

Міністерство освіти та науки України  
Національний університет водного господарства та  
природокористування  
Кафедра автоматизації, електротехнічних та комп'ютерно-  
інтегрованих технологій

**04-03-267**

**МЕТОДИЧНІ ВКАЗІВКИ**

до виконання практичних робіт  
з навчальної дисципліни

**«Електроніка та мікропроцесорна техніка».**

**Частина 2. «Мікропроцесорна техніка»**

для здобувачів вищої освіти першого (бакалаврського) рівня  
за освітньо-професійною програмою

«Автоматизація та комп'ютерно-інтегровані технології»  
спеціальності 151 «Автоматизація та комп'ютерно-інтегровані  
технології» денної та заочної форм навчання

Рекомендовано науково-методичною  
радою з якості ННІ АКОТ

Протокол № 8 від 29. 04. 2020 р.

Методичні вказівки до виконання практичних робіт з навчальної дисципліни «Електроніка та мікропроцесорна техніка». Частина 2. «Мікропроцесорна техніка» (для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Автоматизація та комп'ютерно-інтегровані технології» спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» денної і заочної форм навчання [Електронне видання] / Христюк А. О., Маланчук Є. З. – Рівне : НУВГП, 2020. – 29 с.

Укладачі: Христюк А. О., к.т.н., доцент кафедри АЕКІТ;  
Маланчук Є. З., д.т.н., професор кафедри АЕКІТ.

Відповідальний за випуск: Древецький В. В., д.т.н., професор,  
завідувач кафедри АЕКІТ.

Керівник групи забезпечення спеціальності Древецький В. В.

© Маланчук Є. З.,  
Христюк А. О., 2020  
© НУВГП, 2020

## Зміст

	стор.
Вступ .....	4
1. Перетворення інформації в різних системах числення ....	5
2. Розробка структури мікропроцесорних систем .....	15
3. Розробка структурної схеми і алгоритму функціонування мікропроцесорної системи .....	21

## Вступ

Якісне та осмислене засвоєння теоретичної частини курсу «Електроніка та мікропроцесорна техніка». Частина 2. «Мікропроцесорна техніка» неможливе без надбання практичних навичок шляхом виконання практичних розрахунків та проектних завдань. Виконання запропонованих практичних робіт охоплює основні розділи курсу. Кожна з робіт включає в себе тему, мету, короткі теоретичні відомості, а також порядок виконання роботи. Виконавши розрахункову частину (або виконавши проектне рішення), студент заносить результати до звіту. Далі, самостійно, студент виконує необхідні операції, проводить розрахунки, виконує проектні роботи. Результатом цієї роботи є звіт з практичної роботи. Готовий звіт подається викладачу на папері або завантажується в електронний кабінет системи Moodle. Завантаження звіту з практичної роботи допускається лише з корпоративної пошти студента. При цьому звіт завантажується у відповідний розділ системи Moodle. У розділ відповідної практичної роботи завантажується файл зі звітом у форматі: \*.doc, \*.docx або \*.pdf. Успішне виконання запропонованих практичних робіт сформує у студента інженерні навички у подальшій професійній діяльності, а також закріпить теоретичні знання з курсу «Електроніка та мікропроцесорна техніка». Частина 2. «Мікропроцесорна техніка».

# **Робота 1. Перетворення інформації в різних системах числення**

## **1.1. Мета роботи**

Засвоїти правила перетворення чисел та отримати практичні навички при рішенні задач з використанням різних систем числення

## **1.2. Теоретичні відомості**

Для аналізу процесів, які відбуваються в пристроях мікропроцесорних систем, та їх синтезу необхідно знати правила роботи з числовою інформацією. Діяльність людини пов'язана з використанням десяткової системи числення. При роботі з мікропроцесорною технікою разом з десятковою використовують двійкову та шістнадцяткову системи числення. Будь-яка з них має певний набір цифр і правила їх записування. Загальна кількість усіх цифр системи називається її основою. Поряд написані цифри утворюють числа, причому кожна цифра займає певну позицію.

### **Десяткова система числення**

В десятковій системі числення числа записують десятима різними числами – від 0 до 9, тобто основа системи дорівнює 10. Кожна позиція оцінюється певним значенням: крайня справа нульова позиція (100) містить одиниці; наступна перша позиція (101) – десятки; друга позиція (102) містить сотні і так далі.

