

Кафедра автоматизації технологічних процесів та виробництв

Методичні вказівки для виконання лабораторної роботи №2 "Робота з пам'яттю EEPROM на програмному симуляторі PIC Simulator IDE" з курсу "Проектування мікропроцесорних систем керування технологічними процесами"

Тернопіль 2017

Методичні вказівки для виконання лабораторної роботи №2 «Робота з пам'яттю EEPROM на програмному симуляторі PIC Simulator IDE» з курсу «Проектування мікропроцесорних систем керування технологічними процесами».

Методичні вказівки розглянуті і схвалені кафедрою «Автоматизація технологічних процесів та виробництв», протокол № 4 від 21.11.2016 р.

Відповідальні за випуск

доцент, к.т.н. Медвідь В.Р., асистент Пісьціо В.П.

Лабораторна робота №2 Робота з пам'яттю EEPROM на програмному симуляторі PIC Simulator IDE

1. Робота з програмним симулятором PIC Simulator IDE

Запустивши на виконання PIC Simulator IDE, побачимо основне вікно цієї програми (рис. 1).

| ile Simulation R | ate T | ools (| Options | Help | | リ | | | | |
|---|--------------|-----------|----------|---------|---------|------------|--------------|-----------|--------------|------|
| Program Location | | C:\ | Program | Files\P | C Sim | ulator IDE | E\demo | 3.hex | 2 | _ |
| Microcontroller | PIC16F | 877A | 3 | | | | | | $\mathbf{}$ | |
| Last Instruction | | | <u> </u> | Ne | d Instr | uction - | | | | |
| GOTO | 0x002 | A | | | | DEC | FSZ O | 024,F | | 1 |
| Program Counter and | Working | g Registe | er | _ | Inst | tructions | Counter | | 262 | |
| | | | | | | k Cycles | Counte | er 🗌 | 1536 | |
| W Register | OD | | | | Re | al Time [| Duration | 3 | 84.00 | JS . |
| | | | | | | - | | | | |
| Special Function Reg | isters (S | FRs) | | 6 | | Genera | Purpos | se Hegist | ers (Gi | PRs) |
| Address and Name | Hex Value | 7 6 5 | 4 3 2 | 10 | 2 | Addr. | Hex Value | Addr. | Hex Value | 5 |
| 001h TMR0 | 00 | ГГГ | | | • | 020h | 02 | 030h | 00 | |
| 002h PCL | 2A | | | | 2 | 021h | 00 | 031h | 00 | 1 |
| 003h STATUS | 18 | ГСС | | | | 022h | 08 | 032h | 00 | |
| 004h FSR | 00 | | | | | 023h | 00 | 033h | 00 | |
| 005h PORTA | 00 | ГСС | | | | 024h | OC | 034h | 00 | |
| 006h PORTB | 00 | ГГГ | | | | 025h | 00 | 035h | 00 | |
| 007h PORTC | 00 | ГСС | | | | 026h | 00 | 036h | 00 | |
| 008h PORTD | 00 | ГСС | | | | 027h | 00 | 037h | 00 | |
| 009h PORTE | 00 | | | | | 028h | 00 | 038h | 00 | |
| 00Ah PCLATH | 00 | | | | | 029h | 00 | 039h | 00 | |
| 00Bh INTCON | 00 | | | | | 02Ah | 00 | 03Ah | 00 | |
| 00Ch PIR1 | 00 | | | | | 02Bh | 00 | 03Bh | 00 | |
| 00Dh PIR2 | 00 | | | | | 02Ch | 00 | 03Ch | 00 | |
| 00Eh TMR1L | 00 | | | | | 02Dh | 00 | 03Dh | 00 | |
| 00Fh TMR1H | 00 | ГГГ | | | | 02Eh | 00 | 03Eh | 00 | |
| and the second se | 00 | | | | - | 02Fb | 00 | 0355 | 00 | - |

Рис. 1. Основне вікно програми PIC Simulator IDE

У верхній частині знаходяться різні меню, через які можна отримати доступ до різних основних і додаткових модулів програми (на рис. 1 позначено як «1»).

