

ОСОБЛИВОСТІ ПІДХОДУ ДО ВИВЧЕННЯ ТА ПРОГРАМУВАННЯ СУЧАСНИХ МІКРОКОНТРОЛЕРІВ

Розглянуто можливість удосконалення навчального процесу при вивченні сучасних мікроконтролерів та розробки прикладних програм для них.

Ключові слова: мікропроцесор, мікро-контролер, мова програмування високого та низького рівня, інтегральна схема, центральний процесор, комп'ютер, таймери, пристрої введення/виведення інформації, машинно-орієнтована мова програмування Асемблер, мова програмування високого рівня, прикладні програми, компілятори, апаратні і програмні засоби, сучасні технології навчання.

Постановка проблеми

Традиційна послідовність вивчення мов програмування студентами, які вивчають електроніку і системи керування (спеціальність автоматизація та комп'ютерно-інтегровані технології), є такою:

- на 1-му і 2-му курсах студенти вивчають мови програмування високого рівня, такі як Delphi і C (C++);

- на 3-му курсі студенти вивчають мікропроцесори (мікро-контролери) і машинно-орієнтовану мову низького рівня (Асемблер). При цьому не передбачається вивчення процедури використання мови програмування високого рівня C для програмування мікропроцесорів (мікро-контролерів).

Слід зазначити значне ускладнення сучасних апаратних і програмних засобів, що входять до складу сучасних комп'ютерних систем, зокрема мікро-контролерів, у зв'язку із збільшенням їх можливостей завдяки значному збільшенню рівня їх інтеграції (кількості елементарних логічних елементів на кристалі).

Завдяки ускладненню сучасних апаратних і програмних засобів комп'ютерних систем виникає також проблема, яка пов'язана з основним протиріччям вищих навчальних закладів (ВНЗ) – між об'ємом необхідних знань і часу на їх засвоєння. У зв'язку з наведеним необхідно шукати нові шляхи для підвищення якості навчання, в тому числі за рахунок впровадження активних методів навчання.

Аналіз останніх досліджень і публікацій

Будь-який спеціаліст у галузі комп'ютерних технологій доведе, що наведена послідовність вивчення мов програмування має суттєві недоліки, оскільки спочатку було б доцільно вивчити побудову відповідних мікропроцесорів (мікро-контролерів) разом з машинно-орієнтованою мовою програмування низького рівня, а потім вив-

чати мови програмування високого рівня (Delphi і C (C++)) на більш широкому понятійному рівні. Поняття мікропроцесор включає всі або більшість функцій, які виконуються центральним процесором, який розміщується на одній інтегральній схемі [1]. Поняття мікро-контролер відноситься до невеличкого комп'ютера, який розміщується на одній інтегральній схемі і включає відносно простий центральний процесор у комбінації з функціями, які його підтримують (такі як кварцовий генератор, таймери, сторожовий таймер, послідовні і аналогові пристрої введення / виведення інформації тощо) [2].

Для програмування перших мікропроцесорів (мікро-контролерів) в основному використовувалася машинно-орієнтована мова програмування Асемблер. Останнім часом були розроблені компілятори для використання мов програмування високого рівня (наприклад, C) для програмування мікро-контролерів [3].

Останнім часом спостерігається тенденція значного зростання можливостей сучасних мікро-контролерів і програмних засобів, які забезпечують їх роботу завдяки збільшенню рівня інтеграції (кількості елементарних логічних елементів на кристалі) [4].

Мета статті – удосконалення навчального процесу при вивченні сучасних мікро-контролерів та розробки прикладних програм з урахуванням можливостей їх апаратних і програмних засобів, а також сучасних технологій навчання.

Основні аспекти проблеми

Спочатку необхідно провести аналіз, наскільки наведена послідовність вивчення мов програмування і мікро-контролерів відповідає сучасним тенденціям зростання їх можливостей і програмних засобів, які забезпечують їх роботу?

