

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет
імені Івана Пулюя

Кафедра автоматизації
технологічних
процесів та виробництв

Методичні вказівки
для виконання лабораторної роботи №6 “Приєм і
передача даних з «апаратного» UART і відображення
даних у вікні симуляції апаратного інтерфейсу UART
програмного симулятора PIC Simulator IDE”
з курсу “Проектування мікропроцесорних систем
керування технологічними процесами”

Тернопіль 2017

Методичні вказівки для виконання лабораторної роботи №6 «Приєм і передача даних з «апаратного» UART і відображення даних у вікні симуляції апаратного інтерфейсу UART програмного симулятора PIC Simulator IDE» з курсу «Проектування мікропроцесорних систем керування технологічними процесами».

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

Відповідальні за випуск

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

Лабораторна робота №6

Приєм і передача даних з «апаратного» UART і відображення даних у вікні симуляції апаратного інтерфейсу UART програмного симулятора PIC Simulator IDE

1. Робота з програмним симулятором PIC Simulator IDE

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

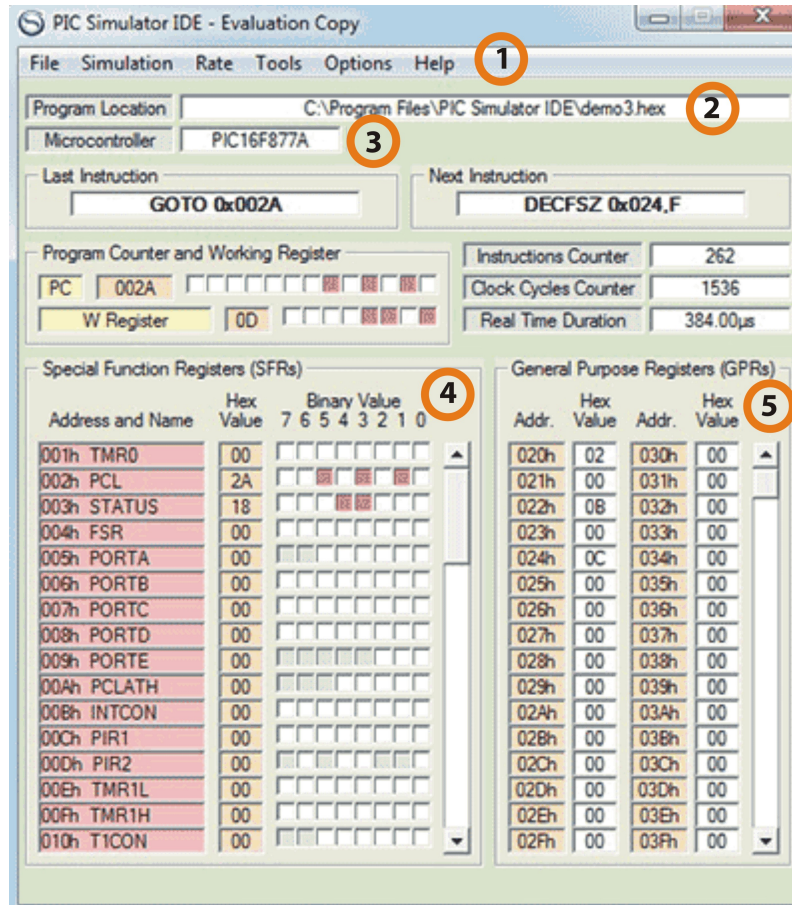


Рис. 1. Основне вікно програми PIC Simulator IDE

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 HEX-файлу або запуск вбудованого компілятора мови асемблер 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);