Далі, в рядку Program Location вказано шлях до обраної програми і її ім'я (на рис. 1 - «2»).

Нижче, в рядку Microcontrollers, відображається тип обраного мікроконтролера (на рис. 1 - «3»).

У нижній частині вікна є дві панелі (позначені як «4» і «5»). У них відображаються стан програми, вміст спеціальних і керуючих регістрів обраного МК.

Послідовність роботи з програмним симулятором наступний:

- запуск програми PIC Simulator IDE;
- вибір типу мікроконтролера, для якого написана програма;

• вибір частоти кварцового генератора (впливає тільки на відображувані програмою дані про час виконання програми або команди, але не на швидкість роботи програми, що налагоджуються в PIC Simulator IDE);

• завантаження програми у вигляді НЕХ-файлу або запуск вбудованого компілятора мови assembler і написання в ньому потрібної програми;

- вибір потрібних модулів віртуальних пристроїв;
- вибір швидкості і режиму роботи програми симулятора;

• запуск процесу симуляції роботи програми на обраному МК.

Якщо потрібно скористатися для роботи з симулятором власною програмою або внести зміни у вже розроблену, необхідно створити або завантажити для цього файл асемблера, з якого після компіляції буде створений необхідний для роботи з симулятором hex-файл.

Для цього:

1. Натиснути Options | Assembler. Відкриється вікно компілятора Assembler – UNTITLED (рис. 2);



Рис. 2 Вікно симулятора з відкритим вікном Assembler

2. У вікні Assembler натисніть опцію File. Розкриється закладка (рис. 3), з якої для створення нового файлу потрібно натиснути New, а для завантаження вже створеного – OPEN.

| Pressentiation Rate Tools Options Program Location C:\Program Files\PIC Simulator IDE\rb0int.hex New Ctrl+N Microcontroller PIC16F84 Clock Frequency 4.0 MHz Save Ctrl+O Last Instruction Next Instruction Save As Print Source Ctrl+P Program Counter and W Register Instructions Counter 0 Print Listing PC 0000 Real Time Duration 0.00 µs | |
|--|-------------|
| Program Location C:\Program Files\PIC Simulator IDE \rb0int.hex New Ctrl+N Microcontroller PIC16F84 Clock Frequency 4.0 MHz Save Ctrl+O Last Instruction Next Instruction Next Instruction Save As Print Source Ctrl+P Program Counter and W Register Instructions Counter 0 Clock Cycles Counter 0 W Register 00 Real Time Duration 0.00 µs 0 0 | |
| Microcontroller PIC16F84 Clock Frequency 4.0 MHz Last Instruction Next Instruction Save Ctrl+S Program Counter and W Register Instructions Counter 0 PC 0000 Clock Cycles Counter 0 W Register 00 Real Time Duration 0.00 µs | - |
| Last Instruction Next Instruction Save As Program Counter and W Register Instructions Counter 0 PC 0000 Clock Cycles Counter 0 W Register 00 Real Time Duration 0.00 µs | |
| Program Counter and W Register Instructions Counter 0 PC 0000 Clock Cycles Counter 0 W Register 00 Real Time Duration 0.00 µs | |
| PC 0000 W Register 00 Real Time Duration 0.00 µs | |
| W Register 00 Real Time Duration 0.00 µs | |
| | |
| Special Function Registers (SFRs) General Purpose Registers (GPRs) | |
| Address and Name Value 7 6 5 4 3 2 1 0 Addr. Value Addr. Value | |
| 001h TMR0 00 00 00Ch 00 01Ch 00 • | - |
| | |
| 002h STATUS 18 00Eh 00 01Eh 00 Lin 1.Col 0 Num | of lines: 1 |
| | |
| | - |
| | |
| | |
| | |
| | - |
| 081h OPTION REG FF 016h 00 026h 00 4 | + |
| 085h TRISA 1F 017h 00 027h 00 Lin 1, Col 0 Num | of lines: 1 |
| 086h TRISB FF 018h 00 028h 00 | |
| 088h EECON1 00 019h 00 029h 00 | |
| TMR0 Prescaler 00 01Ah 00 02Ah 00 | |
| 01Bh 00 028h 00 • | |

