

Тернопільський національний технічний
університет імені Івана Пулюя

Кафедра автоматизації
технологічних процесів
і виробництв



Лабораторна робота № 6
з курсу
”Проектування
мікропроцесорних систем
керування технологічними
процесами”

Програмування мікроконтролера i8051 з
використанням програмної моделі
EdSim51. Сканування клавіатури

Тернопіль 2019

АВТОМАТИЗАЦІЯ

ІНФОРМАЦІЙНО-ТЕХНОЛОГІЧНА

Методичні вказівки до лабораторної роботи № 6 "Програмування мікроконтролера i8051 з використанням програмної моделі EdSim51. Сканування клавіатури" з курсу "Проектування мікропроцесорних систем керування технологічними процесами". Автор: Медвідь В.Р., Тернопіль: ТНТУ, 2019- 7 с.

Для студентів напряму підготовки: 151 "Автоматизація та комп'ютерно-інтегровані технології"

Автор: Медвідь В.Р.

Лабораторна робота присвячена розгляду методики сканування клавіатури на прикладі підпрограм для широкорозповсюджених однокристальних мікро-ЕОМ сімейства MCS-51. Методичні вказівки орієнтовані на студентів, котрі навчаються за напрямом підготовки "Автоматизація та комп'ютерно-інтегровані технології" і можуть бути корисні особам, що вивчають споріднені дисципліни.

Лабораторна робота №6
Програмування мікроконтролера i8051 з використанням програмної моделі EdSim51. Сканування клавіатури.

1. Теоретичні відомості

Сканування клавіатури

У програмі, що подана нижче, показано, як здійснюється сканування клавіатури. Виконання програми зупиняється, як тільки натиснута одна з кнопок.

У той час як жодна клавіша не натиснута, програма сканує лінії row0, row1, row2, row3 в прямому і зворотному напрямку до row0 безперервно.

При натисканні клавіші номер клавіші розташовується в R0.

В програмі використовується наступна нумерація клавіш та виводів клавіатури (рис.1).

11	10	9	row 3
8	7	6	row 2
5	4	3	row 1
2	1	0	row 0

col2 col1 col0

Рис. 1 Нумерація клавіш та виводів клавіатури

Номер натисутої клавіші буде зберігатися в R0. Таким чином, R0 спочатку скидається в ноль.

Слід звернути увагу на те, що при натисканні клавіші на зображенні клавіатури в полі інтерфейсу симулятора курсором миші (рис.3, поле “Keypad”), в регістр R0 буде записаний номер клавіші, що відповідає її нумерації на рис.1. Тобто, при натисканні клавіші “1” на полі “Keypad” симулятора, в регістр R0 запишеться число “11”, що є номером цієї клавіші відповідно до рис.1. При натисканні клавіші “2” – число “10” і т.д.

Кожен ключ сканується, і якщо клавіша не натиснена, вміст R0 збільшується на 1, тобто, при натисканні ключа (клавіші), R0 буде містити номер цього ключа.

Флаг загального призначення F0 в регістрі слова стану програми (PSW) використовується підпрограмою для сканування стовбців, щоб вказати, чи не знайшлася натиснутою клавіша в цьому стовбці.

Якщо після повернення з підпрограми colScan F0 рівний “1”, то це означає, що ключ був знайдений як замкнений (натиснутий).

2. Завдання

1. Дослідити програму сканування клавіатури.
2. Виконати програму сканування клавіатури на програмному симуляторі:

```
start:  
MOV R0, #0      ; очистка R0 — перший ключ є ключ 0  
; scan row0  
SETB P0.3      ; встановлення рядка row3  
CLR P0.0       ; очистка row0  
CALL colScan   ; виклик підпрограми сканування стовбця
```

```

JB F0, finish      ; якщо F0 дорівнює «1», перехід до кінця програми
                   ; (тому що натиснута клавіша була знайдена і її номер в R0)
; scan row1
SETB P0.0          ; встановити “1” на лінії row0
CLR P0.1           ; очистити лінію row1
CALL colScan       ; виклик підпрограми сканування стовбців
JB F0, finish      ; якщо F0=”1”, перейти на кінець програми
                   ; (тому що натиснута клавіша була знайдена і її номер в R0)
; scan row2
SETB P0.1          ; встановити “1” на лінії row1
CLR P0.2           ; очистити лінію row2
CALL colScan       ; виклик підпрограми сканування стовбців
JB F0, finish      ; якщо F0=”1”, перейти на кінець програми
                   ; (тому що натиснута клавіша була знайдена і її номер в R0)
; scan row3
SETB P0.2          ; встановити “1” на лінії row2
CLR P0.3           ; очистити лінію row3
CALL colScan       ; виклик підпрограми сканування стовбців
JB F0, finish      ; якщо F0=”1”, перейти на кінець програми
                   ; (тому що натиснута клавіша була знайдена і її номер в R0)
JMP start          ; повернутися до сканування лінії row 0