Тенденції розвитку можливостей сучасних мікро-контролерів можна показати на прикладі 32-х розрядних мікро-контролерів PIC32MX3XX/4XX

виробництва фірми Microchip, США. Завдяки підвищенню рівня інтеграції з'явилася можливість використання новітнього центрального процесора (ЦП) MIPS32 M4K CPU (з 5-ти ступеневим 32-х розрядним конвеєрним ЦП) у складі сучасних мікро-контролерів, які отримали підвищені можливості щодо тактової частоти (до 80 МГц) і операційної швидкості обробки інформації, збільшення пам'яті (до 512К флеш-пам'яті, до 32К статичного ОЗУ), розширені можливості пристроїв введення / виведення інформації, 16 і 32-х розрядні команди, які оптимізуються для мов програмування високого рівня, таких як «С» тощо. Можна також показати складність процесу вивчення сучасних мікро-контролерів за допомогою наступного факту – повний опис сучасного 32-х розрядного мікро-контролера (його розмір складає приблизно 1x1 см) має понад 1100 аркушів англійською мовою [5, 6].

Вказані мікро-контролери разом з виконанням широкого кола загальних прикладних задач мають також значні апаратні та програмні засоби для виконання задач цифрової обробки інформації, наприклад цифрової фільтрації і перетворення вихідних сигналів датчиків тощо [7], що є важливим для студентів НАУ. Для ілюстрації важливості цього напрямку обробки інформації можна навести приклад літака Boeing 777 (який знаходиться в експлуатації з 1995р. і використовує 1280 бортових цифрових процесорів, причому вказані процесори використовують понад 4 млн. стрічок програмного коду) [8].

Що стосується відношення мов програмування високого та низького рівня слід зазначити, що мова програмування низького рівня забезпечує незначну (або нульову) абстракцію від структури системи команд. Поняття «низький рівень» відноситься до малої або несуттєвої абстракції між мовою програмування і машинною мовою; завдяки чому мова програмування низького рівня іноді описується як «близька до апаратних засобів».

Мова програмування низького рівня іноді розподіляється на 2 категорії: перше покоління (машинний код – мова, яку безпосередньо сприймає мікро-контролер, але ця мова потребує уваги до значної кількості подробиць його побудови) та друге покоління (мова Асемблер). Хоча мова Асемблер не відноситься до природної мови мікро-контролера, але програміст, який її використовує, повинен знати унікальну архітектуру мікропроцесора (такі його елементи, як регістри, команди тощо). Потім ці прості команди проходять процес компонування безпосередньо у машинний код. Можна розглядати мову Асемблер

як перший рівень абстракції по відношенню до машинного коду.

На відміну від мови програмування низького рівня мова програмування високого рівня має значний рівень абстракції від побудови комп'ютера, використовує конструкції натуральної мови, є більш зручною у використанні і більш компактною для різних платформ. Така мова ховає подробиці виконання команд ЦП, такі як методи доступу до пам'яті тощо [7].

Співвідношення мов програмування різного рівня можна показати на прикладі процесу перетворення програм мікро-контролера у машинний код. Практика використання мов програмування для розробки прикладних задач для процесорів цифрової обробки сигналів показує, що програми, які розроблені на мові Асемблер, виконуються скоріше, в той час як програми на мові С легше розробляти і налагоджувати.

Звичайні прикладні програми для персональних і універсальних ЕОМ виконуються, як правило, на мові С. Якщо мова Асемблер використовується взагалі, то її використання обмежується короткими підпрограмами, які повинні виконуватися з високою операційною швидкістю (рис.1), при цьому співвідношення програмістів, які використовують мови С і Асемблер, складає 10:1.

У зв'язку з тим, що прикладні програми для цифрових процесорів цифрової обробки сигналів значно коротше звичайних прикладних програм і з урахуванням важливого значення для них високої операційної швидкості кількість користувачів мов програмування С і Асемблер є приблизно рівними (рис. 2 б).

І лише урахування прибутків, отриманих від втілення програм для процесорів цифрової обробки сигналів, дає перевагу мові програмування Асемблер – на кожний долар прибутків, отриманих при використанні мови С випадає 2 долара прибутків, отриманих при використанні мови Асемблер (рис. 2 в) [11].

Таким чином, спостерігається тенденція зростання кількості програмістів, які використовують мову програмування С для розробки прикладних програм не тільки для звичайних персональних ЕОМ, а також для процесорів цифрової обробки сигналів, тобто мова програмування С поступово стає більш важливою у використанні у порівнянні з мовою програмування низького рівня Асемблер для студентів НАУ.