Рис. 3

3. Після вибору і завантаження файлу (наприклад, rb0int.asm), його текст з'явиться в верхній половині вікна Assembler (рис. 4).

| S PIC Simulator IDE | | | | | - | | S Ass | embler - rb0int.a | asm | | |
|---|-----------------------|---------------|----------|------------|-------------------|------|-----------|-------------------|--------------------------------|-----|-----------------|
| | | | | | | | File E | dit Tools Option | IS | | |
| Program Location | C:\Program Files\P | IC Simulat | or IDE\r | rb0int.hex | | | 0001 | ; Begin | | | - |
| Microcontroller | PIC16F84 Clock Freque | ency | 4.0 M | lHz | | | 0002 | ROL | EQU OxC | | |
| Last Instruction | Ne | et Instructio | on | | | | 0003 | ROH | EQU OxD | | |
| | | | | | | - | 0004 | R1L | EQU UXE | | |
| - | | | | | | | 0005 | R1H R2L | FOU OX10 | | |
| Program Counter and | W Register | Instruct | tions Co | unter | 0 | | 0007 | R2H | EQU Ox11 | | |
| PC 0000 | | Clock C | ycles Co | ounter | 0 | | 0008 | R3L | EQU Ox12 | | |
| W Register | | Beal T | ime Dur | ation | 0.001 | 15 | 0009 | R3H | EQU Ox13 | | |
| , | , | , | | | , | | 0010 | R4L | EQU Ox14 | | |
| - Special Function Regi | isters (SFRs) | Ge | eneral P | urpose Re | gisters (G | PRs) | 0011 | R4H | EQU Ox15 | | |
| | Hex Binary Value | | н | lex | Hex | | 0012 | R5L | EQU Ox16 | | |
| Address and Name | Value 76543210 | Ac | ddr. Va | alue Ado | r. Value | • | 0013 | R5H | EQU UX17 | | |
| 001h TMR0 | | | OCh 0 | 010 | h 00 | | 0014 | W_1. STA | LMP LQU UX4F Tus TEMP Fou o | VAF | |
| 002h PCL | | 00 | DDh 0 | 010 | h 00 | | 0015 | JIA | TOS_TEMP EQU O | 112 | · · |
| 003h STATUS | 18 | 00 | OEh 0 | 011 | h 00 | | Lin 1 Co | 10 | | N | um of lines: 60 |
| 004h FSR | | 00 | OFh C | | h 00 | | | | | | |
| UUSh PURTA | | 0 | 100 U | | n 00 | | 0001 | | | | • |
| 008h FEDATA | | 0 | 12h | | h 00 | | | | | | |
| 009h EEADB | | 0 | 13h 0 | | h 00 | | | | | | |
| 00Ah PCLATH | | 0 | 14h 0 | 00 024 | h 00 | | | | | | |
| 00Bh INTCON | | 0 | 15h 0 | 02 | h 00 | | | | | | • |
| 081h OPTION_REG | FF FF | 0 | 16h 🛛 🗘 | 02 | <mark>h</mark> 00 | | | 4 | | | ÷ |
| 085h TRISA | 1F | 0 | 17h (| 00 02 | h 00 | | Lin 1, Co | 010 | | 1 | Num of lines: 1 |
| 086h TRISB | FF P | 0 | 18h 0 | 00 02 | h 00 | | | | | | |
| U88h EECON1 | | 0 | 19h 0 | 02 | h 00 | | | | | | |
| I MHU Prescaler | | 0 | IAh L | | h 00 | | | | | | |
| | | | | 0 021 | 1 00 | _ | | | | | |

Рис. 4 Завантаження файлу rb0int.asm

4. Для компіляції створеного або завантаженого і потім зміненого файлу, натисніть Tools і у вікні, що розкриється – Assemle. В нижній половині вікна assembler з'явиться відкомпільований файл і одночасно, при відсутності помилок, буде створений одноіменний hex-файл.

