

*ГРУНТУ. Визначення хлороганічних пестицидів та поліхлорбіфенілів. Газово-хроматографічний метод з детектуванням захопленням електронів.(ISO 10382:2002,IDT) 4. Белова Н.А. Екологія, мікроморфологія, антропогенез лісних ґрунтів степової зони України. – Дніпропетровськ: Вид-во ДНУ, 1997. 264 с. 5. Походило Є.В., Столярчук П.Г. Способи імітансного контролю якості // Методи та прилади контролю якості. – 2003. – № 11. – С. 105–108. 6. Походило Є.В., Столярчук П.Г. Імітансний контроль якості продукції // Вісник Нац. ун-ту “Львівська політехніка”. – 2002. – №445. – С. 46–51.*

УДК 004.75

**Є.Я. Ваврук, М. Дутка**  
Національний університет “Львівська політехніка”  
кафедра електронних обчислювальних машин

## **ОРГАНІЗАЦІЯ ЗАВАНТАЖЕННЯ ПАМ'ЯТІ МІКРОКОНТРОЛЕРІВ У ГАРАНТОЗДАТНИХ СИСТЕМАХ КЕРУВАННЯ**

*Ї Ваврук Є., Дутка М., 2010*

**Проаналізовано особливості завантаження пам'яті мікроконтролерів, розроблені програмні засоби завантаження через робочі інтерфейси системи CAN та ETHERNET, проведено дослідження системи.**

**It is analysed features of microcontrollers' memory loading, software of loading via CAN and ETHERNET interfaces are developed, system research is carried out.**

**Вступ.** Для забезпечення гарантоздатної роботи систем керування, побудованих на базі мікроконтролерів (МК), важливою є можливість оперативної зміни даних в їх пам'яті. Залежно від топології системи, типів інтерфейсів та МК для цього використовуються різні програмні та апаратні засоби, здебільшого через інтерфейс JTAG. Проте такий підхід є неефективним у системах, до яких висувуються жорсткі вимоги щодо габаритних параметрів і умов експлуатації (наприклад, бортові системи наземного призначення, системи керування виробництвом тощо). Тому виникає необхідність пошуку інших шляхів завантаження пам'яті МК.

**Аналіз програмних засобів завантаження пам'яті мікроконтролерів.** Сьогодні найчастіше використовуються два основні принципи побудови засобів завантаження [1, 2]:

– із використанням однорівневої програми-завантажувача (ПЗ), яка попередньо записується до постійної пам'яті програм МК. ПЗ розпочинає своє виконання після запуску МК і обмінюється даними та командами запису чи читання із керуючою програмою (КП), розміщеною в процесорі керування (в загальному випадку ним може бути персональний комп'ютер – ПК). Після виконання програмою-завантажувачем запису необхідної програми до пам'яті МК керування передається цій програмі.

– із використанням дворівневої програми-завантажувача, яка складається із двох частин: первинної ПЗ та вторинної ПЗ. Первинна ПЗ попередньо записується в пам'ять МК. Первинна ПЗ розпочинає своє виконання після запуску МК. Основною функцією первинної ПЗ є завантаження вторинної ПЗ в оперативну пам'ять МК та виклик її процедур. Вторинна ПЗ – набір специфічних для МК процедур, які викликаються із первинної ПЗ, для реалізації процесів читання та запису даних в пам'ять програм, даних або конфігурації МК. Такий підхід забезпечує створення універсальніших засобів завантаження при використанні меншого об'єму пам'яті МК.

Одним із виробників МК, який пропонує готові рішення для завантаження програм, є фірма Microchip, у виробках якої пропонується завантаження через послідовні інтерфейси USART [3] та CAN [4]. В основу принципу роботи покладено використання однорівневої стартової ПЗ. Ця програма спочатку перевіряє останній байт у довгостроковій пам'яті даних, яка реалізована в вигляді репрограмованого запам'ятовуючого пристрою (РПЗП) МК і залежно від його значення передає керування основній виконавчій програмі (ОВП) чи продовжує своє виконання, очікуючи на команди запису чи читання через інтерфейс USART чи CAN. В обох випадках використовується мова програмування Assembler, що забезпечує компактний код ПЗ. Основний недолік наведених засобів завантаження полягає в тому, що після запису ОВП та скидання останнього байта в РПЗП даних неможливо знову запустити ПЗ для запису чи читання. В цьому випадку розробник ОВП повинен передбачити додаткову процедуру, яка б забезпечувала встановлення вищезгаданого байта при виконанні ОВП. Тобто ОВП повинна виконувати частину роботи засобів завантаження, що значно знижує універсальність. Дані ПЗ розроблені працівниками фірми Microchip для роботи із інтерфейсами USART та CAN на базі МК типу PIC18F.