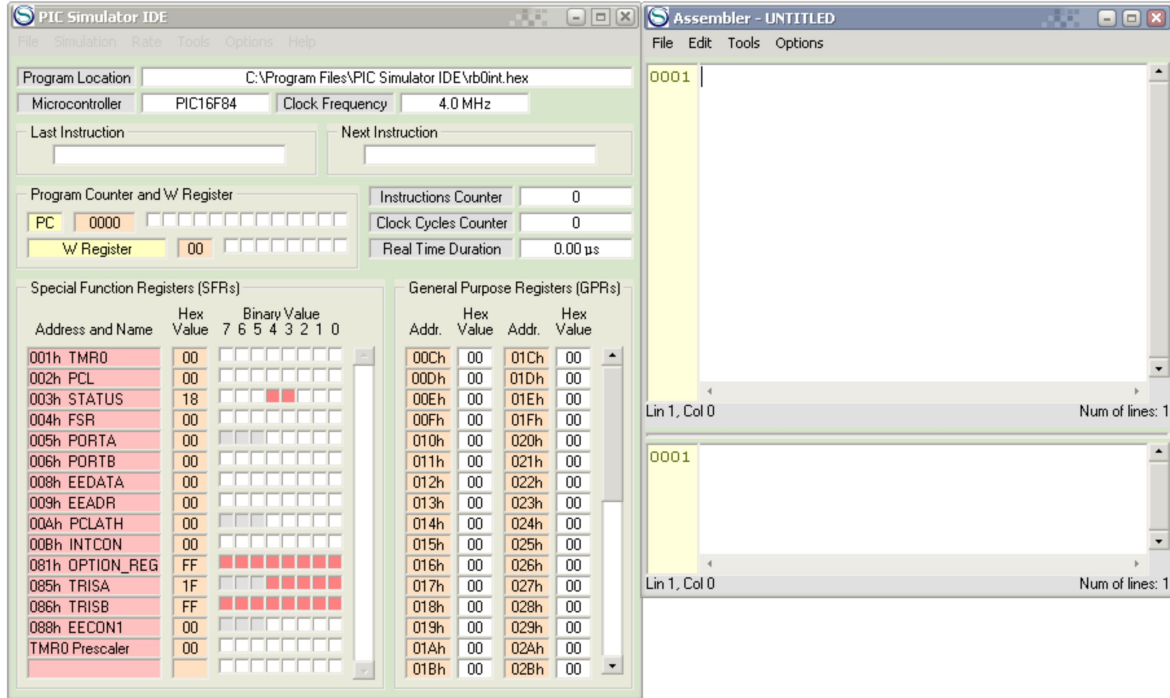


Рис. 2 Вікно симулятора з відкритим вікном Assembler

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.

