

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

**Кафедра автоматизації**  
**технологічних**  
**процесів та виробництв**

**Методичні вказівки**  
**для виконання лабораторної роботи №7 “Ввід**  
**інформації через паралельний порт**  
**мікроконвертера ADuC842”**  
**з курсу “Розробка систем керування на основі**  
**OMEOM”**

**Тернопіль 2016**

Методичні вказівки для виконання лабораторної роботи № 7 «Від інформації через паралельний порт мікроконвертера ADuC842» з курсу «Розробка систем керування на основі OMEOM».

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

Відповідальні за випуск

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

## Лабораторна робота № 7

### Ввід інформації через паралельний порт мікроконвертера ADuC842

#### 1. Мета роботи

- 1.1. Вивчити особливості роботи паралельних портів мікроконтролера.
- 1.2. Вивчити схеми підключення кнопок і датчиків до цифрових мікросхем.
- 1.3. Навчитися визначати стан кнопок за допомогою програми.

#### 2. Попередня підготовка до роботи

- 2.1. Вивчити схеми паралельних портів мікроконтролерів.
- 2.2. Вивчити схеми підключення кнопок і клавіатури до паралельних портів мікроконтролерів.
- 2.3. Скласти програму, що засвічує світлодіоди, які відповідають номерам бітів порту вводу P0, встановлених в одиничний стан.

#### 3. Підключення контактів до паралельного порту

Джерела дискретної інформації можуть мати різну фізичну природу. Вони можуть перебувати на значній відстані від контролера, мати різну напругу живлення, але їх дані повинні бути безпомилково зчитані керуючою програмою мікропроцесорної системи.

Практично завжди при роботі з зовнішніми датчиками потрібна гальванічна розв'язка датчиків і керуючої мікропроцесорної системи.

Для вирішення зазначених проблем та реалізації гальванічної розв'язки датчиків і мікропроцесорного пристрою всі датчики з точки зору схеми являють собою контакти, які працюють на замикання. Тому, схема підключення датчика і кнопки не розрізняються.

З боку мікропроцесорного пристрою необхідно перетворити замикання / розмикання контактів в логічні рівні, необхідні для належного функціонування мікропроцесорного пристрою.

Така схема приведена на рисунку 1.

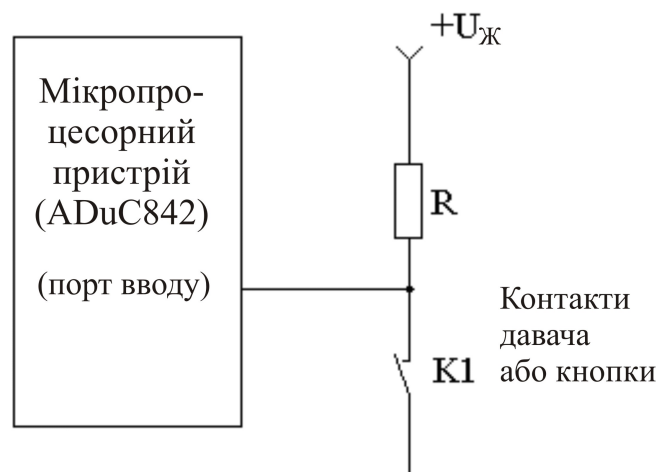


Рисунок 1. Підключення кнопки до паралельного порту вводу

У наведеній схемі резистор R подає одиничний потенціал на вхід паралельного порту. Застосування резистора дозволяє закоротити вхід паралельного порту на корпус пристрою, не викликаючи надмірного струму через контакт. При замиканні контактів кнопки на вхід паралельного порту подається потенціал логічного нуля. Ці логічні рівні можуть бути виявлені за допомогою програми, завантаженої в мікроконтролер.

У мікроконтролері порт P1 може бути використаний тільки для введення інформації. На рисунку 2 наведена схема одного біту паралельного порту P1. Зовнішній вивід порту може бути підключений або до внутрішньої шини, або до входу АЦП.

Для того, щоб порт працював на введення цифрової інформації, необхідно в його тригер записати логічний 0 (в цьому випадку ключ буде в верхньому положенні).

Таким чином, перед введенням інформації в порт P1 необхідно записати 0. Шістнадцяткова адреса порту P1 дорівнює 90.