Компанія Microchip розробила програмний засіб MPLAB IDE, який використовує операційну систему Windows і який працює на персональних комп'ютерах для розробки прикладних програм для мікро-контролерів власного виробництва.

Він також має назву IDE (інтегроване середовище розробки прикладних програм), оскільки дозволяє розробляти і налагоджувати прикладні програми для вбудованих мікро-контролерів власного виробництва [12] на мовах програмування С і Асемблер (рис. 3).

Для ілюстрації зразків програм, які можна написати і налагодити за допомогою MPLAB IDE, наведені 2 програми на мовах С і Асемблер.

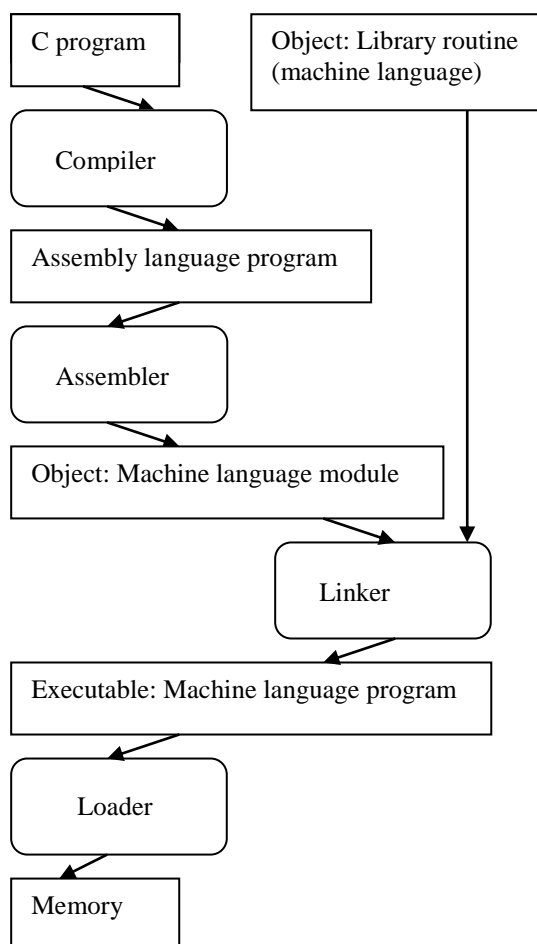


Рис. 1. Послідовність перетворення програм мікро-контролера у машинний код

Умовні позначки:

- С program – програма на мові програмування С;
- Compiler – компілятор (транслятор);
- Assembly language program – програма на мові програмування Асемблер;
- Object: Machine language module – об’єкт: модуль програми на машинній мові;
- Object: Library routine (machine language) – об’єкт: бібліотечний модуль (на машинній мові);
- Linker – програма-укладач (компоновщик);
- Executable: Machine language program – робоча програма (завантажувальний модуль);
- Loader – програма завантаження;
- Memory – пам’ять.

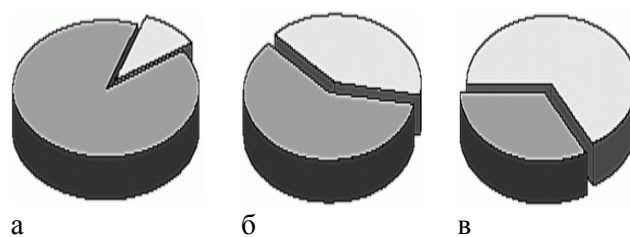


Рис. 2. Використання мов програмування С і Асемблер

- а) звичайними програмістами;
- б) програмістами, які розробляють програми для цифрових процесорів обробки сигналів;
- в) урахування доходу від втілення програм для цифрових процесорів обробки сигналів.

