забирає велику частину часу на розв'язання задач взаємодії цих частин в часі. Застосування операційних систем реального часу здатна спростити розробку відповідного забезпечення.

Ключові слова: ARM, операційна система реального часу.

I.V. GULA, L.E. BAYDUCH
Khmelnitsky national university
A.V. KLEPIKOVSKYI, YE.H. MAKHROVA
Bukovina state medical university

REAL TIME OPERATING SYSTEM USAGE FOR MEASUREMENT PROCESS AND HANDLING OF MEASUREMENT INFORMATION

The use of modern technical solutions in the form of embedded electronic measuring systems based on modern microcontrollers with ARM core provides potential consumers opportunities in performance measurement tasks. A large amount of memory capacity and fast access to the external memory allows us to perform a wide range of tasks. However, software that is able to manage software applications such as measurement modules, processing the received signal, the interaction of display and input-output frequencies trim most of the time for solving problems of interaction of these parts in time. The use of real-time operating system is capable to simplify the development of appropriate software.

Keywords: ARM, real time operating system.

Вступ. Розвиток технологій виробництва промислових ПК (ППК) привели до посилювання конкуренції на ринку систем автоматизації виробництва між апаратними рішеннями на базі мікроконтролерів і програмними комплексами, що працюють під управлінням спеціалізованих операційних систем (ОС) на інтегрованих ППК платформах.

Постановка задачі дослідження. Виконання задачі з підтримки технологічного процесу вимагає від сучасного інженера не тільки розуміння операцій, що складають технологічний процес, а також і чіткого розуміння необхідності дотримання точних часових інтервалів в процесі виконання всього циклу технологічних операцій. Вже не є чимось дивним застосування високоточного технологічного обладнання не тільки на спеціалізованих підприємствах, а й доступність такого обладнання для малочисельних підприємств. Від технологічного обладнання часто вимагається досягнення як багатофункціональності так і простоти в реалізації та, відповідно, мінімальної вартості. З іншого боку, розробка програмного забезпечення для реалізації цієї багатофункціональності також вимагає від інженера-розробника розуміння всіх складнощів створення багатозадачної системи.

Одним з найбільш спірних і складних термінів систем автоматизації є англійський вираз "realtime" і відповідне йому поняття – "фізичний (реальний) час". Поняття це застосовується в різних науково-технічних областях і має на увазі якісь дії, тривалість яких визначається зовнішніми процесами за умов фізичних процесів. Термін "realtime" також встановлює чітке обмеження в розумінні того, що процес, до якого він відноситься, не може бути виконаний за окремі інтервали часу. Ці інтервали часу не можуть бути відокремлені між собою, або виконання цих процесів не допускає зміни часу їх виконання. Зазвичай до спеціалізованих ОС що реалізуються в контексті систем реального часу (СРЧ) відносяться операційної системи реального часу (ОСРЧ, англійською – Real-Time Operating Systems, RTOS). У літературі зустрічається достатньо велика кількість визначень цього поняття, але терміни "СРЧ" можуть бути визначені як комбінація наступних двох визначень [3]:

1. Система називається системою реального часу, якщо правильність її функціонування залежить не тільки від логічної коректності обчислень, але і від часу, за який ці обчислення проводяться. Тобто для подій, що відбуваються в такій системі, то, КОЛИ ці події відбуваються, так само важливо, як логічна коректність самих подій [3, 4].

2. Реальний час (програмне забезпечення) (IEEE 610.12 – 1990): Відноситься до системи або режиму роботи, в якому обчислення проводяться протягом часу, визначуваного зовнішнім процесом, з метою управління або моніторингу зовнішнього процесу за наслідками цих обчислень.

Обидва ці вирази підкреслюють головну вимогу до СРЧ – ці системи повинні виконувати свої операції вчасно. Можливість такої реалізації визначена Мартіном Тіммерманом (Martin Timmerman) [6]:

Системи реального часу – це системи, які передбачено у сенсі часу реакції реагують на не передбачені за часом появи зовнішні події. З цього визначення витікає головна властивість систем реального часу: передбаченість або детермінованість [5]. Тільки завдяки цій властивості розробник може гарантувати функціональність і коректність спроектованої системи. При цьому власне швидкість реакції системи важлива тільки щодо швидкості протікання зовнішніх процесів, за якими СРЧ повинна стежити, або якими повинна управляти.