Напруги подаються на порт P1 в схемі стенду за допомогою ключів S0, S1, S2, S3.

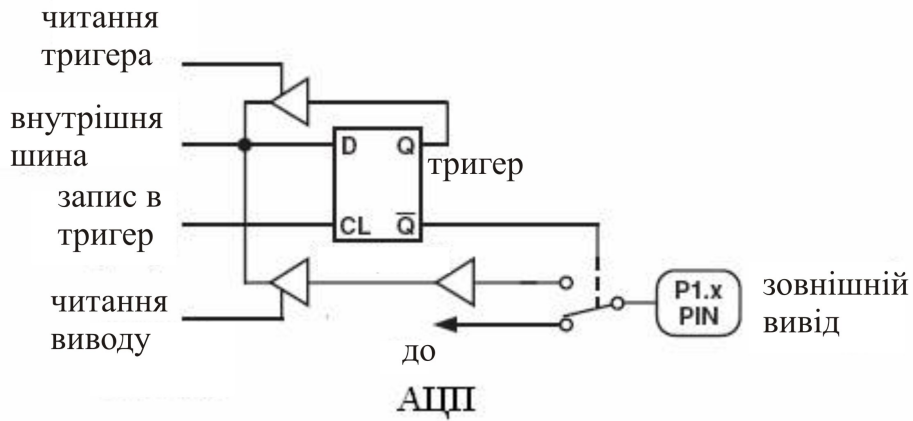


Рисунок 2. Схема одного біта порту P1 AduC842

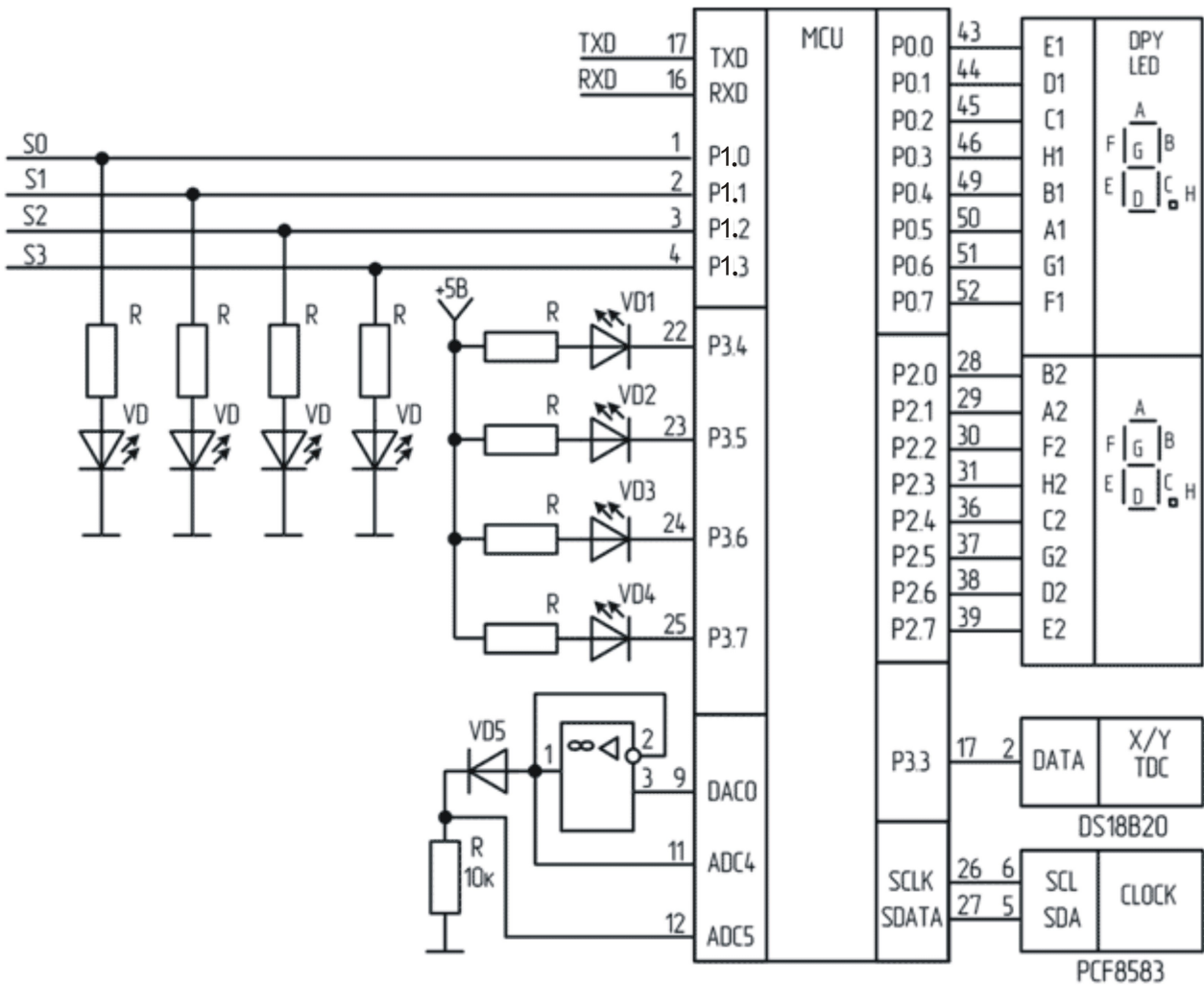


Рисунок 3. Схема виводу даних на семисегментний дисплей

#### 4. Завдання до роботи в лабораторії

4.1. Розробити програму, що аналізує стан бітів порту вводу P1.0, P1.1, P1.2, P1.3 і засвічує відповідні світлодіоди VD1, VD2, VD3, VD4.

Нижче наводиться таблиця, в якій показано, які світлодіоди повинні запалюватися при установці відповідних значень бітів порту P1 (світлодіод засвічується рівнем логічного "0" на виході відповідної лінії порту P3).

| Номер варіанту | P1.3 | P1.2 | P1.1 | P1.0 | Повинні запалитися світлодіоди |
|----------------|------|------|------|------|--------------------------------|
| 1              | 0    | 0    | 0    | 1    | VD1                            |
| 2              | 0    | 0    | 1    | 0    | VD2                            |
| 3              | 0    | 0    | 1    | 1    | VD1, VD2                       |
| 4              | 0    | 1    | 0    | 0    | VD3                            |
| 5              | 0    | 1    | 0    | 1    | VD1, VD3                       |
| 6              | 0    | 1    | 1    | 0    | VD2, VD3                       |
| 7              | 1    | 0    | 0    | 0    | VD4                            |
| 8              | 1    | 0    | 0    | 1    | VD1, VD4                       |
| 9              | 1    | 0    | 1    | 0    | VD2, VD4                       |
| 0              | 1    | 0    | 1    | 1    | VD1, VD2, VD4                  |

4.2. Створити новий проект в Keil.