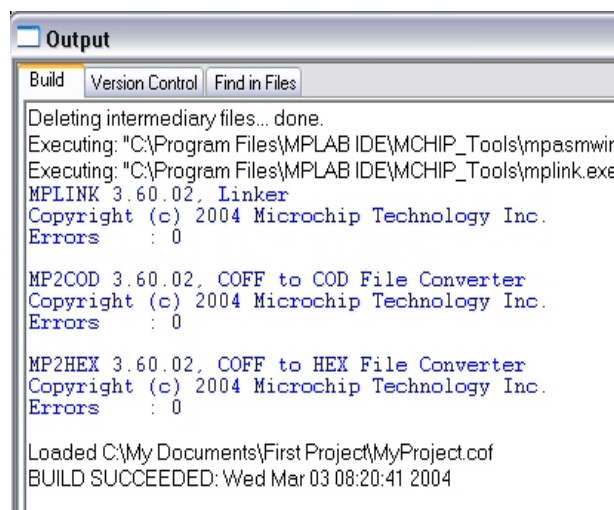


Рис. 3. Вигляд вихідного файла MPLAB IDE після успішного завершення налагодження прикладної програми

Перша програма розроблена на мові С і здійснює включення світло-діодів на спеціальній демонстраційній платі, яка входить до складу комплекту для розробки прикладних програм PICkit 1 Flash Starter Kit [13] (рис. 4).

Друга програма демонструє можливості команд мови Асемблер для введення і збереження інформації у регістрі WREG мікро-контролера PIC16F684 [12] (рис. 5).

Необхідно також провести аналіз можливих дидактичних засобів для пошуку шляхів підвищення якості навчального процесу у вказаному напрямку з урахуванням часових обмежень процесу навчання у вищих навчальних закладах.

По-перше необхідно визначити складові, які впливатиме на якість освіти, підготовки і перепідготовки спеціалістів, які використовують комп’ютерні технології. Сучасні літературні

джерела з педагогіки вищої школи, наприклад [15], дає такий перелік вказаних складових, який залежатиме від:	- якості освітніх програм;
- якості викладання (навчального процесу і педагогічної діяльності);	- якості матеріально-технічної бази, інформаційно-освітнього середовища;
- якості науково-педагогічних кадрів;	- якості студентів, учнів, абітурієнтів;
	- якості управління;
	- якості досліджень.

```
#include <pic.h>
__CONFIG(INTIO & WDTDIS & PWRTEN & MCLRDIS & UNPROTECT \
& UNPROTECT & BORDIS & IESODIS & FCMDIS);
main()
{
    PORTA = 0;
    CMCON0 = 7;          //Turn off Comparators (виключення компараторів)
    ANSEL = 0;          //Turn off ADC (виключення АЦП)
    TRISA = 0x49;       //Enable PORTA LED Outputs (включення виходів
                        //світло-діодів у PORTA)
    while(1 == 1)       //Loop Forever (нескінченний цикл)
    {
        if (0 == (PORTA & (1 << 3)))
            PORTA = '3'; //Turn on Four LEDs (включення 4-х світло-
                        //діодів)
        else
            PORTA = 4;   //Turn on Remaining Four LEDs (включення інших
                        //світло-діодів)
    }
}
```

Рис. 4. Програма розроблена на мові C здійснює включення світло-діодів на спеціальній демонстраційній платі, яка входить до складу комплекту для розробки прикладних програм PICkit 1 Flash Starter Kit

```
title "asmWREG - Demonstrate Loading and Saving WREG"
LIST R=DEC
INCLUDE "p16f684.inc"
PAGE
org 0
nop          ;Required for MPLAB ICD2 – необхідно для програматора
            ;MPLAB ICD2
movlw 123    ;Load WREG with Literal – записати у WREG літеру
            ;константу
clrw        ;Clear/Load WREG with 0 – очистити WREG
movlw 55     ;Load WREG with Decimal Literal - записати у WREG 10-ную
            ;константу
movlw 0x55   ;Load WREG with Hex Literal - записати у WREG 16-ную
            ;константу
movlw b'00110111' ;Load WREG with Binary Literal - записати у WREG 2-ную
            ;константу
movlw 'U'    ;Load WREG with ASCII Character - записати у WREG символ
            ;коду ASCII
movwf FSR    ;Move WREG to FSR – перенести дані із WREG у FSR
clrw        ;Clear WREG – очистити WREG
movf FSR, 0  ;Move FSR to WREG - перенести дані із FSR у WREG
movf FSR, 1  ;Store data in FSR – залишити дані у FSR
goto $      ;Finished, Everything Okay – успішне закінчення
end
```

Рис. 5. Програма команд мови Асемблер для введення і збереження інформації у регістрі WREG мікроконтролера PIC16F684

Для проведення аналізу можливих дидактичних засобів з метою підвищення якості навчального процесу необхідно врахувати особливості розвитку інформатики як наукової дисципліни, яка останнім часом дуже швидко розвивається. Тому модель інформатики як наукової дисципліни повинна враховувати її сучасний стан і динамічну складову її розвитку.