**Постановка задачі.** Наведений вище аналіз вказує на існування проблеми завантаження пам'яті мікроконтролерів через стандартні робочі інтерфейси системи. Тому основною задачею є розроблення програмних засобів завантаження пам'яті через зовнішні інтерфейси системи CAN та Ethernet.

**Завантаження пам'яті мікроконтролерів.** Для визначення шляхів завантаження наведемо типову структурну схему базової системи керування (рис.1). До її складу входять: центральні процесори (ЦП<sub>1</sub>-ЦП<sub>N</sub>), периферійні компоненти – дві групи мікроконтролерів (МК<sub>11</sub>-МК<sub>1M</sub>, МК<sub>21</sub>-МК<sub>2L</sub>), група мережевих контролерів (МерК<sub>1</sub>-МерК<sub>M</sub>), адаптер – перетворювач інтерфейсів (ПІ), які з'єднані засобами наведених на схемі інтерфейсів. Як ЦП може використовуватись ПК.

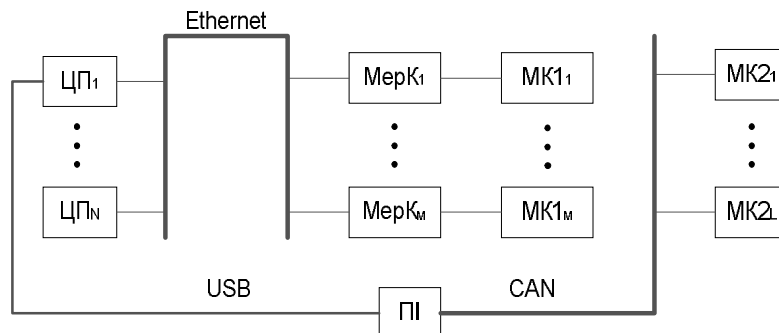


Рис. 1. Структурна схема системи

Один з ЦП керує периферійними компонентами та здійснює їх моніторинг. Вибір типів МК та топологій і організації роботи систем пов'язані між собою: необхідно, щоб МК підтримували обмін вибраними каналами зв'язку та забезпечували виконання основної роботи.

Тому для розроблення засобів завантаження програм в пам'ять МК ставляться певні вимоги до архітектури пам'яті самих МК та до наявності певної периферії для забезпечення зв'язку із керуючим ЦП. Система команд МК повинна містити команди стирання, запису та читання постійної пам'яті програм (флеш пам'ять чи РПЗП) із програмного режиму. Також накладається умова достатності об'єму для зберігання як ПЗ, так і ОВП, наявність комутаційних вузлів, що забезпечують роботу МК в системі, ефективність лінії проектування та ціна.

Враховуючи ці вимоги як МК<sub>21</sub> використовують МК типу PIC18F4585 фірми Microchip з вбудованим CAN модулем та послідовними портами типу MSSP, USART [5], як ПІ – універсальний адаптер USB – CAN типу GW-002-100 фірми SysTec [6], який поставляється разом із відповідними драйверами та бібліотеками функцій.

Оскільки МК із вбудованим Ethernet-модулем є відносно дорогими, доцільніше використовувати дешевший МК із мінімально необхідним набором вбудованих модулів для роботи конкрет-

ного компонента системи. Тоді для побудови системи необхідно використати зовнішній Ethernet контролер (Мерк на рис. 1), наприклад, Ethernet контролери серії Wxxxx фірми WIZnet. У системі використаний контролер типу W5100 [7] – однокристальний мережевий контролер з апаратною реалізацією всіх протоколів транспортного, сеансового і мережевого рівнів стека TCP/IP (TCP, IP, UDP, ICMP, ARP, DLC і MAC) і протоколом PPPoE (PPP over Ethernet)) з PAP/CHAP протоколами аутентифікації. Для зв'язку із МК передбачаються інтерфейси SPI та EMI.