Будь-яке десяткове число можливо записати у розгорнутому вигляді. Для цього кожному цифру в числі треба помножити на значення позиції (вагу), яку вона займає, і всі добутки додати:

$$\begin{aligned} &10^3 \ 10^2 \ 10^1 \ 10^0 \\ &6 \ 3 \ 9 \ 4 = 6 \times 10^3 + 3 \times 10^2 + 9 \times 10^1 + 4 \times 10^0. \end{aligned}$$

Якщо десяткове число має дробову частину, то її відокремлюють від цілої комою (або крапкою). Значення першої зліва від коми позиції дорівнює  $10^{-1}$ , наступних позицій – другої і третьої відповідно дорівнює  $10^{-2}$  та  $10^{-3}$  і так далі. Розгорнута форма запису такого числа має вигляд:

$$10^{-1} \ 10^{-2} \ 10^{-3}$$

$$0,367 = 3 \cdot 10^{-1} + 6 \cdot 10^{-2} + 7 \cdot 10^{-3}.$$

Десяткове число може бути записане у формі з фіксованою комою, наприклад: 6,74; +8,9; -17,21. Другим способом його запису є форма зображення числа з плаваючою комою, при якому запис того самого числа у різних варіантах залежить від місця коми:  $258 \rightarrow 2,58 \times 10^2$ ;  $25,8 \times 10^1$ ;  $2580 \times 10^{-1}$  і тому подібне. Число, записане у формі з плаваючою комою у вигляді  $M \times 10^p$  є нормалізованим, причому  $0,1 \leq M < 1$  (наприклад,  $3675 \rightarrow 0,3675 \cdot 10^4$ ). В нормалізованому числі вираз перед знаком множення називають *мантисою*, а показник степеня основи системи числення – *порядком*.

Використання пакету підпрограм арифметики чисел з плаваючою комою підвищує точність обчислювань при роботі з різноманітними числовими даними.

### Двійкова система числення

Мікропроцесорні системи працюють з електричними сигналами, які кодуються двійковими числами і тому двійкова система числення лежить в основі їх роботи. Двійкова система числення, або система з основою “2” використовує тільки дві цифри “0” та “1”. Ці двійкові цифри мають назву бітів (від слів “binary digit” – двійкова одиниця). Фізично біт “0” представляють низькою напругою “L” (LOW), величина якої прямує до “0”, а біт “1” – напругою високого рівня певного значення “H” (HIGH), наприклад +5В.

Як і у десятковому числі, кожна цифра двійкового числа займає певну позицію. Крайня справа позиція у двійковому числі має значення, що дорівнює  $2^0$ , наступні відповідно  $2^1$ ,  $2^2$ ,  $2^3$  і так далі. Десяткове число з двійкового отримують за формулою:

$$N = a_1 \times 2^{n-1} + a_2 \times 2^{n-2} + \dots + a_{n-1} \times 2^1 + a_n \times 2^0,$$

де  $a$  – число, значення якого дорівнює “0” або “1”;  $n$  – кількість цифр у двійковому числі (довжина двійкового числа). Наприклад, розгорнуту форму двійкового числа 11101 можна записати:

$$11101 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0.$$

Щоб не плутати числа, складені з тих самих цифр, але таких, що належать до різних систем числення, після кожного числа в дужках зазначають систему числення після кожного числа в дужках зазначають систему числення, наприклад  $11101_{(2)} = 29_{(10)}$ .

Дробова частина двійкового числа відокремлюється від цілої за допомогою коми, причому кожна позиція справа від коми має таке значення:  $2^{-1}$ ,  $2^{-2}$ ,  $2^{-3}$  і так далі. Отже, наприклад, розгорнута форма запису двійкового числа:

$$0,011_{(2)} = 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = 0,375_{(10)}$$

Двійкові числа, як і десяткові записують у двох формах: з фіксованою і плаваючою комою.

Наприклад:

з фіксованою комою

$$1011_{(2)} = 27_{(10)};$$

$$0,111_{(2)} = 0,875_{(10)};$$

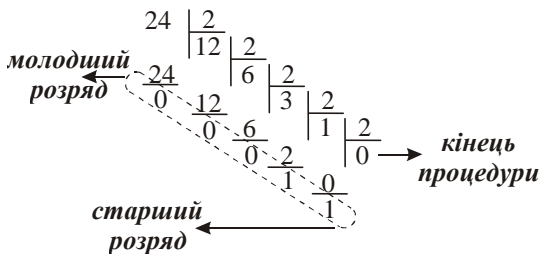
$$-11,01_{(2)} = -3,375_{(10)};$$

з плаваючою комою

$$1110 \times 10^{11}_{(2)} = 14 \times 2^3_{(10)};$$

$$0,11 \times 10^{110}_{(2)} = 0,875 \times 2^6_{(10)}.$$

У практичній роботі часто буває потрібно перетворювати числа десяткової системи числення в числа двійкової системи. Для цього ціле десяткове число ділять на основу двійкової системи числення, тобто на 2. Остача від ділення на 2 може дорівнювати 0 або 1. Значення остачі присвоюється молодшому розряду шуканого двійкового числа. Результат першого ділення знову ділиться на 2. Остачі ("0" або "1") присвоюється наступний розряд двійкового числа. Подібна процедура повторюється доти, поки результат ділення не буде дорівнювати "0". Остача від останньої дії ділення є значенням старшого розряду двійкового числа.