Дослідження також свідчать, що загальноприйнятого змісту інформатики як наукової, так й навчальної дисципліни на теперішній час не існує. Ця обставина призводить до значних ускладнень при розробці сучасних навчальних систем. Тому необхідно створити сучасну модель інформатики як предмету навчання, яка б включала результати, що отримані більшістю вчених і викладачів, а також використати об'єктно-орієнтований підхід для створення її складної моделі [16].

Для вирішення вказаної складної проблеми можна запропонувати використання Теорії рішення винахідницьких задач ТРВЗ (ТРИЗ - російською мовою) – технологію творчості, яка заснована на ідеї того, що винахідницька творчість пов'язана із змінами у техніці, яка розвивається за визначеними законами і що створення нових засобів труда повинно підкорятися об'єктивним закономірностям (незважаючи на суб'єктивне к ним відношення) [17].

Вказана технологія творчості дозволяє за умови її системного використання створити методологічну основу систематизації знань за окремими предметами ВНЗ, в основному технічним. Ми також повинні брати до уваги основне протиріччя ВНЗ, яке пов'язано із об'ємом знань і часу на їх засвоєння.

Технологія ТРВЗ використовується як засіб для формування творчого, сильного мислення, і, як засіб для передачі систематизованих, скомп'юнованих знань.

Для підвищення якості навчання пропонується ознайомити студентів перших курсів з дисципліною з умовною назвою «Розвиток творчого уявлення (РТУ)». Після вивчення цього курсу студенти повинні утворити стереотип діалектичного мислення (відповідний розділ ТРВЗ – РТУ). Крім того, студенти повинні вивчити закони розвитку технічних систем (ЗРТС).

Потім студенти повинні вивчати дисципліну з умовною назвою «Діалектика систем», причому зміст цієї дисципліни необхідно адаптувати під інформаційні технології - у нашому випадку для розвитку діалектичного мислення студентів.

Після засвоєння дисципліни «Діалектика систем» студенти повинні вивчати спеціальні технічні

дисципліни, навчальні програми яких перероблені згідно з ЗРТС. При цьому студенти повинні самостійно отримувати нові знання на основі попередніх і ЗРТС. Студенти також повинні навчатися використанню принципів ТРВЗ для рішення задач, які стояли перед вченими, розробниками техніки у відповідних галузях знань, тобто займаються активним навчанням. На цьому етапі навчання використовуються дидактичні можливості ТРВЗ з систематизації знань прикладних наук [18].

Висновки

Таким чином, наведена існуюча послідовність вивчення мов програмування високого рівня С(С++) не відповідає сучасним тенденціям розвитку мікро-контролерів, програмних засобів для розробки і налагодження прикладних програм для них, науковій методиці вивчення мов програмування різного рівня, сучасним активним технологіям навчання і вивчається окремо від вивчення мікро-контролерів і програмних засобів для розробки і налагодження прикладних програм для них. Така методика вивчення мов програмування, на думку автора, потребує значних витрат часу викладачів і студентів.

Зважаючи на наведене, для підвищення якості навчального процесу з урахуванням часових обмежень ВНЗ пропонується змінити послідовність вивчення мов програмування студентами, які вивчають електроніку і системи керування:

1. На першому етапі необхідно вивчити структуру сучасних мікро-контролерів і використання машинно-орієнтованої мови програмування Асемблер для розробки прикладних задач;
2. Водночас із вивченням мови Асемблер необхідно організувати вивчення мови програмування високого рівня С та її використання для програмування сучасних мікро-контролерів;
3. На другому етапі необхідно вивчити особливості побудови і використання мови програмування високого рівня С++ для рішення інженерних задач, а також інші мови програмування високого рівня.
4. Для скорочення часу на вивчення складних апаратних і програмних засобів сучасних мікро-контролерів пропонується застосувати нові активні методи навчання (технологію ТРВЗ).