| S PIC Simulator IDE | | | | 🗵 😒 Assembler - rb0int.asm | 🛛 |
|--|-----------------------------------|--|-------------------------------|--|--------------|
| | | | | File Edit Tools Options | |
| Program Location C:\Program Files\PIC Simulator IDE \rb0int.hex Microcontroller PIC16F84 Clock Frequency 4.0 MHz | | | hex | 0001 ; Assemble F7 0002 Assemble & Load F8 Assemble With MPASMW/IN F9 | Ē |
| Last Instruction | Nex | t Instruction | | 0004 R1L EQU 0xE 0005 R1H EQU 0xF 0005 R2L EQU 0xF | |
| Program Counter and Counter an | | Instructions Counter Clock Cycles Counter | 0 | 0007 R2H EQU 0x11 0008 R3L EQU 0x12 | |
| W Register | sters (SFBs) | Real Time Duration | 0.00 µs e Registers (GPRs) | 0009 R3H EQU 0x13 0010 R4L EQU 0x14 a) 0011 R4H EQU 0x15 | |
| Address and Name | Hex BinanyValue Value 76543210 | Hex Addr. Value | Hex Addr. Value | 0012 R5L EQU 0x16 0013 R5H EQU 0x17 0014 R5H EQU 0x17 | |
| 001h TMR0 002h PCL | | 00Ch 00 00Dh 00 | 01Ch 00 01Dh 00 01Ch 00 | 0015 STATUS_TEMP EQU 0x4E | • |
| 005h STATOS 004h FSR 005h PORTA | | 00Fh 00 010h 00 | 01Fh 00 020h 00 | Lin 1, Col 0 Num (| of lines: 60 |
| 006h PORTB 008h EEDATA | | 011h 00 012h 00 | 021h 00 022h 00 | 0001 0002 PIC ASSEMBLER LISTING | |
| 009h EEADR 00Ah PCLATH 00Bh INTCON | | 013h 00 014h 00 015h 00 | 023h 00 024h 00 025h 00 | 0003 Line Adress Optice Instruction 0004 0005 0001 0000 ; Begin | |
| 081h OPTION_REG 085h TRISA | FF 1F | 016h 00 017h 00 | 026h 00 027h 00 | Lin 5, Col 0 Num | of lines: 66 |
| 086h THISB 088h EECON1 TMR0 Prescaler | | 019h 00 019h 00 01Ah 00 | 028h 00 029h 00 02Ah 00 | | |
| | | 01Bh 00 | 02Bh 00 🔄 | | |

2. Завдання на лабораторну роботу «Робота з пам'яттю EEPROM PIC – контролера».

1. Вивчити програмну модель PIC Simulator IDE.

2. Вивчити команди обміну даними з пам'яттю EEPROM PIC – контролера.

3. Написати і дослідити роботу програми з Прикладу 1 та дослідити вміст регістрів контролера, які використовуються при виконанні цієї програми.

4. Користуючись вікном "Program Memory Editor" або " PIC Disassembler" записати перші 7 команд виконуваної програми на Асемблері та в шістнадцяткових кодах.

5. Записати для вибраних команд коментар щодо їх призначення (див. Приклад 2).

Приклад 1

Цей приклад заповнює всю EEPROM різними значеннями і вводить МК в нескінченний цикл.

Текст програми з файлу eeprom.asm має наступний вигляд:

; Begin

R0L EQU 0xC R0H EQU 0xD R1L EQU 0xE R1H EQU 0xF R2L EQU 0x10 R2H EOU 0x11 R3L EQU 0x12 R3H EQU 0x13 R4L EOU 0x14 R4H EQU 0x15 R5L EQU 0x16 R5H EQU 0x17 ORG 0x0000 BCF PCLATH,3 BCF PCLATH,4 GOTO L0001 ORG 0x0004 RETFIE L0001: ; 1: Dim a As Byte 'eeprom address The address of 'a' is 0x18 a EQU 0x18 ; 2: Dim b As Byte 'eeprom data The address of 'b' is 0x19 b EQU 0x19 : 3: ; 4: For a = 0 To 63 'go through whole eeprom memory CLRF 0x18 L0002: MOVF 0x18,W SUBLW 0x3F BTFSS STATUS.C **GOTO L0003** ; 5: b = 255 - a 'set the data value to be written MOVF 0x18,W SUBLW 0xFF MOVWF 0x19 ; 6: Write a, b 'perform the writing to eeprom

