

УДК 004.41

Судомир В. - ст. гр. СІм-52

Тернопільський національний технічний університет імені Івана Пулюя

ВИКОРИСТАННЯ ФУНКЦІЙНОЇ ПАРАДИГМИ ПРИ ПРОГРАМУВАННІ МІКРОКОНТРОЛЕРІВ

Науковий керівник: к.т.н., доцент Луцків А. М.

Sudomyr V.

Ternopil Ivan Puluj National Technical University

USE OF FUNCTIONAL PARADIGM IN PROGRAMMING MICROCONTROLLERS

Supervisor: Candidate of technical sciences, docent Lutskiv A. M.

Ключові слова: функційна парадигма, мікроконтролер;

Keywords: functional paradigm, microcontroller.

У контексті бурхливого розвитку технології IoT дедалі більше створюється програм для мікроконтролерів.

На сьогодні при створенні програмного забезпечення, домінуючою є процедурна та іноді об'єктно об'єктно-орієнтована парадигми. Проте з урахуванням таких факторів як: робота мікроконтролерів у поєднанні з різними датчиками, необхідністю економії процесорного часу, написання легко читабельного коду та легкого і ефективного паралельного програмування для багатоядерних мікроконтролерів, оптимальним є використання функційної парадигми програмування.

Функційне програмування - це парадигма програмування, яка суттєво відрізняється від імперативної та об'єктно-орієнтованої парадигм програмування. Функційні програми, як правило, уникають станів та зміни даних, таких як встановлення змінної. Фактично, функційне програмування прагне використовувати підхід з одним ініціюванням тобто використовуються так звані незмінювані значення("immutable"). Функційна парадигма має багато корисних концепцій таких як : рекурсії та хвостові рекурсії, які замінюють звичайні цикли; чисті функції, в яких повернені данні залежать тільки від вхідних даних; функції вищих порядків, які можуть приймати в якості аргументів, а також повертати як результат обчислень інші функції; лінійні обчислення в яких значення не потрібно обчислювати, якщо вони не будуть використовуватися. Таким чином функційне програмування не тільки забезпечує кращу масштабованість компактного, читабельного та ефективного коду. Тому такий вид програмування успішно використовується в широкому спектрі застосувань, тому що використання мов високого рівня є більш ефективним використанням часу для створення П.З., але часто може бути менш ефективним у використанні обчислювальних ресурсів. Тому функційна парадигма не у всіх випадках буде ефективною для програмування мікроконтролерів.

На сьогоднішній день існують лише декілька інструментів для функційного програмування мікроконтролерів різними мовами програмування, одними з таких інструментів є Juniper та Copilot.

Juniper підтримує багато ознак типових для функційних мов програмування, включаючи алгебраїчні типи даних, записи, відповідність шаблону, незмінні структури даних, параметричні поліморфні функції і анонімні функції (lambda). Деякі імперативні концепції програмування також присутні в Juniper, наприклад, для циклів "while", "do while", можливість позначити змінні як "mutable" і змінні посилання.

Copilot – це предметно-орієнтована мова на Haskell, яка компілюється в embedded C. Copilot подібний до мов типу Luster. Copilot містить інтерпретатор, декілька back-end компіляторів та інші засоби.