Основна частина. З приведених визначень виходить, що СРЧ покликані вирішувати завдання, в яких важливі не тільки правильність рішення, але і терміни, в які ці рішення ухвалюються. У зарубіжній літературі термін, в межах якого повинне бути ухвалене рішення називається *deadline*, що може бути переведене як критичний термін обслуговування. Якщо невиконання завдання в критичний термін обслуговування означає, що вона взагалі не була виконана, то такі завдання називають завданнями жорсткого реального часу. В більшості випадків про завдання жорсткого реального часу говорять тоді, коли порушення термінів критичного обслуговування може завдати значного матеріального або фізичного збитку. До завдань м'якого реального часу відносять випадки, коли порушення критичного часу обслуговування веде до неприємних, але допустимих наслідків (наприклад, вимагає додаткової обробки).

СРЧ в галузі вимірювальної техніки у більшості випадків вирішують комбінацію завдань жорсткого і м'якого реального часу, а також завдань, що не мають критичного терміну обслуговування. Іноді завдання може переходити із статусу м'якого реального часу при пропуску деякого терміну обслуговування в статус завдання жорсткого реального часу з призначенням критичного терміну обслуговування.

Якщо СРЧ будується як програмний комплекс, то, в загальному вигляді, вона може бути представлена як комбінація трьох компонент: прикладного програмного забезпечення, операційної системи реального часу (ОСРЧ) і апаратного забезпечення. При розробці СРЧ необхідний ретельний аналіз відповідності характеристик цих трьох компонент вимогам зовнішнього об'єкту, для управління або моніторингу яким ця СРЧ призначена. Фактично апаратне забезпечення визначає часові рамки виконання, а отже визначає подальшу здатність до використання тієї чи іншої СРЧ. У цілому ряду завдань автоматизації програмні комплекси повинні працювати як складова частина крупніших автоматичних систем без безпосередньої участі людини.

Отже, створення багатофункціональної та, зазвичай одночасно з цим, багатозадачної системи керування технологічним процесом при застосуванні обмеження у процесорній складності зводиться фактично до двох шляхів розвитку:

- це створення власного програмного продукту, що буде адаптовано під конкретну реалізацію апаратного забезпечення;
- використання вже готового рішення у якості готової операційної системи реального часу.

Перший варіант вимагає капітальних вкладень в початкову розробку як програмного забезпечення, так і створення та ведення документації. Будь-яка потреба в розширенні програмного забезпечення як підтримка нового типу файлової системи або підтримка певного типу апаратного забезпечення, наприклад USB-HOST з можливістю "гарячого" підключення, вимагатиме знову та знову залучення додаткових грошових ресурсів, не кажучи про складність ведення робочої групи, що буде орієнтуватись у створеному програмному продукті.

Застосування операційної системи реального часу дозволяє спростити процес створення багатофункціональної системи за рахунок застосування принципово іншого підходу в процесі розробки програмного коду.

Для реалізації вимірювальної системи потрібно виконувати ряд принципово різних за тривалістю паралельних операцій. Реалізація вимірювальної системи на базі широковідомих операційних систем типу Windows або Linux не дозволяє вирішувати ці задачі оскільки ці операційні системи не задовольняють вимоги щодо часу реакції на зовнішню подію [5]. Крім того, незважаючи на широку доступність розробленого програмного забезпечення під ці операційні системи, слід вказати на особливості апаратної реалізації. Так основні плати побудовані із застосуванням шини PCI/PCI express вимагають введення спеціалізованих плат розширення з одного боку, а також спеціалізованих драйверів для операційної системи.

