

ОСОБЛИВОСТІ ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ ДЛЯ УПРАВЛІННЯ ТРАНСФОРМАТОРНИМ ЗАРЯДНИМ ПРИСТРОЄМ

Кошелюк М.В. – гр. БКІ-19, бакалавр, koshelmak@gmail.com

Стаценко Д.В. – к.т.н., доцент, statsenko.dv@knutd.com.ua

Київський національний університет технологій та дизайну

Метою роботи є вивчення можливостей різних мов програмування мікроконтролерів для розробки програмного забезпечення для управління трансформаторним зарядним пристроєм.

Програмування для мікроконтролерів, як і програмування для універсальних комп'ютерів пройшло великий шлях розвитку від програмування в машинних кодах до застосування сучасних інтегрованих систем написання програм, налагодження та програмування мікроконтролерів. В даний час вихідний текст програми пишеться на одній з мов програмування.

Процес перетворення операторів вихідної мови програмування в машинні коди мікропроцесора називається трансляцією вихідного тексту. В даний час ручна трансляція програм практично не використовується. Трансляція здійснюється спеціальними програмами- трансляторами.

Існує два великі класи програм-трансляторів: компілятори і інтерпретатори. При використанні компіляторів весь вихідний текст програми перетворюється в машинні коди, і саме ці коди записуються в пам'ять мікроконтролера. При використанні інтерпретатора в пам'ять мікроконтролера записується вихідний текст програми, а телеканал передається при зчитуванні чергового оператора. Природно, що швидкодія інтерпретаторів набагато нижче в порівнянні з компіляторами, в зв'язку з цим, при використанні оператора в циклі він транслюється багаторазово. Однак при програмуванні на мові високого рівня обсяг коду, який потрібно зберігати у внутрішній пам'яті може бути значно менше в порівнянні з виконуваним кодом. Ще однією перевагою застосування інтерпретаторів є легка переносимість програм з одного процесора на інший. С++ має більше можливостей для об'єктно-орієнтованого програмування, а також може використовувати шаблони та перевантаження операторів. Однак, ці можливості вимагають більше пам'яті та компіляторних операцій, що може призвести до погіршення продуктивності на мікроконтролерах з обмеженою кількістю ресурсів.

Мова програмування С є низькорівневою мовою, що дозволяє програмістам працювати з пам'яттю та апаратними ресурсами мікроконтролерів

Платформа: ІНФОРМАЦІЙНІ СИСТЕМИ. КОМП'ЮТЕРНІ СИСТЕМИ ТА МЕРЕЖІ. ТЕХНОЛОГІЇ INTERNET OF THINGS ТА SMART-СИСТЕМИ

без зайвих накладних витрат. Вона має простий та зрозумілий синтаксис, що дозволяє легко розуміти та змінювати код. Крім того, мова C має широку підтримку та доступність для багатьох платформ, що робить її популярним вибором для програмування мікроконтролерів.

Java та Python є високорівневими мовами програмування, які мають більш абстрактний підхід до програмування. Це дозволяє програмістам швидко розробляти складні програми та зменшує кількість помилок у коді. Однак, використання високорівневих мов програмування може призвести до витрати більше пам'яті та зниження швидкості виконання коду, особливо на мікроконтролерах з обмеженою кількістю ресурсів.

Assembler є найближчим до мови машинного коду та дозволяє програмістам працювати з кожним бітом і байтом мікроконтролера. Проте, використання Assembler може бути дуже складним та вимагати великої кількості часу для написання програми, що призводить до складності у підтримці та розширенні коду.

Висновок. В результаті проведеного дослідження мов програмування мікроконтролерів для управління трансформаторним зарядним пристроєм, було обрано мову програмування C. Ця мова є оптимальним вибором для програмування мікроконтролерів, оскільки вона надає програмістам повний контроль над використанням ресурсів та дозволяє ефективно використовувати обмежену кількість пам'яті. Крім того, вона має простий синтаксис та є достатньо ефективною, щоб забезпечити швидку роботу коду на мікроконтролерах.

Л і т е р а т у р а

1. Barr M. Programming embedded systems: With C and GNU development tools. 2-ге вид. Sebastopol, CA : O'Reilly, 2007. 301 с.
2. Gadre D. Programming and Customizing the AVR Microcontroller. 2-ге вид. McGraw-Hill/TAB Electronics, 2009. 576 с.
3. Gu C. Building Embedded Systems: Programmable Hardware. Apress, 2016. 322 с.
4. Norris D. Python for Microcontrollers: Getting Started with MicroPython and Pyboard. McGraw-Hill Education, 2016. 288 с.