Як МК1 в системі використовується МК типу C8052F120 фірми Silicon Laboratories [8] на базі ядра типу 8051 з інтерфейсом приєднання зовнішньої пам'яті EMI та послідовними портами типів SMBus/I2C, SPI, UART. Цей МК з'єднується із МерК через інтерфейс EMI.

### **Організація завантаження пам'яті**

У процедурі завантаження пам'яті беруть участь всі компоненти системи, що наведені на рис. 2, де КП – керуюча програма, АД – мережний адаптер, ЛЗ – лінія зв'язку, ІК – інтерфейсний контролер, ОВП – основна виконавча програма. ПЗ розміщується в пам'яті програм МК і починає виконуватись відразу після його пуску. На рис.2 пунктирною лінією обведені компоненти, які можуть бути зовнішніми відносно ЦП чи МК. Обмін даними та командами між КП і ОВП здійснюється за допомогою АД, ІК та ЛЗ.

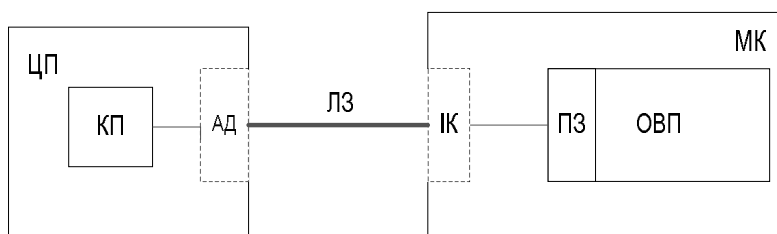


Рис. 2. Загальна структура завантажувача

Розроблення граф-схеми роботи програми-завантажувача. Граф-схему алгоритму наведено на рис. 3. Система завантаження програм та даних в МК складається із керуючої програми (розміщена в пам'яті ЦП) та ПЗ (розміщена в РПЗП програм МК). При запуску МК керування передається ПЗ. Після цього ПЗ очікує на команди від КП, опитуючи циклічно ІК (CAN чи Ethernet) про наявність вхідних даних. При отриманні команди ПЗ виконує її (запис, читання, стирання вказаних областей даних МК), при цьому обмінюючись даними із КП. Після завершення виконання чергової команди ПЗ очікує на наступну протягом певного часу. Якщо команда від КП не надходить, то ПЗ передає керування записаній ОВП.



Рис. 3. Граф-схема алгоритму роботи ПЗ

### ***Організація завантаження пам'яті МК типу C8051F120 по каналах Ethernet***

КП та ПЗ обмінюються блоками розміром 1024 байтів за протоколом UDP, де код команди записується в першому байті посилки. В конкретному випадку для з'єднання із МК визначаються (при початковому програмуванні його через JTAG) такі параметри: IP адреса, номер локального порту, коди команд запису/читання. Очікування команд та їх виконання відбуваються в циклі: спочатку, протягом 10 секунд, очікується посилка із командою від КП. Якщо команду прийнято, відбувається її виконання, після чого ПЗ знову очікує протягом певного часу на чергову посилку із командою, інакше керування передається записаній ОВП.

При виконанні будь-якої команди, після отримання пакета по каналу Ethernet від КП відсилається пакет-відповідь, в першому байті якого міститься одиниця. ПЗ підтримує виконання чотирьох команд, які зумовлені наявністю двох областей постійної пам'яті, а саме РПЗП даних та програм: читання з РПЗП даних, запис в РПЗП даних, читання з РПЗП програм, запис в РПЗП програм.

При виконанні команди читання з РПЗП даних у відповідь на посилку із командою відсилається посилка-відповідь, що містить в перших 256-ти байтах прочитані дані.