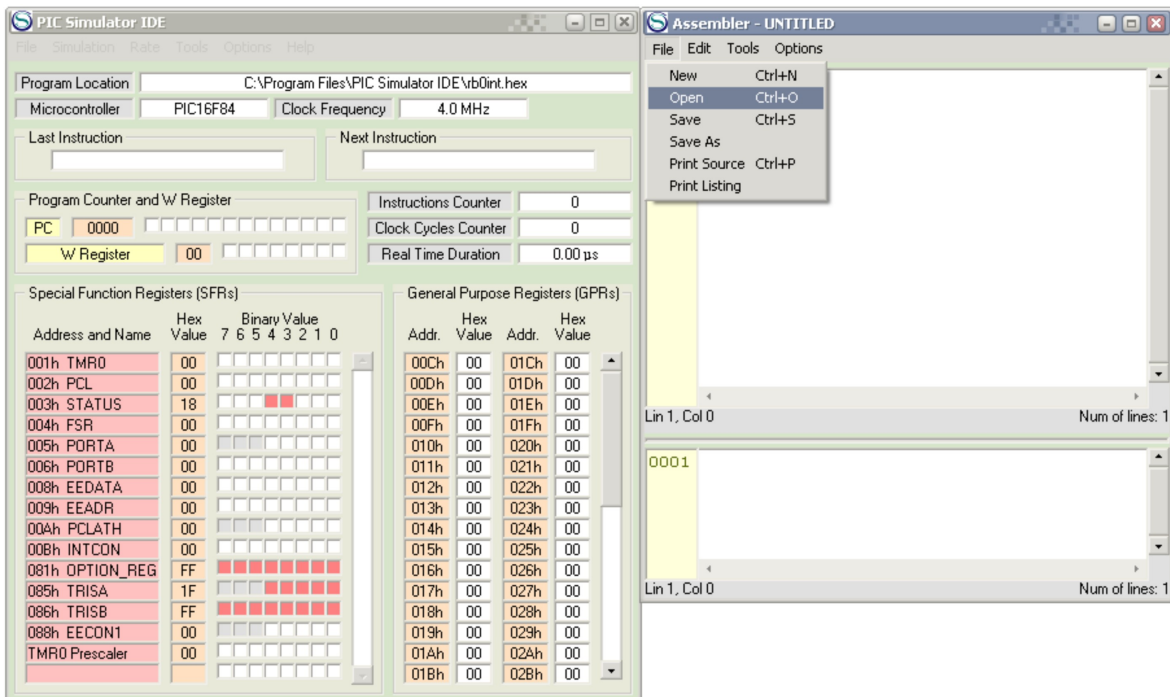


Рис. 3

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

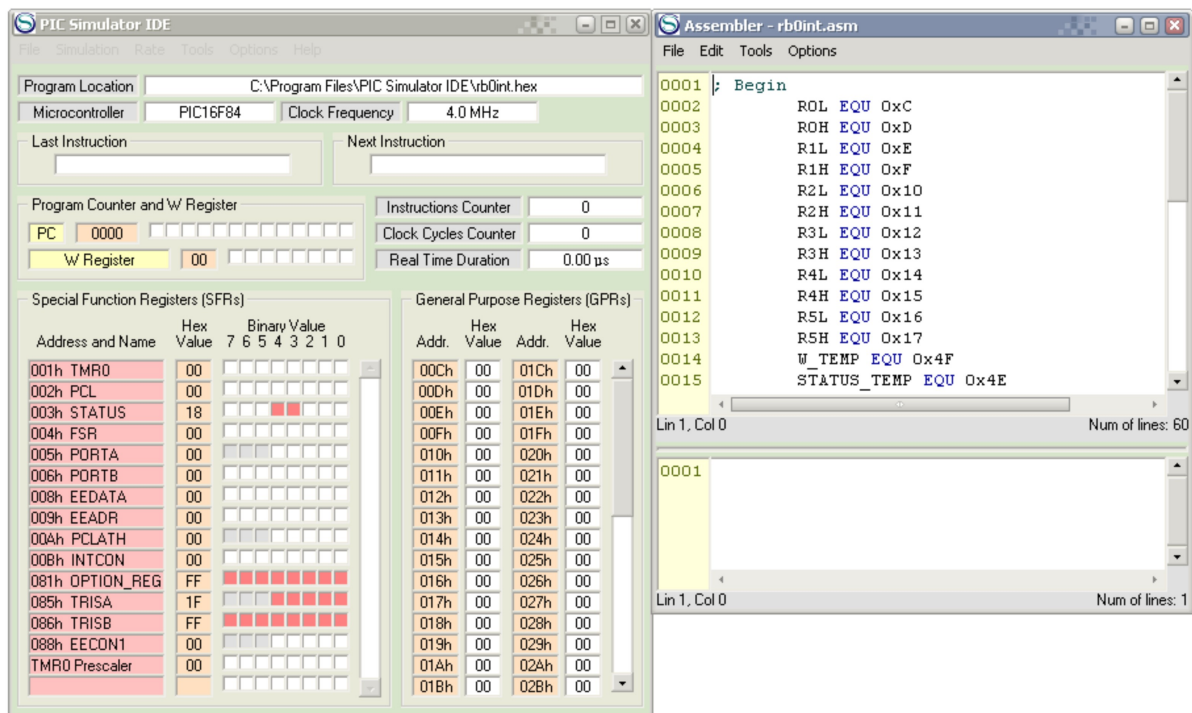


Рис. 4 Завантаження файлу rb0int.asm

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

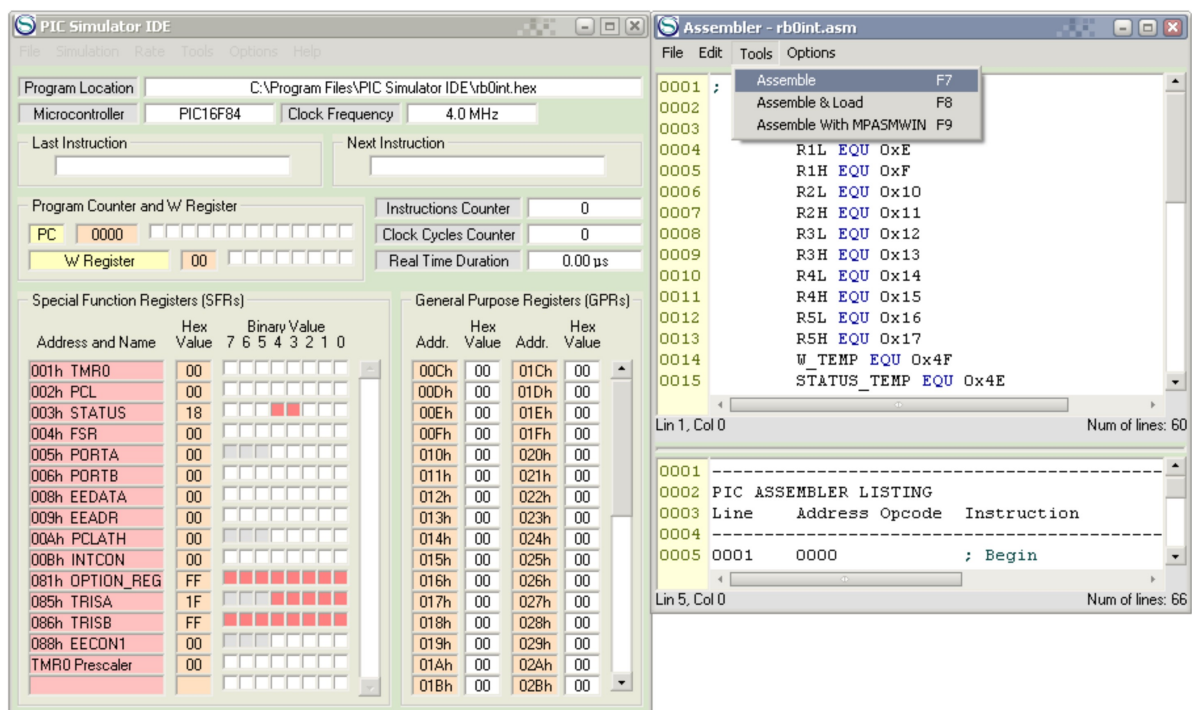


Рис. 5

2. Завдання на лабораторну роботу: прийом і передача даних з «апаратного» UART і відображення даних у вікні симуляції апаратного інтерфейсу UART програмного симулятора PIC Simulator IDE.

1. Вивчити програмну модель PIC Simulator IDE.
2. Вивчити команди обміну даними через порти 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 “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

У цій програмі використовується кілька різних команд роботи з зовнішніми пристроями. На початку програма посилає 6 рядків даних з «апаратного» UART, далі відповідає на отримані байти, посылаючи один рядок тексту для кожного отриманого байту.