4.3. Налаштувати проект наступним чином: вибрати мікроконтролер AduC842, не забудьте встановити галочку навпроти Create HEX file (інакше не буде створено hex-файл, навіть якщо немає помилок).

4.4. Записати відповідний HEX-файл в пам'ять мікроконтролера.

4.5. Переконайтеся за значеннями бітів у відповідних лініях порту, що горять відповідні світлодіоди.

4.6. Виконайте покрокове налагодження програми з використанням кнопки F11. На кожному кроці виконання програми запишіть значення використовуваних регістрів відповідно до таблиці 1.

Таблиця 1. Результати виконання команд

| №   | Команда | Код | Виконувана операція | Вміст використовуваних регістрів і комірок пам'яті до і після виконання |                         | Пояснення |
|-----|---------|-----|---------------------|---|-------------------------|-----------|
|     |         |     |                     | До  | Після                   |           |
| 1   |         |     |                     | A/00<br>PC/00<br>PSW/00   | A/F2<br>PC/01<br>PSW/01 |           |
| 2   | ...     | ... | ...                 | ...   | ...                     | ...       |
| ... | ...     | ... | ...                 | ...   | ...                     | ...       |
| ... | ...     | ... | ...                 | ...   | ...                     | ...       |

#### 5. Методичні вказівки до виконання роботи

При розробці програми необхідно перед собою мати електричну схему.

Приклад програми, що запалює світлодіоди VD1, VD2, VD3 та VD4 за умови, що P1.0, P1.1, P1.2, P1.3 приймають відповідні значення ("0" чи "1").

Наприклад, світлодіоди VD1, VD2 запаляться в тому випадку, якщо на вході порту P1 буде присутнє число 3 (код на вході порту P1 буде дорівнювати 00000011).