При виконанні команди запису в РПЗП даних спочатку у відповідь на пакет з командою відсилається пакет-відповідь. Після цього очікується пакет з даними для запису протягом 5 с. Якщо дані отримано, то вони записуються в РПЗП даних (крім останнього байту, який не змінюється), після чого відсилається пакет-відповідь з довільним вмістом.

При виконанні команди читання з РПЗП програм відразу після отримання пакета з командою відбувається циклічне відсилання вмісту відповідної чергової прочитаної сторінки РПЗП програм до КП та очікування посилки-відповіді. Якщо посилку-відповідь не отримано протягом 5 с, то процедура завершується, інакше ПЗ передає чергову посилку із вмістом чергової сторінки РПЗП програм. У разі отримання всіх необхідних посилок-відповідей виконується читання 122 сторінок РПЗП програм (записуються сторінки 0, 6...126, оскільки 1–5 зайняті ПЗ, а 127-ма зарезервована).

При виконанні команди запису в РПЗП програм спочатку у відповідь на пакет з командою від КП відсилається пакет-відповідь. Після цього відбувається циклічне очікування пакета з черговими даними, які, за умови отримання, записуються до відповідної сторінки РПЗП програм, після чого відсилається пакет-відповідь. Перед записом до першої сторінки РПЗП програм після отримання даних для запису в останні два байти РПЗП даних записуються другий та третій байти із отриманого пакета із даними. Ці байти визначають початкову адресу ОВП в першій команді безумовного переходу, яка виконується після пуску МК, код якої записаний у першому байті першого пакета. На місце адреси ОВП в другий та третій байти посилки записується початкова адреса ПЗ. Якщо пакет з даними для запису не отримано протягом 5 с, то процедура запису завершується, і керування передається головній функції ПЗ. У разі виникнення помилки при записі сторінок в останні байти РПЗП даних записується адреса ПЗ. Ці байти використовуються для визначення адреси переходу (передавання управління) за умови недержання команди від КП протягом часу очікування.

Розробляючи ОВП, слід врахувати наявність ПЗ у РПЗП програм МК на 1–5 сторінках.

### ***Організація завантаження пам'яті МК типів PIC18F2585/2680/4585/4680 по каналах CAN***

За основу взято запропоновану фірмою виробником вищезгаданих МК ПЗ по каналах CAN 2.0. Адресація пристроїв відбувається за допомогою старших 8-ми біт розширеного ідентифікатора повідомлення CAN, тому допускається використання в мережі до 256 МК, кожен з яких приймає повідомлення тільки із своєю адресою. КП та ПЗ обмінюються фреймами розміром 8 байт із використанням розширених ідентифікаторів, де код команди записується в два його молодші біти. В конкретному випадку для з'єднання із МК визначаються при початковому програмуванні такі параметри: адреса, відносний час очікування на команди.

Оскільки програма-завантажувач фірми виробника накладає певні обмеження на ОВП, необхідно змінити алгоритм запуску ПЗ та передачі управління ОВП відповідно до граф-схеми (див. рис. 3). При запуску МК управління передається спочатку ПЗ, яка очікує протягом певного

часу на надходження вхідних команд від КП. У випадку неодержання команд управління передається ОВП.

Очікування команд та їх виконання відбувається циклічно: спочатку протягом вказаного часу очікується надходження фрейму із командою. Якщо команду прийнято, вона виконується, після чого знову очікується надходження чергової команди. Якщо фрейм з командою не надійшов протягом вказаного часу, то керування передається ОВП.

При виконанні будь-якої команди відсилається фрейм-відповідь, в розширеному ідентифікаторі якого в старших 8-ми бітах розміщено адресу відправника, а у двох молодших визначається його вміст (00 після команди читання – вміст конфігураційних регістрів ПЗ, 01 після команди читання – дані, в всіх інших випадках невизначені дані). Запис в РПЗП програм відбувається блоками по 64 байти (8 фреймів), а в інші області пам'яті по 8 байтів (1 фрейм). Читання будь-якої області пам'яті відбувається блоками по 8 байтів.

ПЗ підтримує виконання 4-х різних команд, які передбачають різне трактування 8-байтного поля даних CAN фрейму:

- отримання вмісту регістрів конфігурації ЗВ (код 10b);
- отримання вмісту 8-байтної області даних за адресою, визначеною в конфігураційних регістрах ЗВ (код 11b);
- запис до конфігураційних регістрів ЗВ (код 00b);
- запис в 8-байтну область пам'яті МК за адресою, визначеною в конфігураційних регістрах ЗВ (код 01b).