```

finish:

```

JMP $              ; виконання програми переходить сюди, коли натиснутий ключ
                   ; знайдено

```

colScan:

```

; підпрограма сканування стовбця
JNB P0.4, gotKey  ; якщо col0 рівне “0” - ключ знайдений
INC R0            ; в іншому випадку перейти до наступного ключа
JNB P0.5, gotKey  ; якщо col1 рівне “0” - ключ знайдений
INC R0            ; в іншому випадку перейти до наступного ключа
JNB P0.6, gotKey  ; якщо col2 рівне “0” - ключ знайдений
INC R0            ; в іншому випадку перейти до наступного ключа
RET              ; повернення з підпрограми — ключ не знайдено

```

gotKey:

```

SETB F0           ; ключ знайдено- записати “1” в F0
RET              ; і повернутися з підпрограми

```

Схема підключення клавіатури до МК51 показана на рис. 2.

3. Послідовність виконання роботи

3.1. Вивчити команди відповідно до завдання. Вивчення кожної команди проводити наступним чином:

3.1.1. Відкрити інтерфейс емулятора, двічі клацнувши клавішею миші на архівованому файлі «EdSim51.jar». Відкриється інтерфейс програмного емулятора, зображений на рис. 3.

Середнє поле емулятора, що називається “**Панель коду Асемблера**”, в верхній частині містить кнопки “**Reset**”, “**Assm**”, “**Run**”, “**Load**”, “**Save**”, “**Copy**”, “**Paste**”.

Панель коду використовується для:

- **набору команд** програми з клавіатури. Для цього курсор встановлюється в верхній частині панелі і вводиться програма по одній команді в рядку (при потребі, з міткою та коментарем)(див. рис. 3);