Література

1. *An Introduction to Microcomputers. Volume 1. Basic Concepts* by Adam Osborne, 2nd Edition, Osborne-McGraw Hill, Berkely, California. ISBN 0-931988, 1980. – 486 p.

2. *Embedded Systems Dictionary* by Jack Ganssle and Mike Barr. CMP Books, 2003. – 291 p.
3. *MPLAB C Compiler For PIC32 MCUs. User's Guide*. DS51686B. Microchip Technology Inc. 2009. – 122 p.
4. *PIC32MX3XX/4XX Family Data Sheet*. 64/100-Pin General Purpose and USB 32-Bit Flash Microcontrollers. DS61143E. Microchip Technology Inc., 2008. – 1138 p.
5. *MIPS Technologies MIPS32 M4K Synthesizable Processor Core*. Berkely Design Technology, Inc., 2007. – 8 p.
6. *PIC32MX Family Reference Manual*. DS61132B. Microchip Technology Inc., 2009. – 1138 p.
7. *32-Bit Language Tools Libraries*. DS51685D. Microchip Technology Inc., 2009. – 150 p.
8. «*CrossTalk*». The Journal of Defense Software Engineering. The Critical Need for Software Engineering Education by Dr. Lyle N. Long, the Pennsylvania State University. Jan., 2008. – 6 p.
9. *Salomon David. Assemblers and Loaders* / David Salomon. Ellis Horwood Ltd. UK., 1993. – 294 p.
10. *Clark University. Computer Science Department. CSCI 140 Computer Organization* by prof. J. Breecher. 2004. – 35 p. : cs.clarku.edu/~jbreecher/public/comp_org/Lectures/PIC_ISA_Part2.ppt.
11. *The Scientist and Engineer's Guide to Digital Signal Processing* by [Stephen W. Smith](#), 2-nd Edition. California Technical Publishing, 2003. – 640 p.
12. *MPLAB® IDE User's Guide*. DS51519B. Microchip Technology Inc., 2006. – 288 p.
13. *PICkit™ 1 Flash Starter Kit. User's Guide*. DS40051D. Microchip Technology Inc., 2004. – 76 p.
14. «*123 PIC® MCU Experiments for the Evil Genius*» by Myke Predko. – 384 p.
15. *Андреев А. А.* Педагогика высшей школы. Новый курс / А. А. Андреев – М. : Московский международный институт эконометрики, информатики, финансов и права, 2002. – 264 с.
16. *Степанов А. Г.* Объектно-ориентированная модель информатики как предмета обучения : материалы конференции [«Развитие технической основы современных информационных образовательных технологий»] приоритетного национального проекта «Образование» ИТО, – 2006. – С. 3.
17. *Альтшуллер Г. С.* Творчество как точная наука / Г. С. Альтшуллер. – Петрозаводск : Скандинавия, 2004. – 208 с.
18. *Бердоносков В. Д.* Россия. Концепция внедрения ТРИЗ в высшую школу / В. Д. Бердоносков, А. Р. Куделько : материалы конференции ТРИЗ – 2007, Москва. – С. 3.
19. *Лонг Л.* Критична необхідність удосконалення освіти програмістів / Л. Лонг. Журнал програмного забезпечення для оборони «CrossTalk», Січень 2008. – С. 6.

В. М. Конюшко

ОСОБЕННОСТИ ПОХОДА К ИЗУЧЕНИЮ И ПРОГРАММИРОВАНИЮ СОВРЕМЕННЫХ МИКРОКОНТРОЛЛЕРОВ

РЕЗЮМЕ

Рассмотрена возможность усовершенствования учебного процесса при изучении современных микроконтроллеров и разработке прикладных программ для них.

Ключевые слова: микропроцессор, микро-контролер, язык программирования высокого и низкого уровней, интегральная схема, центральный процессор, комп'ютер, таймеры, устройства ввода / вывода информации, машинноориентированный язык программирования Асемблер, язык программирования высокого ряда, прикладные программы, компиляторы, аппаратные и программные средства, современные технологии обучения.

V. M. Konyushko

THE PECULIARITIES OF THE WAY TO STUDY AND PROGRAMMING OF MODERN MICROCONTROLLERS

SUMMARY

The possibility to improve the training process in order to study modern microcontrollers and to develop application programs for them.

