

ОСОБЛИВОСТІ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПОЛЬОТНОГО КОНТРОЛЕРА

Розроблення програмного забезпечення (ПЗ) для безпілотного повітряного судна (БПС) є ключовим елементом, що визначає його функціональність, ефективність і безпеку. Сучасні завдання, покладені на безпілотні системи, вимагають від БПС високого ступеня автономності, здатності до аналізу даних у реальному часі та адаптації до змін навколишнього середовища. Без спеціалізованого ПЗ ці функції неможливо реалізувати, оскільки воно відповідає за керування обладнанням, опрацювання інформації з датчиків і камер, а також за ухвалення рішень на основі заданих алгоритмів [1].

Створення якісного ПЗ дає змогу інтегрувати БПС у складні екосистеми, взаємодіючи з іншими пристроями і забезпечуючи виконання специфічних завдань – від доставки вантажів до рятувальних операцій. Воно гарантує надійність і безпеку роботи, мінімізуючи вплив людського фактору, а також захищає БПС від збоїв. В умовах стрімкого зростання ринку і підвищених вимог до безпілотних систем, розробка ПЗ стає не просто необхідністю, а запорукою конкурентоспроможності.

Існує кілька підходів до розроблення ПЗ для вбудованих систем (firmware), кожен із яких застосовується залежно від вимог до системи та її завдань. Одним із поширених методів є використання скінченних автоматів. Цей підхід зручний для реалізації простих і передбачуваних систем із чітко визначеними станами та переходами між ними. Програмна модель скінченного автомата легко піддається тестуванню, що спрощує розробку та налагодження ПЗ. Однак цей підхід може бути обмежений у масштабованості, оскільки зі зростанням складності системи, управління станами стає менш прозорим.

Інший важливий підхід – застосування операційних систем реального часу (скор. англ. RTOS). Цей підхід доцільний для складніших систем, де потрібна висока точність виконання завдань із часовими обмеженнями. RTOS забезпечує механізм планування завдань, управління ресурсами та синхронізації, що дає змогу розробнику зосередитися на бізнес-логіці застосунку. Такий підхід спрощує масштабування і розширення системи, але вимагає додаткових ресурсів. Однак для таких систем як БПС, даний підхід більш доцільний [2].

Більшість ПЗ для вбудованих систем реалізується з використанням процедурного програмування мовою C (автопілот «Betaflight»). Є кілька для цього вагомих причин, однак основна причина – невеликі ресурси мікроконтролерів. Використання об'єктно-орієнтованого програмування (ООП) для складних систем доцільніше, тому для застосування цієї парадигми для вбудованих систем розроблено діалект мови програмування C++ – Embedded C++, що дає змогу усунути основний недолік. Однак ООП зосереджений на реалізації коду, тоді як вбудовані системи – на обробці даних. ООП настільки проникла в культуру розроблення застосунків, що коли розробляєш, важко уявити, крім об'єктів, щось ще. Класичне використання ООП не завжди ефективне для сучасних вимог високопродуктивного коду, де потрібна оптимізація пам'яті та обчислень.

Виходом є спільне використання ООП і дата-орієнтованого програмування (ДОП). ДОП – парадигма програмування, яка ставить у центр уваги організацію даних, що дає змогу підвищити ефективність роботи з пам'яттю та продуктивність ПЗ. Також це дає змогу спростити: масштабованість, паралельну обробку даних, реалізацію алгоритмів машинного навчання тощо.

Не менш важливим аспектом вбудованих систем є обробка переривань і подій. Подієво-орієнтоване програмування є ефективним підходом для розроблення firmware БПС, оскільки воно дає змогу системі реагувати на зміни в навколишньому середовищі та внутрішнього стану пристрою в реальному часі. У цьому підході основна увага приділяється подіям – сигналам, що надходять від сенсорів, оператора або інших підсистем.

В даному випадку основним шаблоном проектування firmware є Pub-Sub. Цей шаблон ідеально підходить для систем із високим ступенем асинхронності, таких як БПС. Застосування Pub-Sub дає змогу організувати взаємодію між різними підсистемами.

Список використаних джерел

1. Петросян А.Р., Граф М.С. Архітектура бортового комп'ютера безпілотного повітряного судна. Тези доповідей науково-практичної конференції, 24-25 листопада 2022 року. Київ : "Національний авіаційний університет", 2022. С.24-25.
2. Bhardwaj S. Real-time operating system for autonomous drone control. Innovative Research Thoughts. 2023. Vol. 9, no. 4. P. 134–143. URL: <https://doi.org/10.36676/irt.2023-v9i4-019>.