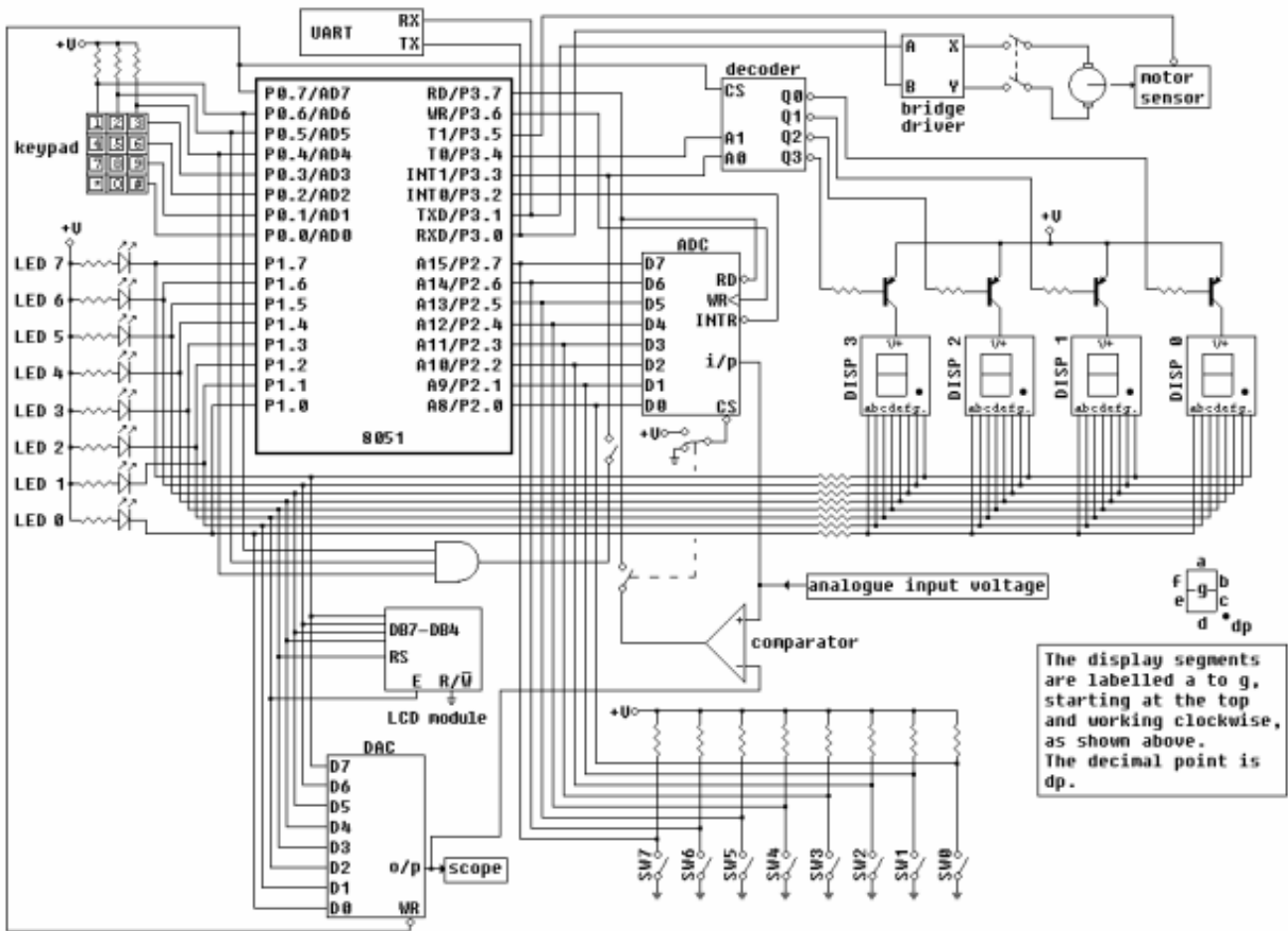


Рис. 2 Схема підключення клавіатури до МК51

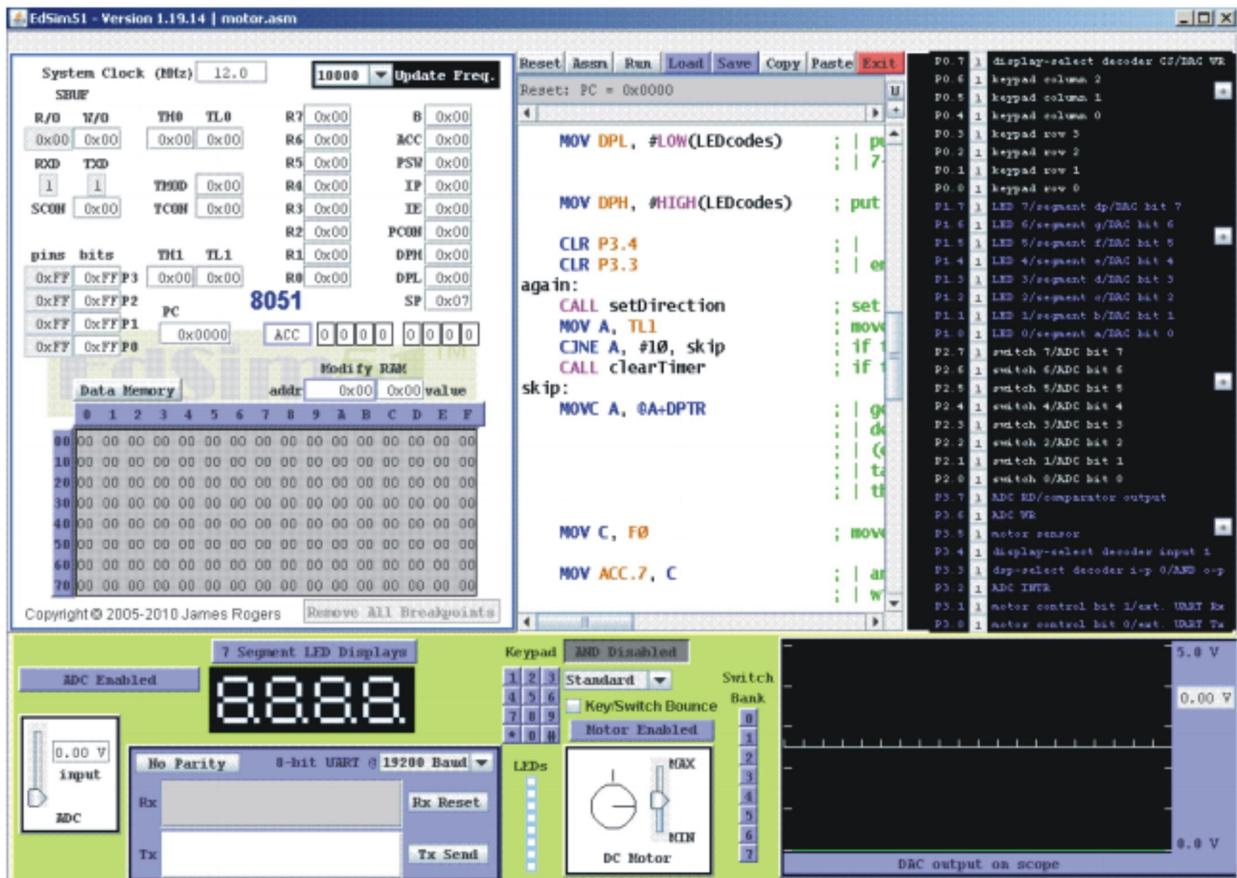


Рис. 3 Інтерфейс програмного емулятора

- **завантаження** вже існуючої програми. Для цього необхідно на панелі вгорі натиснути кнопку **“Load”** і вказати шлях до потрібного файлу;
- **запису** набраного файлу. Для цього потрібно натиснути кнопку **“Save”** і вказати шлях для збереження файлу.

3.1.2. Перед виконанням програми необхідно натиснути кнопку **“Assm”** панелі для асемблювання програми. Після цього, якщо команда записана невірно, в рядку під верхнім рядом кнопок панелі (на рис.1 виділений сірим кольором) з'явиться повідомлення про помилку, а **колір рядка зміниться на червоний**. Червоним кольором буде виділена також невірно написана команда.

Якщо помилки відсутні, зліва від команд набраної програми з'являться адреси, і сама програма буде готова до виконання. Після асемблювання кнопка **“Assm”** зміниться на кнопку **“Step”**. Таким чином, є можливим виконувати програму покомандно **в кроковому режимі**, натискаючи кнопку **“Step”** після виконання кожної команди, або **в автоматичному режимі**, коли виконується вся програма, натиснувши один раз кнопку **“Run”**. В останньому випадку програму слід закінчувати командою **“Stop”**.

При написанні програми можна користуватися для копіювання її фрагментів та вставки в будь-якому місці **“Панелі коду Асемблера”** кнопками **“Copy”** та **“Paste”**.

Щоб зупинити виконання програми і скинути в початковий стан реєстри мікроконтролера емулятора необхідно натиснути кнопку **“Reset”**.

3.1.3. Записати в звіт зміни в вікнах реєстрів мікроконтролера, які використовуються при виконанні програм, за прикладом, наведеним в табл. 1 (для 5..6 команд з програми).