Отже,  $24_{(10)} = 11000_{(2)}$ .

При перетворенні десяткового дробу, дробову частину двійкового числа множать на 2. Якщо результат буде менший від 1, то старшому розряду шуканої дробової частини двійкового числа присвоюють значення “0”, а якщо більший ніж “1”, то “1”. Результат попередньої операції знову множать на 2, причому для множення беруть тільки його дробову частину. Аналогічну операцію проводять з новим результатом. Описану процедуру повторюють доти, поки результат наступного множення точно не дорівнюватиме “1”, або поки не буде досягнуто необхідної точності.

Приклад:

1.  $0,625 \times 2 = 1,25 \rightarrow 1$  старший розряд дробової частини двійкового числа
2.  $0,25 \times 2 = 0,5 \rightarrow 0$  наступний розряд дробової частини двійкового числа
3.  $0,5 \times 2 = 1 \rightarrow 1$  молодший розряд дробової частини двійкового числа

Таким чином,  $0,625_{(10)} = 0,101_{(2)}$ .

Якщо десяткове число має дробову частину, то в двійкову систему перетворюють цілу частину згідно першому правилу, а дробову частину – за другим і потім записують загальний результат.

### **Шістнадцяткова система числення**

Числа у двійковій системі числення займають багато позицій і, щоб спростити їх використання, застосовують шістнадцяткову систему числення (hexadecimal). Її основою є число 16 і вона використовується як засіб скорочення запису двійкових чисел при написанні програм. Шістнадцяткова система містить 16 символів: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E і F, тобто з 0 по 9 – у вигляді цифр, а з 10 по 15 – у вигляді латинських літер, де літера A визначає десять, B – одинадцять і так далі до літери F, яка відповідає п’ятнадцяти. Кожний шістнадцятковий символ може бути визначений, як сполучення 4-х біт. Для перетворення двійкового числа в шістнадцяткове треба почати з молодшого байта (8 біт) і поділити число на групи з 4-х біт еквівалентною шістнадцятковою цифрою. Для переводу двійкових чисел у шістнадцяткові та десяткові і навпаки



доцільно користуватись таблицею 1, яку з часом бажано запам'ятати, як таблицю множення.

Таблиця 1

Двійкові числа	Десяткові числа	Шістнадцяткові числа
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

Перетворимо двійкове число 1001 1110 в шістнадцяткове. Для цього ділимо двійкове число на групи 1110<sub>(2)</sub> і 1001<sub>(2)</sub>. Кожну з них замінюємо відповідним шістнадцятковим числом E<sub>(16)</sub> і 9<sub>(16)</sub>. Тоді шукане шістнадцяткове число дорівнює 1001 1110<sub>(2)</sub> = 9E<sub>(16)</sub>.

Навпаки, для перетворення шістнадцяткового числа 7F<sub>(16)</sub> у двійкове число кожен шістнадцяткову цифру замінюють відповідним двійковим числом з 4-х біт:

$$F_{(16)} = 1111_{(2)} \text{ і } 7_{(16)} = 0111_{(2)}.$$

Тоді еквівалентне двійкове число буде дорівнювати

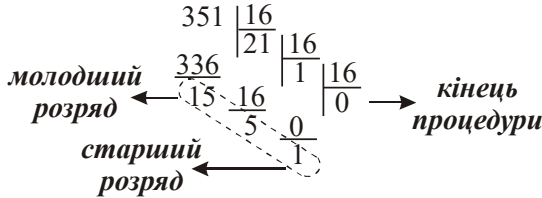
$$7F_{(16)} = 0111 1111_{(2)}.$$

Важливим є також перетворення шістнадцяткових чисел у десяткові і навпаки. Розглянемо процес перетворення шістнадцяткового числа 2C6E в десяткове:

$$\begin{aligned} 2C6E_{(16)} &= 2 \times 16^3 + C \times 16^2 + 6 \times 16^1 + E \times 16^0 = \\ &= 2 \times 4096 + 12 \times 256 + 6 \times 16 + 14 \times 1 = 11374_{(10)}. \end{aligned}$$

Значеннями перших чотирьох шістнадцяткових цифр є відповідно 4096, 256, 16 і 1. Десяткове число містить 14 ( $E_{16}$ ) одиниць, 6 чисел  $16^1$ , 12 ( $C_{16}$ ) чисел  $16^2$  і 2 числа  $16^3$ . Сума цих чисел є десятиковим числом  $11374_{(10)}$ .