Текст програми з файлу uart.asm має наступний вигляд:

; Begin

```
R0L EQU 0x20
R0H EQU 0x21
R1L EQU 0x22
R1H EQU 0x23
R2L EQU 0x24
R2H EQU 0x25
R3L EQU 0x26
R3H EQU 0x27
R4L EQU 0x28
R4H EQU 0x29
R5L EQU 0x2A
R5H EQU 0x2B
ORG 0x0000
BCF PCLATH,3
BCF PCLATH,4
GOTO L0002
ORG 0x0004
RETFIE
```

L0002:

; 1: Dim i As Byte 'declare a variable

; The address of 'i' is 0x2C

```
i EQU 0x2C
```

; 2: Hseropen 9600 'open hardware uart port for baud rate 9600

; Exact baud rate: 9615

; Baud rate error: 15 (.156%)

```
BSF STATUS,RP0
MOVLW 0x19
MOVWF SPBRG
BSF TRISC,6
BSF TRISC,7
MOVLW 0x24
MOVWF TXSTA
BCF STATUS,RP0
MOVLW 0x90
MOVWF RCSTA
```

```

; 3: 'WaitMs 1000 'this delay should be used in a real device
; 4:
; 5: For i = 10 To 5 Step -1 'for-next loop
    MOVLW 0x0A
    MOVWF 0x2C
L0003:
    MOVLW 0x05
    SUBWF 0x2C,W
    BTFSS STATUS,C
    GOTO L0004
; 6: Hserout "Number: ", #i, CrLf 'send formatted output to serial port
    MOVLW 0x4E
    CALL HS01
    MOVLW 0x75
    CALL HS01
    MOVLW 0x6D
    CALL HS01
    MOVLW 0x62
    CALL HS01
    MOVLW 0x65
    CALL HS01
    MOVLW 0x72
    CALL HS01
    MOVLW 0x3A
    CALL HS01
    MOVLW 0x20
    CALL HS01
    MOVF 0x2C,W
    MOVWF R2L
    CLRF R2H
    CALL HS21
    MOVLW 0x0D
    CALL HS01
    MOVLW 0x0A
    CALL HS01
; 7: 'WaitMs 500 'this delay should be used in a real device
; 8: Next i
    MOVLW 0x01
    SUBWF 0x2C,F
    BTFSC STATUS,C
    GOTO L0003
L0004: MOVLW 0x1F
    ANDWF STATUS,F
; 9:
; 10: loop:
L0001:
; 11: Hserin i 'wait to receive a byte on serial port
    CALL HS10
    MOVWF 0x2C
; 12: Hserout "Number: ", #i, CrLf 'send formatted output to serial port
    MOVLW 0x4E
    CALL HS01

```

```

    MOVLW 0x75
    CALL HS01
    MOVLW 0x6D
    CALL HS01
    MOVLW 0x62
    CALL HS01
    MOVLW 0x65
    CALL HS01
    MOVLW 0x72
    CALL HS01
    MOVLW 0x3A
    CALL HS01
    MOVLW 0x20
    CALL HS01
    MOVF 0x2C,W
    MOVWF R2L
    CLRF R2H
    CALL HS21
    MOVLW 0x0D
    CALL HS01
    MOVLW 0x0A
    CALL HS01
; 13: Goto loop 'loop forever
    GOTO L0001
; End of program
L0005:GOTO L0005
; Division Routine
D001: MOVLW 0x10
    MOVWF R3L
    CLRF R2H
    CLRF R2L
D002: RLF R0H,W
    RLF R2L,F
    RLF R2H,F
    MOVF R1L,W
    SUBWF R2L,F
    MOVF R1H,W
    BTFSS STATUS,C
    INCFSZ R1H,W
    SUBWF R2H,F
    BTFSC STATUS,C
    GOTO D003
    MOVF R1L,W
    ADDWF R2L,F
    MOVF R1H,W
    BTFSC STATUS,C
    INCFSZ R1H,W
    ADDWF R2H,F
    BCF STATUS,C
D003: RLF R0L,F
    RLF R0H,F
    DECFSZ R3L,F