Суттєвий розвиток та широке розповсюдження 8/16-бітних мікроконтролерів, а останнім часом - 32-х бітних мікроконтролерів з одночасним збільшенням вбудованої периферії та стрімким зменшенням ціни навіть у дрібних партіях складає потужну конкуренцію для 16/32-х бітних процесорів сімейства x86 від Intel/AMD. Яскравим підтвердженням цьому є розробка Windows 8 для архітектури ARM [1]. Отже, очевидним є використання ОСРЧ для вбудованих систем на базі архітектури ARM.

Вибір операційної системи реального часу для вимірювальної установки та практична імплементація

На сьогодні відомо 169 операційних систем реального часу [2] для мікроконтролерів. Частина з цього списку - це системи розроблені під вузькі сімейства мікроконтролерів. З наведеного списку 83 мають відкритий програмний код, що дозволяє модифікувати операційну систему під свої потреби. Для створення апаратної частини вимірювальної установки доступно широка номенклатура мікроконтролерів. Найбільш поширене сімейство - це мікроконтролери на базі архітектури ARM [1].

Для створення вимірювального пристрою обрано вже готову платформу на базі контролера

SMT32F103RB та модулем дисплею, що оснащено 2.8 дюймовим TFT індикатором з керуванням на базі контролера ILI9325 та контролером сенсорного керування ADS7843 (рис. 1).

Для керування процесом вимірювання обрано операційну систему RTX Kernel від Keil. Перевага цієї операційної системи надана оскільки:

- надається у вихідних кодах;
- широка підтримка як по кількості задач так і по їх пріоритетах;
- гнучка взаємодія між задачами;
- висока швидкість та низький час затримок при перериваннях;
- підтримка багатопоточності;
- інтегрована підтримка у середовищі програмування Keil μ Vision4 (рис. 2).



Рис. 1. STM32 модуль з модулем TFT дисплею

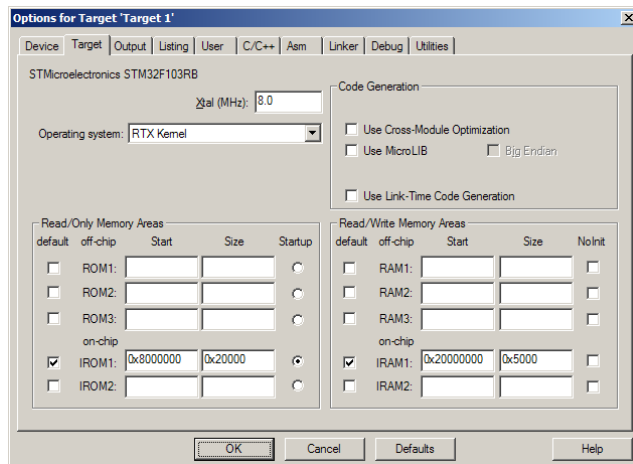


Рис. 2. Вікно налаштування проекту у середовищі Keil μ Vision

Використана операційна система не є найкращою в загальному розумінні, оскільки підтримується тільки розробником цієї системи. ОСРЧ RTX Kernel зручна для швидкого створення програмних додатків відносно простих вбудованих рішень. Для створення більш потужних рішень більш краща FreeRTOS. FreeRTOS має відкритий вихідний код, добре документована та підтримує більше

Підтримка у середовищі програмування Keil μ Vision4 дозволяє почати створення проекту майже без затримок. Взаємодія з операційною системою ґрунтується на декількох базових операціях.

os_sys_init - ініціалізація операційної системи та запуск першої задачі.

os_tsk_create - ініціалізація задачі. Дозволяє повідомити операційну систему про певну процедуру, яка буде взаємодіяти з системою як окрема задача, для якої виділяється процесорний час.

OS_TID - ідентифікатор задачі. Використовується для зберігання інформації про задачу.

os_evt_set та **os_evt_wait_or** - процедури обміну повідомленнями-сигналами між задачами. Дозволяють передавати та перевіряти появу певного стану від задачі до задачі.

os_tsk_delete_self - завершення задачі.

Таким чином, взаємодія між задачами в операційній системі зводиться до ініціалізації окремих задач, таких як вимірювання вхідного сигналу, обробка отриманої інформації, запис результатів із застосуванням наявного в операційній системі механізму взаємодії.