Процедура перетворення десятикового числа  $351_{(10)}$  в шістнадцяткове має такий вигляд:



У результаті отримаємо:

$$351_{(10)} = 1515_{(16)} = 15F_{(16)}.$$

Більшість мікропроцесорів виконує арифметичні операції додавання і віднімання, а деякі – також операції множення і ділення двійкових чисел.

### Двійкове додавання

Дію двійкового додавання розпочинають з додавання молодших розрядів доданків. Якщо результат додавання більше 1, то до наступного розряду переносять одиницю, а в молодшому розряді пишуть нуль. Потім додають цифри наступних розрядів з урахуванням одиниць, перенесених з попереднього розряду до одержання суми, яку ми шукаємо.

Таблиця додавання двійкових чисел має такий вигляд:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

З таблиці слідує, що додавання двох одиниць дає нуль у наймолодшому розряді, а одиниця переноситься в наступний старший розряд.

Приклад:

1		1	1	1		перенесення
+	1	0	0	0	1	перший доданок
	1	0	1	1	1	другий доданок
	1	0	1	0	0	сума



систем для реалізації операцій множення використовують операції додавання і зсуву.

### Двійкове ділення

Ділення – це операція, зворотна множенню. Двійкові числа, як і десяткові, ділять відповідно до правил множення і віднімання двійкових чисел:

Двійкове ділення	Десяткове ділення
$\begin{array}{r} 11001100 \overline{) 1100} \\ \underline{1100} \phantom{0000} \\ 01 \phantom{0000} \\ \underline{0} \phantom{0000} \\ 011 \phantom{000} \\ \underline{0} \phantom{000} \\ 110 \phantom{00} \\ \underline{0} \phantom{00} \\ 1100 \\ \underline{1100} \\ 0 \end{array}$	$\begin{array}{r} 204 \overline{) 12} \\ \underline{12} \phantom{00} \\ 84 \phantom{0} \\ \underline{84} \\ 0 \end{array}$

Реалізувати операцію ділення в мікропроцесорних системах більш складно, ніж операцію множення.

Розглянута вище система запису десяткових чисел за допомогою цифр “0” і “1” є *натуральним двійковим кодом*.

У мікропроцесорних системах для спрощення математичних операцій віднімання, ділення і деяких інших використовують також інші двійкові коди, які дістають від перетворення натурального двійкового коду. Це прямий, зворотний і додатковий коди.

## 1.3. Програма роботи

1. Вивчити особливості систем числення, ознайомитися з особливостями перетворення чисел з однієї системи числення в іншу.
2. Вивчити приклади перетворення чисел з однієї системи числення в інші.
3. Відповідно до свого варіанту виконати перетворення набору чисел.

## 1.4. Контрольні запитання

1. Що таке система числення?

2. Які бувають системи числення?
3. Чому в комп'ютерах застосовується двійкова система числення?
4. Скільки чисел можна закодувати за допомогою двох знаків у вісімковій системі числення?
5. Яке найбільше число можна закодувати за допомогою двох знаків у вісімковій системі числення?
6. За яким правилом переводяться числа з десяткової системи числення в двійкову?
7. Які особливості десяткової системи числення?
8. Які особливості шістнадцяткової системи числення?
9. Які особливості двійкової системи числення?
10. За яким правилом переводяться числа з десяткової системи числення в шістнадцяткову?

### 1.5. Завдання для самостійного виконання

Необхідно здійснити перетворення чисел відповідно до вашого варіанта:

- Перевести число *A* з двійкової системи числення у десяткову систему. Виконати перевірку, зробивши зворотне переведення.
- Перевести число *B* з вісімкової системи числення у десяткову систему. Виконати перевірку, зробивши зворотне переведення.
- Перевести число *C* з шістнадцяткової системи числення у десяткову систему. Виконати перевірку, зробивши зворотне переведення.
- Перевести число *D* з десяткової системи числення у шістнадцяткову систему. Виконати перевірку, зробивши зворотне переведення.
- Перевести число *E* з шістнадцяткової системи числення у вісімкову систему. Виконати перевірку, зробивши зворотне переведення.