```



```

    GOTO D002
    MOVF R0L,W
    RETURN
; Hardware Serial Communication Routines
HS01: BTFSC PIR1,TXIF
    GOTO HS02
    GOTO HS01
HS02: MOVWF TXREG
    RETURN
HS10: BTFSC PIR1,RCIF
    GOTO HS11
    GOTO HS10
HS11: MOVF RCREG,W
    RETURN
; Hserout Decimal Conversion Routine
HS21: BSF R3H,7
    MOVLW 0x27
    MOVWF R1H
    MOVLW 0x10
    CALL HS22
    MOVLW 0x03
    MOVWF R1H
    MOVLW 0xE8
    CALL HS22
    CLRF R1H
    MOVLW 0x64
    CALL HS22
    CLRF R1H
    MOVLW 0x0A
    CALL HS22
    MOVF R2L,W
    GOTO HS23
HS22: MOVWF R1L
    MOVF R2H,W
    MOVWF R0H
    MOVF R2L,W
    MOVWF R0L
    CALL D001
    MOVF R0L,W
    BTFSS STATUS,Z
    BCF R3H,7
    BTFSC R3H,7
    RETURN
HS23: ADDLW 0x30
    CALL HS01
    RETURN
; End of listing
END

```

3. Послідовність роботи з симулятором при виконанні програми

Переглянемо результати роботи цієї програми в PIC Simulator IDE.
Для цього виконаємо наступне, вибравши модель МК PIC16F877:

1. Запустити PIC Simulator IDE;
 2. Натиснути Options | Select Microcontroller;
 3. Вибрати **PIC16F877** і натиснути кнопку Select;
 4. Натиснути File | Load Program;
 5. Натиснути Options | Change Clock Frequency;
 6. Ввести «4» і натиснути кнопку ОК;
 7. Натиснути Options | Change UART Trans? Mit / Receive Time;
 8. Ввести «100» і натиснути кнопку ОК;
 9. Натиснути File | Load Program;
 10. Вибрати файл uart.hex і натиснути кнопку Open;
 11. Натиснути Tools | Hardware UART Simulation Interface (відкриється вікно симуляції апаратного інтерфейсу UART);
 12. Вибрати Rate | Extremely Fast simulation rate;
 13. Натиснути Simulation | Start (почнеться виконання програми).
- Після цього необхідно почекати, поки програма видасть 6 рядків тексту в послідовний порт. Для пересилання даних в порт використовується одна з трьох доступних кнопок інтерфейсу UART. Програма відповідає, посылаючи рядок даних.
- Останні два кроки можна кілька разів повторити і подивитися на результати.
- Вигляд екрану з виконуваною програмою показаний на рис. 6.