У табл. 1 наведено формат повідомлення ПЗМК.

Таблиця 1

#### Формат повідомлення ПЗМК

Розширений ідентифікатор			Фрейм даних							
Адреса	--	Ко-манда	Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	Байт 6	Байт 7	Байт 8
AAAA AAAA	XXXX XX	00	ADDR L	ADDR H	ADDR U	RESVD	CTLBT	SPCMD	CPDTL	CPDTH
AAAA AAAA	XXXX XX	01	DATA0	DATA1	DATA2	DATA3	DATA4	DATA5	DATA6	DATA7
AAAA AAAA	XXXX XX	10	--	--	--	--	--	--	--	--
AAAA AAAA	XXXX XX	11	--	--	--	--	--	--	--	--

При виконанні операцій читання чи запису пам'яті МК типу PIC18F2585/2680/4585/4680 необхідно дотримуватись карти пам'яті згідно з табл. 2.

Таблиця 2

#### Карта пам'яті МК типу PIC18F2585/2680/4585/4680

Діапазон адрес	Тип та призначення
0x000 – 0x1FF	РПЗП завантажувача
0x200 – 0x2FFFFFF	РПЗП програм
0x300000 – 0x3FFFFFF	Конфігураційна пам'ять
0x400000 – 0xEFFFFFF	Область пам'яті, яка не використовується
0xF00000 – 0xFFFFFFF	РПЗП даних

У ПЗ визначено 8 програмних конфігураційних регістрів, які визначають режими його роботи. Доступ до цих регістрів відбувається при використанні команд запису та читання відповідних регістрів (коди команд XXXXXX10b та XXXXXX00b відповідно).

Регістри конфігурації:

- Address Low – молодший байт вказівника адреси.
- Address High – середній байт вказівника адреси.
- Address Upper – старший байт вказівника адреси.
- Reserved – призначений для вирішення проблеми розширення.
- Control Bits – контрольні біти, які визначають основні операції завантажувача. Їх опис наведено у табл. 3.

• Command – код команди завантажувача для виконання спеціальних функцій. Коди команди та їх опис наведено в табл. 4.

- Data A and Data B – зарезервовані для розширення.

При розробленні коду ОП із використанням спеціалізованого для даних МК середовища розроблення та відлагодження програм MPLAB фірми Microchip необхідно врахувати такі особливості:

– у перших 512-ти байтах пам'яті програм розміщується ПЗ, тому запис у цю область та читання не відбувається;

– вектори скиду та переривань зміщено на 512 байтів.

Таблиця 3

### Опис контрольних бітів

Біт	Назва	Опис	
0	WRITE_UNLOCK	0	Заборона запису (за замовчуванням)
		1	Дозвіл запису до будь-якої пам'яті
1	ERASE_ONLY	0	Дозвіл запису після стирання (за замовчуванням)
		1	Не записувати після стирання
2	AUTO_ERASE	0	Не стирати перед записом автоматично
		1	Стерти, якщо на 64-байтному кордоні, тоді записати (за замовчуванням)
3	AUTO_INC	0	Зміна вказівника вручну
		1	Збільшення вказівника автоматично після виконання команди (за замовчуванням)
4	ACK	0	Не відсилати підтвердження
		1	Відсилати порожній фрейм після кожної операції запису (за замовчуванням)

Таблиця 4

### Опис кодів команд

Команда	Код	Опис
NOP	00h	Немає операції
RESET	01h	Програмне скидання
INIT_CHK	02h	Ініціалізація внутрішньої контрольної суми та перевірка регістрів
CHK_RUN	03h	Перевірка контрольної суми
інші	04h – FFh	Не визначено

Тому необхідно змінити файли-скрипти (файли типу 18f????.lkr, наприклад 18f4585.lkr) лінкера. Потрібно додати рядок визначення сегмента *boot* та змінити адреси для сегментів *vectors* і *page*, як показано нижче.

```

...
CODEPAGE NAME=boot      START=0x0      END=0x1FF    PROTECTED
CODEPAGE NAME=vectors   START=0x200    END=0x229    PROTECTED
CODEPAGE NAME=page      START=0x22A    END=0x7FFF
...