Номер варіанта	Число <i>A</i>	Число <i>B</i>	Число <i>C</i>	Число <i>D</i>	Число <i>E</i>
1	1101111	157	19A	948	AB9
2	1000111	452	FD45	279	45F
3	1110110	7765	392E	346	ED3
4	1001111	4443	CA3	244	569

<b>5</b>	1000110	445	<i>57B2</i>	864	<i>4B8</i>
<b>6</b>	1111101	2771	<i>35F2</i>	646	<i>AD89</i>
<b>7</b>	11011	3352	<i>5B4</i>	212	<i>45D1</i>
<b>8</b>	101111111	7235	<i>78C</i>	356	<i>34DA</i>
<b>9</b>	11100011	2356	<i>541E</i>	831	<i>C567</i>
<b>10</b>	1101101	6674	<i>45BB</i>	953	<i>78B1</i>
<b>11</b>	11001001	5532	<i>91A3</i>	267	<i>49FA</i>
<b>12</b>	111010101	2341	<i>9B37</i>	373	<i>F9D6</i>
<b>13</b>	11110111	3451	<i>18C4</i>	731	<i>3C78</i>
<b>14</b>	11111011	2341	<i>97F</i>	852	<i>E42F</i>
<b>15</b>	101011001	5632	<i>26C</i>	368	<i>A491</i>
<b>16</b>	100101001	2317	<i>D29</i>	942	<i>895C</i>
<b>17</b>	11100001	4563	<i>AA22</i>	257	<i>34D5</i>
<b>18</b>	10011101	7612	<i>56F1</i>	538	<i>DC38</i>
<b>19</b>	110101101	2337	<i>91FA</i>	479	<i>67CA</i>
<b>20</b>	10010010	4567	<i>8717</i>	735	<i>D348</i>
<b>21</b>	1100110	7654	<i>F9D6</i>	462	<i>4A35</i>
<b>22</b>	1010101	7332	<i>AD89</i>	137	<i>6C42</i>
<b>23</b>	1011001	2167	<i>45D1</i>	752	<i>28A3</i>
<b>24</b>	101101	3654	<i>34DA</i>	368	<i>6B2C</i>
<b>25</b>	1110101	3265	<i>C567</i>	429	<i>35F2</i>
<b>26</b>	1000101	1432	<i>78B1</i>	529	<i>5B4</i>
<b>27</b>	1011101	6532	<i>49FA</i>	267	<i>78C</i>
<b>28</b>	1011010	2533	<i>F9D6</i>	634	<i>541E</i>
<b>29</b>	1000101	5671	<i>3C78</i>	827	<i>45BB</i>
<b>30</b>	10011011	2561	<i>E42F</i>	187	<i>91A3</i>

## **Робота 2. Розробка структури мікропроцесорних систем**

### **2.1. Мета роботи**

Навчитися розробляти структурні схеми мікропроцесорних пристроїв та систем.

### **2.2. Теоретичні відомості**

Мікропроцесорні системи по своїй складності, вимогам і функціям можуть значно відрізнятись більш надійними параметрами, обсягом програмних засобів, бути одно процесорними й багатопроцесорними, побудованими на одному типі мікропроцесорного набору або декількох, і т.д. У зв'язку з цим процес проектування може видозмінюватись в залежності від вимог, що висуваються до систем. Наприклад, процес проектування мікропроцесорного пристрою, буде складатись з розробки структури, написання програми та виготовлення пристрою.

При проектуванні багатопроцесорних мікропроцесорних систем, що містять кілька типів мікропроцесорних наборів, необхідно вирішувати питання організації пам'яті, взаємодії з процесорами, організації обміну між пристроями системи і зовнішнім середовищем, узгодження функціонування пристроїв, що мають різну швидкість роботи, і т.д. Нижче приведена зразкова послідовність етапів, типових для створення мікропроцесорної системи:

1. Формалізація вимог до системи.
2. Розробка структури й архітектури системи.
3. Розробка й виготовлення апаратних засобів і програмного забезпечення системи.
4. Комплексне налагодження і впровадження пристрою.

Етап 1. На цьому етапі складаються зовнішні специфікації, перелічуються функції системи, формалізується технічне завдання (ТЗ) на систему, формально викладаються задуми розробника в офіційній документації.

Етап 2. На даному етапі визначаються функції окремих пристроїв і програмних засобів, вибираються мікропроцесорні набори, на базі яких буде реалізована система, визначається взаємодія між

апаратними і програмними засобами, часові характеристики окремих пристроїв і програм.