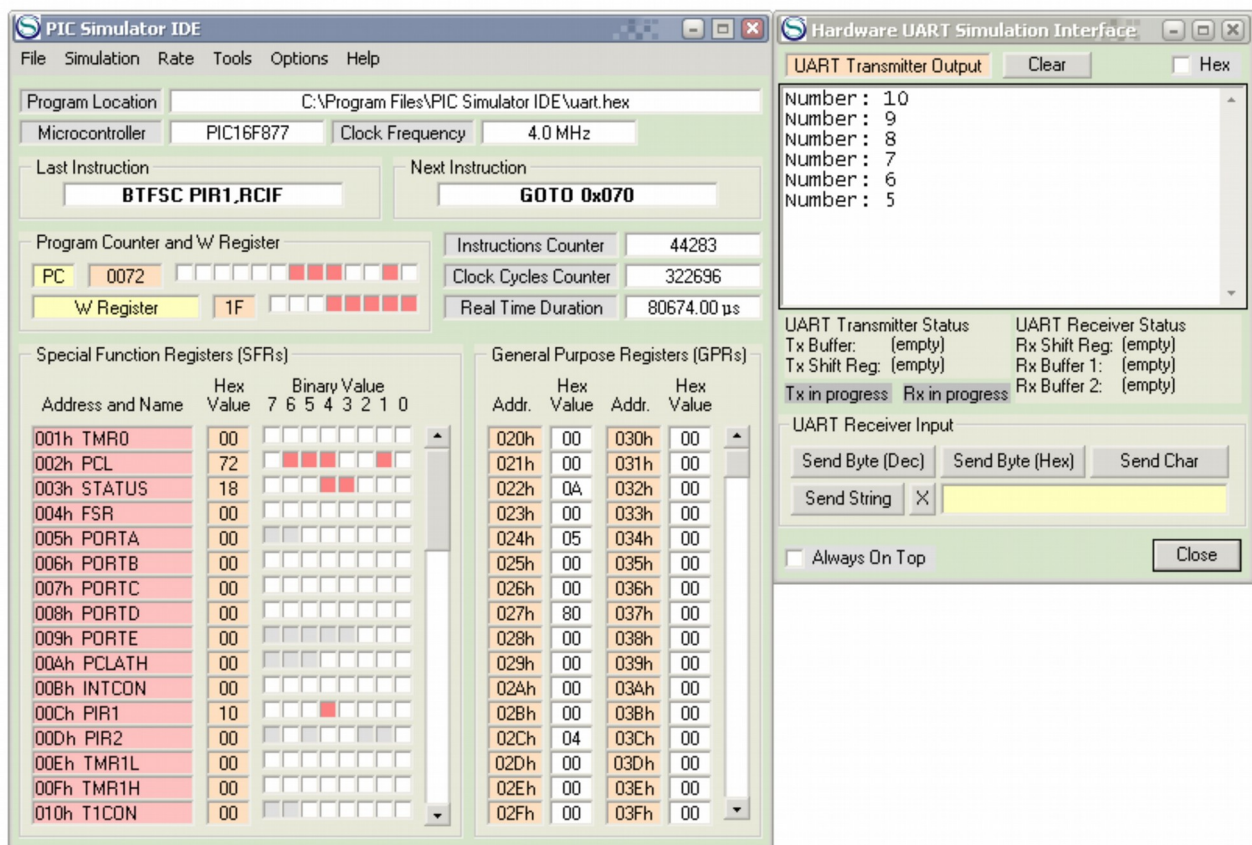


Рис. 6 Вигляд екрану з виконуваною програмою «Прийм і передача даних з "апаратного" UART і відображення даних на LCD екрані»

```

PIC DISASSEMBLER LISTING
Address Opcode Instruction
-----
0000 118A BCF PCLATH,3
0001 120A BCF PCLATH,4
0002 2805 GOTO L1
0003 0000 NOP
0004 0009 RETFIE
0005 1683 L1: BSF STATUS,RPO
0006 3019 MOVLW 0x19
0007 0099 MOVWF 0x19
0008 1707 BSF 0x07,6
0009 1787 BSF 0x07,7
000A 3024 MOVLW 0x24
000B 0098 MOVWF 0x18
000C 1283 BCF STATUS,RPO
000D 3090 MOVLW 0x90
000E 0098 MOVWF 0x18
000F 300A MOVLW 0x0A
0010 00AC MOVWF 0x2C
0011 3005 L5: MOVLW 0x05
0012 022C SUBWF 0x2C,W
0013 1C03 BTFSS STATUS,C
0014 2831 GOTO L2
0015 304E MOVLW 0x4E
0016 206B CALL L3
0017 3075 MOVLW 0x75
0018 206B CALL L3
0019 306D MOVLW 0x6D

```

Рис. 7 Фрагмент виконаної програми у вікні PIC Disassembler

З вікна “PIC Disassembler” (рис. 7) 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 PCLATH

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 регістра помаранчевим кольором означає наявність “1”, білим - “0”.

Вміст регістрів записуємо в шістнадцятковому коді за Прикладом 3.

Приклад 3.

Регістр	Вміст регістра
W register	1F

і т. д.

4. Контрольні запитання

1. Послідовність роботи послідовного порту UART мікроконтролера.
2. Як здійснюється програмування порту UART мікроконтролера.
3. Формат та призначення регістрів PC, W, PORTA...PORTE.
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. PIC-микроконтроллеры. Практика применения. М.: ДМК, 2002.
3. Предко М. Создайте робота своими руками на PIC- 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 PIC-микроконтроллерам.: Пер. с нем.-К.: “МК-Пресс”, 2007.-256 с., ил.