Таблиця 1. Результати виконання команд

№	Команда	Код	Виконувана операція	Вміст використовуваних реєстрів і комірок пам'яті до і після виконання		Пояснення
				До	Після	
1	MOV A,R0	E8	Пересилання байту даних з реєстру R0 в акумулятор A	A/00 PC/00 PSW/00	A/F2 PC/01 PSW/01	
2
...
...

***Примітка**

1. Якщо Ви хочете виконати якусь з команд над вмістом реєстру чи комірки пам'яті, наприклад, команду пересилання з реєстру в реєстр, необхідно в реєстр, з якого буде здійснене пересилання, командою MOV попередньо записати якесь значення операнду (адресу чи константу).

3. Програма, що виконується, буде записана в пам'ять програм, вміст якої можна побачити, натиснувши на кнопку **“Data memory”** в нижній частині **“Панелі пам'яті даних та програмної пам'яті”**, що знаходиться зліва від **“Панелі коду Асемблера”**. Після натискання кнопка **“Data memory”** зміниться на кнопку **“Code memory”**, тобто буде висвічуватися в полі пам'яті вміст пам'яті програм.

4. Контрольні запитання

1. Використовуючи електричну принципову схему, пояснити, як під'єднана клавіатура до мікроконтролера?
2. Яке призначення лінії P0.4...P0.6 мікроконтролера в схемі на рис.1?
3. Яке призначення лінії P0.0...P0.3 мікроконтролера в схемі на рис.1?
4. Пояснити алгоритм роботи програми.

5. Література.

1. Проектирование цифровых устройств на однокристальных микроконтроллерах / В. В. Сташин и др. - М.: Энергоатомиздат, 1990. – 224 с.
2. Микропроцессоры / Под ред. Преснухина Л.Н., т. 1, 2, 3. - М.: Высшая школа, 1986.
3. Бойко Н.П., Стеклов В.К. Системы автоматического управления на базе микро-ЭВМ. - К.: Техника, 1989. - 182 с.