Етап 3. Після визначення функцій, реалізованих апаратурою, і функцій, реалізованих програмами, схемотехніки і програмісти одночасно приступають до розробки і виготовлення відповідно дослідного зразка і програмних засобів. Розробка й виготовлення апаратури складаються з розробки структурних і принципівих схем, виготовлення прототипу, автономного налагодження. Розробка програм складається з розробки алгоритмів; написання тексту вихідних програм; трансляції вихідних програм в об'єктні програми; автономного налагодження.

Етап 4. Комплексне налагодження.

На кожному етапі проектування мікропроцесорної системи людьми можуть бути внесені похибки і прийняті невірні проектні рішення. Крім того, в апаратурі можуть виникнути дефекти.

На практиці при розробці проекту, розробляється загальна структура системи (пристрою), розглядаються зв'язки між структурними компонентами, визначається низка задач, які розв'язуватимуться кожним структурним компонентом. Після розробки загальної схеми, деталізується реалізація кожного структурного компоненту, алгоритму його роботи, задач, які він розв'язуватиме. Визначаються порти мікропроцесора (мікроконтролера), які будуть задіяні при проектуванні мікропроцесорної системи (пристрою) для взаємодії із зовнішнім середовищем.

За необхідністю визначаються сигнали, які керуватимуть роботою системи (пристрою), що розробляється. За результатами даного етапу розробляється схема електрична структурна.

Так для реалізації пристрою для контролю мікроклімату за допомогою давачів на базі мікропроцесора до складу узагальненої структури системи необхідно включити ряд пристроїв. Основними складовими цих пристроїв є такі:

- інтерфейс UART – для обміну даними між сторонами елементами;
- пристрої давачів для визначення фізичних величин стану мікроклімату;
- блоки підтримки протоколів.

Приклад такої структури наведено на рис. 2.1.



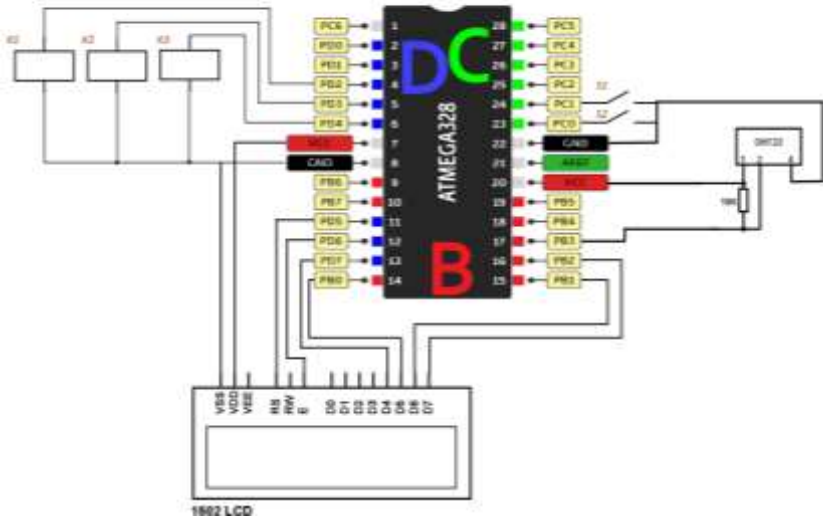


Рис. 2.1. Структурна схема мікропроцесорного пристрою контролю мікроклімату

Іншим варіантом виконання структурної схеми може бути схема, наведена на рис.2.2.

Далі проводять аналіз та обґрунтування вибору засобів, які будуть використовуватись для досягнення мети проектування. Даний етап присвячений визначенню конкретних мікропроцесорів та/або мікроконтролерів, які обираються для проектування системи (пристрою), обсягу та типу пам'яті, периферійних пристроїв, які пов'язані з мікропроцесорами, інтерфейсів тощо.

Для цього виконується огляд відомих засобів та обґрунтовується вибір одного з них. Якщо при проектуванні необхідно розробити новий мікропроцесор, то обґрунтовується недостатність відомих засобів для досягнення мети та визначаються переваги, якими повинен володіти новий мікропроцесор порівняно з відомими.