Перш ніж відлагоджувати програму, слід попередньо встановити програмно відповідні рівні на входах S0, S1, S2, S3, тобто замість першої команди MOV P1, 0 потрібно записати значення її другого байту відповідно до Вашого варіанту. Наприклад, для варіанту 3 це буде число 3 (або двійкова комбінація 00000011) і, відповідно, першою командою буде команда MOV P1, 3.

```
MOV P1, 0 ; в усі розряди порту P0 записуємо нулі (переводимо
           ; порт в режим прийому цифрової інформації)
MOV P3, 1 ; в усі розряди порту P3 записуємо одиниці
           ; (переводимо порт в початковий режим – всі діоди
           ; пне світяться)

start:
MOV A, P0 ; ввести дані в A з порту P0
ANL A, #00001111b ; виділити чотири молодші розряди порту P0
SUBB A, 1 ; відняти від A логічну «1»
JZ adr1 ; якщо A=0, перейти на мітку adr1
SUBB A, 2 ; відняти від A логічну «2»
JZ adr2 ; якщо A=0, перейти на мітку adr2
SUBB A, 3
JZ adr3
SUBB A, 4
JZ adr4
SUBB A, 5
JZ adr5
SUBB A, 6
JZ adr6
SUBB A, 8
JZ adr7
SUBB A, 9
JZ adr8
SUBB A, A
JZ adr9
SUBB A, B
JZ adr0

adr1:
CLR P3.4 ; засвітити діод VD1
CALL delay ; перейти на підпрограму часової затримки
JMP start ; перейти на початок програми
adr2:
CLR P3.5 ; засвітити діод VD2
CALL delay ; перейти на підпрограму часової затримки
JMP start ; перейти на початок програми
adr3:
CLR P3.4 ; засвітити діод VD1
CLR P3.5 ; засвітити діод VD2
CALL delay ; перейти на підпрограму часової затримки
JMP start ; перейти на початок програми
adr4:
CLR P3.6 ; засвітити діод VD3
```

|              |   |
|--------------|---|
| CALL delay   | ; перейти на підпрограму часової затримки |
| JMP start    | ; перейти на початок програми             |
| adr5:        |   |
| CLR P3.4     | ; засвітити діод VD1                      |
| CLR P3.6     | ; засвітити діод VD3                      |
| CALL delay   | ; перейти на підпрограму часової затримки |
| JMP start    | ; перейти на початок програми             |
| adr6:        |   |
| CLR P3.5     | ; засвітити діод VD2                      |
| CLR P3.6     | ; засвітити діод VD3                      |
| CALL delay   | ; перейти на підпрограму часової затримки |
| JMP start    | ; перейти на початок програми             |
| adr7:        |   |
| CLR P3.7     | ; засвітити діод VD4                      |
| CALL delay   | ; перейти на підпрограму часової затримки |
| JMP start    | ; перейти на початок програми             |
| adr8:        |   |
| CLR P3.4     | ; засвітити діод VD1                      |
| CLR P3.7     | ; засвітити діод VD4                      |
| CALL delay   | ; перейти на підпрограму часової затримки |
| adr9:        |   |
| CLR P3.5     | ; засвітити діод VD2                      |
| CLR P3.7     | ; засвітити діод VD4                      |
| CALL delay   | ; перейти на підпрограму часової затримки |
| adr0:        |   |
| CLR P3.4     | ; засвітити діод VD1                      |
| CLR P3.5     | ; засвітити діод VD2                      |
| CLR P3.7     | ; засвітити діод VD4                      |
| CALL delay   | ; перейти на підпрограму часової затримки |
| END          | ; закінчити виконання програми            |
|              |   |
| delay:       |   |
| MOV R0, #200 | ; підпрограма часової затримки            |
| DJNZ R0, \$  |   |

## 6. Зміст звіту

6.1. Мета роботи.

6.2. Еквівалентна схема підключення кнопки до паралельного порту.

6.3. Графічна схема алгоритму програми.

6.4. Оригінальний текст програми (без урахування виправлень, зроблених в ході трансляції та налагодження програми).

6.5. Вміст файлу лістингу вихідного тексту програмного модуля (налагоджений варіант).

6.6. Вміст hex - файлів.

6.7. Висновки про виконану лабораторну роботу.