MOVF 0x18,W MOVWF EEADR MOVF 0x19,W MOVWF EEDATA BSF STATUS, RP0 BSF EECON1, WREN MOVLW 0x55 **MOVWF EECON2** MOVLW 0xAA MOVWF EECON2 BSF EECON1,WR L0004: BTFSC EECON1.WR **GOTO L0004** BCF EECON1, WREN BCF EECON1, EEIF BCF STATUS, RP0 ; 7: Next a MOVLW 0x01 ADDWF 0x18,F BTFSS STATUS,C GOTO L0002 L0003: MOVLW 0x1F ANDWF STATUS,F ; End of program L0005:GOTO L0005 ; End of listing **END**

3. Послідовність роботи з стимулятором при виконанні програми

Завантажимо програму в PIC Simulator IDE, виконавши наступне:

1. Запустити PIC Simulator IDE;

2. Натиснути Options | Select Microcontroller;

3. Вибрати PIC16F84 і натиснути кнопку Select;

4. Натиснути File | Load Program;

5. Вибрати файл eeprom.hex і натиснути Open (програмне забезпечення завантажено);

6. Натиснути Tools | EEPROM (редактор EEPROM; відкриється вікно Memory Editor EEPROM);

7. Натиснути Tools | PIC Diosassembler. Відкриється вікно з дизасембльованою програмою у вигляді кодів та команд асемблера.

8. Натиснути Options | Change (час запису EEPROM);

9. Ввести нове значення 100 і натиснути ОК (треба бути уважними при виборі цього значення, тому що дійсне значення - приблизно 20 000 тактових циклів при частоті тактового генератора 4 МГц; в цьому прикладі ми використовуємо зменшене значення - це значно прискорить процес моделювання, але не порушить роботу програми);

10. Вибрати Rate|Extremely Fast sim ulation rate;

11. Натиснути Options Infinite Loop Stops Simulation;

12. Натиснути Simulation|Start (почнеться моделювання).

Програма заповнить пам'ять EEPROM даними і введе МК в нескінченний цикл, після виявлення якого PIC Simulator IDE автоматично зупинить роботу.

Вигляд екрану з виконуваною програмою показаний на рис. 6.