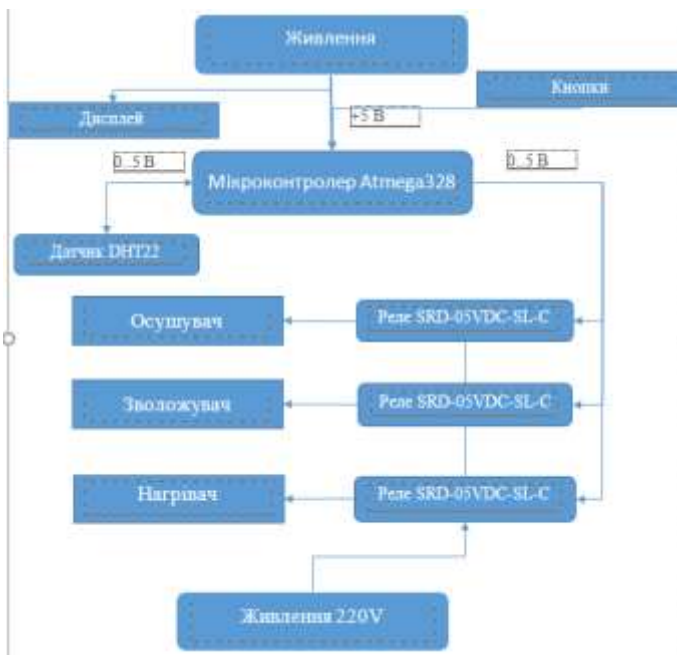


Рис. 2.2. Структурна схема мікропроцесорного пристрою

### 2.3. Програма роботи

1. Вивчити особливості розробки структурних схем мікропроцесорних пристроїв та систем.
2. Вивчити приклади розробки структурних схем мікропроцесорних пристроїв та систем.
3. Відповідно до свого варіанту розробити структурну схему пристрою та обрати технічні засоби для його реалізації.

### 2.4. Контрольні запитання

1. Що таке структурна схема?
2. Які етапи створення структурних схем?
3. Чому необхідно розробляти структурні схеми мікропроцесорних пристроїв?
4. Що є основою для розробки мікропроцесорних пристроїв?

5. Наведіть послідовність виконання побудови структурної схеми?
6. На якому етапі здійснюється розробка структурної схеми?
7. Як виконують побудову структурних схем?
8. Як відображаються зв'язки між різними елементами системи?
9. Чи можливо розробити дослідний зразок пристрою без розробки структурної схеми?
10. На основі чого виконується підбір технічних засобів для розробки дослідного зразка пристрою?

## **2.5. Завдання для самостійного виконання**

Розробити структурну схему для запропонованого варіанту мікропроцесорного пристрою.

1. Розробка мікропроцесорного пристрою вимірювання та стабілізації температури.
2. Розробка мікропроцесорного пристрою вимірювання та стабілізації вологості.
3. Розробка мікропроцесорного пристрою вимірювання та стабілізації мікроклімату.
4. Розробка мікропроцесорного пристрою вимірювання та стабілізації рівня сипучого матеріалу.
5. Розробка мікропроцесорного пристрою вимірювання та стабілізації рівня рідини.
6. Розробка мікропроцесорного пристрою визначення кута відхилення об'єкта від вертикалі.
7. Розробка мікропроцесорного пристрою визначення положення об'єкта в просторі.
8. Розробка мікропроцесорного пристрою вимірювання та стабілізації рівня рідини.
9. Розробка мікропроцесорного пристрою охоронної сигналізації рухомого об'єкта.
10. Розробка мікропроцесорного пристрою виведення інформації з клавіатури на дисплей.
11. Розробка мікропроцесорного пристрою розпізнавання об'єктів (система «свій-чужий»).
12. Розробка мікропроцесорного пристрою контролю доступу до об'єкта (можливі кілька варіантів).
13. Розробка мікропроцесорного пристрою вимірювання та тиску.

14. Розробка мікропроцесорного пристрою керування кроковим двигуном.
15. Розробка мікропроцесорного пристрою збереження вимірної інформації на зовнішній носій.
16. Розробка мікропроцесорного пристрою передачі вимірної інформації на web-сервер.
17. Розробка мікропроцесорної системи пожежної сигналізації.
18. Розробка мікропроцесорного пристрою регулювання освітленням в приміщенні.
19. Розробка мікропроцесорного пристрою обліку руху об'єктів.
20. Розробка мікропроцесорного пристрою «електронна панель приладів».
21. Розробка мікропроцесорного пристрою сортування дрібних об'єктів за
22. Розробка мікропроцесорного пристрою дистанційного регулювання поливу.
23. Розробка мікропроцесорного пристрою дистанційного блокування двигуна середньої потужності.
24. Розробка мікропроцесорного пристрою вимірювання та стабілізації витрати рідини.
25. Розробка мікропроцесорного пристрою контролю руху сипучого матеріалу.
26. Розробка мікропроцесорного пристрою вимірювання маси.
27. Розробка мікропроцесорного пристрою вимірювання та стабілізації швидкості обертання двигуна.
28. Розробка мікропроцесорного пристрою опитування давачів по протоколу RS485.
29. Розробка мікропроцесорного пристрою передачі вимірної інформації по протоколу RS485.
30. Розробка мікропроцесорного пристрою візуалізації інформації.
31. Розробка мікропроцесорного пристрою «ехолот».
32. Розробка мікропроцесорного пристрою «калькулятор».
33. Розробка мікропроцесорного пристрою «вольтметр».
34. Розробка мікропроцесорного пристрою «кораблик».
35. Розробка мікропроцесорного пристрою «автономна машинка».
36. Розробка мікропроцесорного пристрою «плоттер».
37. Розробка мікропроцесорного пристрою «металошукач».
38. Розробка мікропроцесорного пристрою «контроль викиду CO».