Фрагмент коду наведено нижче. Для спрощення, програмний код задач не показано.

```
// Ініціалізація системних змінних
#include <RTL.h>
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include "measure.h"
#include "STM32F10x.h"

// налаштування операційної системи
OS_TID t_command;
OS_TID t_measure_dig;
OS_TID t_measure_an;
OS_TID t_calculate;
OS_TID t_work MMC;

// опис задач
__task void task_measure_dig(void) { // програмний код для спрощення не приведено
}
__task void task_measure_an(void) {}
__task void task_calculate(void) {}
```

```
__task void task_work_MMC(void) {}

// задача task_command використовується для опису всіх наступних задач та запуску їх
__task void task_command(void)
{
    t_command = os_tsk_self();
    t_measure_dig = os_tsk_create(task_measure_dig,1);
    t_measure_an = os_tsk_create(task_measure_an,2);
    t_calculate = os_tsk_create(task_calculate,3);
    t_calculate = os_tsk_create(task_work_MMC,4);
}

// головна процедура. Відбувається запуск системи
int main(void)
{
    os_sys_init(task_command); // запуск системи та задачі
}
```

Далі, залишається перевірити роботу створеного програмного рішення із застосуванням вбудованих засобів відлагодження, що є в Keil μ Vision.

Висновки

Створення прикладних програм із широкими функціональними можливостями із застосуванням сучасного програмного забезпечення такого як середовище програмування Keil μ Vision та доступних безкоштовних операційних систем реального часу дозволяє створювати багатозадачні вбудовані рішення із мінімальними витратами часу від розробника. Наведена в роботі ОСРЧ RTX Kernel від Keil при взаємодії з середовищем Keil μ Vision дає потужний потенціал для створення та відлагодження багатозадачних вбудованих рішень.

Література

1. Нездоровін В. П. Архітектура ARM як потенційна основа ГРІД-інфраструктури наукової бази України / В. П. Нездоровін, К.Л. Горященко, Є. Г. Махрова // Вісник ХНУ. - 2012. - №1. - С.204-208
2. http://en.wikipedia.org/wiki/List_of_real-time_operating_systems
3. Сорокин С. Системи Реального Часу. / С. Сорокин // СТА. – 1997. -№2. – 3 22-29.
4. Верхалст Ерік. Завдання розробки ОСРВ для цифрової обробки сигналів / Ерік Верхалст // Мир комп'ютерної автоматизації. – 1997. – №4.
5. Бліськавіцький А. А. Операційні системи реального часу (огляд) / А. А. Бліськавіцький, С. В. Кабаєв. // Засоби і системи комп'ютерної автоматизації. <http://www.asutp.ru>.
6. Martin Timmerman, Bart Van Beneden, Lourent Uhres. RTOS Evaluation Kick Off! // Real-Time Magazine. – 1998. – N3 pp 6 – 10

References

1. Nezdorovin V. P., Gorjashhenko K.L., Mahrova S. G. Arhitektura ARM jak potencijna osnova GRID-infrastrukturi naukovoi bazi Ukraїni. Visnik HNU. 2012. №1. P. 204-208
2. http://en.wikipedia.org/wiki/List_of_real-time_operating_systems
3. Sorokin S. Sistemi Real'nogo Chasu. STA. 1997. №2. P. 22-29.
4. Verhalst Erik. Zavedannja rozrobki OSRV dlja cifrovoi obrobki signalov. Mir komp'juternoї avtomatizacїi. 1997. №4.
5. Blis'kavickij A. A., Kabajev S. V. Operacijni sistemi real'nogo chasu (ogljad). Zasobu i sistemi komp'juternoї avtomatizacїi. <http://www.asutp.ru>.
6. Martin Timmerman, Bart Van Beneden, Lourent Uhres. RTOS Evaluation Kick Off! Real-Time Magazine. – 1998. – N3 pp 6 – 10

Рецензія/Peer review : 14.7.2014 р.

Надрукована/Printed : 16.7.2014 р.