```

Також необхідно змінити програму стандартного завантажувача ( c018.o або c018i.o). Для цього необхідно скопіювати файл c018.c або c018i.c із директорії ...mcc18\src\traditional\startup\ у директорію проекту та змінити рядок **#pragma code \_entry\_scn=0x000000** на **#pragma code \_entry\_scn=0x000200**. Після цього додати вказаний файл в проект, скопіювати його (у вікні списку файлів проекту натиснути праву кнопку миші на вищевказаному файлі і в меню вибрати пункт Compile), після чого вилучити файл із проекту. При написанні проекту потрібно явно вказати розміщення головної функції програми та у випадку використання обробки переривань також необхідно враховувати зміщення в 512 байтів. Так, запис обробника переривання для вектора вищого пріоритету матиме такий вигляд:

```
#pragma code InterruptVectorHigh = 0x208; //а не 0x008
Void InterruptVectorHigh (void){ ... }
```

Оголошення головної функції із явним зазначенням адреси розташування матиме такий вигляд:

```
#pragma code _s_Main_Section =0x000800 //Absolute adress of code section
void main (){... //program code}
```

Запропоновану методику завантаження пам'яті МК було опробовано під час розроблення системи комплексного тренажера літака Л-39 у складі: 6 ПК, 20 периферійних компонентів на базі МК типу PIC18F4585, які об'єднані мережею CAN, і 15 периферійних компонентів на базі МК типу C8051F120, об'єднаних у мережу Ethernet з використанням зовнішнього ІК типу W5100. Для організації доступу ПК до мережі CAN використовується універсальний адаптер USB – CAN типу GW-002-100 фірми SysTec.

Отримані результати наведено у табл. 5.

Таблиця 5

#### Результати проведення опробування

Параметр	Тип МК	
	C8051F120	PIC18F4585
Об'єм РПЗП програм МК, Кбайт	122*	47,5**
Час запису, с.	10	18
Час читання, с.	4	5
Розмір ПЗ, Кбайт	5	0,5

\* із 128 КБайт 5 КБайт зайняті ПЗ і 1 Кбайт – зарезервована область пам'яті

\*\* із 48 КБайт 0,5 Кбайт зайняті ПЗ

**Висновок.** Розроблені засоби завантаження та відповідні їм алгоритми роботи зводять до мінімуму залежність основної виконавчої програми від програми-завантажувача. У випадку завантажувача для мережі CAN розробник основної програми повинен тільки врахувати зміщення векторів переривання та основної функції, оскільки програма-завантажувач розміщується на місці стандартних векторів переривання і виконує перенапрявлення на нові, зміщені на 512 байтів. В випадку для мережі Ethernet програміст повинен врахувати лише наявність програми-завантажувача у відповідній області пам'яті програм.

1. <http://www.scienceprog.com/usb-bootloaders-for-avr-microcontrollers>. 2. *Martyn Gallop, Joanne McNamee. HCS12 Load RAM and Execute Bootloader User Guide. //Freescale Semiconductor. Application Note AN2546, Rev. 1, 12/2004.* 3. *Ross M. Fosler, Rodger Richey. A FLASH Bootloader for PIC16 and PIC18 Devices. // Microchip Technology Inc. Application Note AN85, Rev. 1 2002 .* 4. *Ross M. Fosler. A CAN Bootloader for PIC18F CAN Microcontrollers. // Microchip Technology Inc. Application Note AN247, Rev. 1 2003.* 5. *Microchip Technology Inc. PIC18F2585/2680/4585/4680 Data Sheet 2004.* 6. *USB-CAN modul. GW-001, GW-002, 3004006 Systems Manual. // SYS TEC electronic GmbH 2006.* 7. *WIZnet Co., Inc. W5100 Datasheet 2006.* 8. *Silicon Laboratories. C8051F120 - 127, C8051F130 - 133 Mixed Signal ISP FLASH MCU Family Datasheet 2004.*