| S PIC Simulator IDE | | | | | | | | |
|---|---|--|--|---|---|-------------------|---|--|
| | | | File Option | าร | | | | |
| Program Location Microcontroller Last Instruction GOTO Program Counter and PC 0023 W Register Special Function Reg Address and Name 001h TMR0 002h PCL 003h STATUS 004h FSR 005h PORTA 005h TRISA 065h TRISA 065h TRISA | C:\Program Files\ PIC16F84 Clock Free Used States (SFRs) Hex Binary Value Value 7 6 5 4 3 2 1 0 00 23 1A 00 00 00 00 00 00 00 00 00 00 00 00 00 | PIC Simulator IDE \eeprom. juency 4.0 MHz lext Instruction GOTO 0x0 Instructions Counter Clock Cycles Counter Real Time Duration General Purpose Hex Addr. Value 00Ch 00 00Eh 00 00Eh 00 00Eh 00 00Eh 00 00Fh 00 01Dh 00 01Dh 00 01Dh 00 01Dh 00 01Dh 00 01Bh 00 01Bh 00 01Bh 00 01Bh 00 | hex 23 5835 32572 8143.00 µs Registers (GPRs) Addr. Value 01Ch 00 01Eh 00 01Eh 00 022h 00 022h 00 022h 00 022h 00 022h 00 022h 00 025h 00 00 00 00 00 00 00 00 00 00 | PIC DISJ Address 0000 0001 0002 0003 0004 0005 0006 0007 0008 0007 0008 0007 0008 0007 0008 0007 0008 0007 0008 0007 0008 0007 0008 0000 0000 0000 0000 0000 0000 0000 0000 | ASSEMBLE Opcode 118Å 120Å 2805 0000 0198 0818 3C3F 1C03 2821 0818 3CFF 0099 0818 3CFF 0099 0818 089 0819 0088 1683 1508 3055 0089 30ÅÅ 0089 1488 1888 | L1: L4: L3: | G Tion BCF PCLATH,3 BCF PCLATH,4 GOTO L1 NOP RETFIE CLRF 0x18,W SUBLW 0x3F BTFSS STATUS,C GOTO L2 MOVF 0x18,W SUBLW 0xFF MOVF 0x18,W SUBLW 0xFF MOVF 0x18,W MOVWF 0x19 MOVF 0x19,W MOVWF 0x09 BSF STATUS,RPO BSF 0x08,2 MOVLW 0x55 MOVUF 0x09 MOVLW 0xAA MOVWF 0x09 BSF 0x08,1 BTFSC 0x08,1 | |
| S FEDROM Mersons Editor | | | | | | | | |
| 0 1 2 00h FF FE FD 10h EF EE ED 20h DF DE DD 30h CF CE CD ✓ Always On Top | 3 4 5 6 7 8 FC FB FA F9 F8 F7 EC EB EA E9 E8 E7 DC DB DA D9 D8 D7 CC CB CA C9 C8 C7 | 9 A B C D 1 F6 F5 F4 F3 F2 F E6 E5 E4 E3 E2 E D6 D5 D4 D3 D2 D C6 C5 C4 C3 C2 C Clear C | E F 1 F0 1 E0 1 D0 1 C0 | | | | | |

Рис. 6 Форма екрану з виконуваною програмою «Робота з EEPROM»

З вікна "PIC Disassembler" вибираємо сім перших команд з їх шістнадцятковими кодами і знаходимо з таблиці кодів асемблера PIC-контролера коментар щодо призначення цих команд (див. Приклад 2, де наведено такий запис для однієї команди).

| Приклад 2 | | |
|-------------|---------------|--|
| Код команди | Команда | Виконувана операція (коментар) |
| 118A | BCF PCLATH, 3 | ; скинути в "0" 3-ій біт регістра РСLАТН |
| і т.д. | | |

Вміст регістрів контролера, які використовуються при виконанні програми, знаходимо з області регістрів Adress and Name, яка розташована в лівій нижній частині основного вікна симулятора (виділені рожевим кольором). Всі регістри возьмирозрядні.

В процесі виконання програми по зміні кольору комірок видно, вміст яких регістрів змінюється. Забарвлення комірки відповідного розряду прегістра померанчевим кольором означає наявність "1", білим - "0". Вміст регістрів записуємо в шістнадцятковому коді за Прикладом 3.

| Приклад 3. | |
|------------|----------------|
| Регістр | Вміст регістра |
| PORTA | 1F |

і т.д.

4. Контрольні запитання

1. Структура пам'яті EEPROM.

2. Структура блоку регістрів мікроконтролера.

3. Формат регістра STATUS.

4. Призначення та позначення основних елементів програмної моделі мікроконтролера.

5. Література

1. Данилин А. Программа-симулятор PIC Simulator IDE / Данилин А. // Современная электроника. 2006.- №4. -С. 68-76.

2. Тавернье К. РІС-микроконтроллеры. Практика применения. М.: ДМК, 2002.

3. Предко М. Создайте робота своими руками на РІС- контроллере./ Майкл Предко; Пер. с английского Земского Ю.В. – М.: ДМК Пресс, 2006. – 408 с.: ил. – (В помощь радиолюбителю).

4. Кениг А. и М. Полное руководство по РІС-микроконтроллерам.: Пер. с нем.-К.: "МК-Пресс", 2007.-256 с., ил.