## **Робота 3. Розробка структурної схеми і алгоритму функціонування мікропроцесорної системи**

### **3.1. Мета роботи**

Отримати навички у створенні алгоритмів функціонування мікропроцесорних пристроїв та систем та побудові блок-схем алгоритмів.

### **3.2. Теоретичні відомості**

Алгоритм - сукупність правил, що однозначно визначають процес перетворення вхідних і проміжних даних у результат рішення задачі. Опис алгоритму являє собою загальну схему рішення задачі. Алгоритмічний процес це - процес послідовного перетворення конструктивних об'єктів, що проходить дискретними кроками (тобто зміна відбувається стрибкоподібно), кожний крок полягає в зміні одного конструктивного об'єкта іншим. Алгоритми характеризуються обчислювальною складністю і ємкісною складністю. За видом використовуваної обчислювальної моделі алгоритми діляться на послідовні (або детерміновані), паралельні (або недетерміновані), розподілені та ін.

Алгоритм може бути реалізований в мікроконтролері, якщо він містить тільки елементарні команди. Такими елементарними, тобто не потребуючими деталізації, можна вважати такі команди або операції:

- 1) початок, кінець;
- 2) список даних;
- 3) введення, виведення;
- 4) обчислювальні операції, реалізовані оператором присвоювання

Для алгоритму характерні наступні властивості:

- детермінованість, чи визначеність, тобто однозначність - його розуміння для будь-якого виконавця, що приводить до точного виконання однієї і тієї ж послідовності дій;

- результативність, чи спрямованість, тобто властивість досягнення за кінцеве число досить простих кроків шуканого результату розглянутої задачі;

- масовість, тобто придатність для рішення будь-якої задачі з деякого класу задач.

Розрізняють наступні способи представлення алгоритмів: текстуальний, операторний і графічний.

Найбільше поширення в даний час одержав графічний спосіб, при якому обчислювальний процес розчленовується на окремі операції, що відображаються у виді умовних графічних символів (блоків).

Блоки з'єднуються між собою в визначеній послідовності лініями чи стрілками. У середині блоків у виді формул чи тексту вказується інформація, що пояснює, характеризує виконуваними ними дії. Блоки звичайно мають наскрізну нумерацію. Номер ставиться у верхньому лівому куті блоку в розриві ліній.

Блок-схема алгоритму зображає послідовність блоків, з'єднаних між собою стрілками, які вказують послідовність виконання і зв'язок між блоками. В середині блоків записується їх короткий зміст.

Блок-схема - це спосіб представлення алгоритму в графічній формі, у вигляді геометричних фігур, сполучених між собою лініями (стрілками). Форма блока визначає тип дії, а текст всередині блоку дає детальне пояснення конкретної дії. Стрілки на лініях, що сполучають блоки схеми, вказують послідовність виконання команд, передбачених алгоритмом. Блок-схеми, за рахунок наочності спрощують створення ефективних алгоритмів, розуміння роботи вже створених, а як наслідок і їх оптимізацію. Існуючі стандарти на типи блоків дозволяють легко адаптувати алгоритми, створені у вигляді блок-схем до будь-яких існуючих на сьогоднішній день мов програмування.

Зображення блоків у алгоритмі, їх розміри, товщина ліній, кут нахилу ліній тощо, регламентуються Державним стандартом "Схеми алгоритмів, програм, даних і систем", а саме : 19.701-90 (ISO 5807-85).

Блоки у блок-схемі з'єднуються лініями потоків. У кожен блок може входити не менше однієї лінії, з блоку ж (окрім логічного) може виходити лише одна лінія потоку . З логічного блоку завжди виходять дві лінії потоку: одна у випадку виконання умови, інша - при її невиконанні. Бажано, щоб лінії потоку не перетинались.

Алгоритм може бути детальним, або спрощеним (деякі зрозумілі блоки можуть не записуватись, інакше алгоритм збільшується в розмірі).

Основні види блок-схем :

- лінійні (нерозгалужені);
- розгалужені;
- циклічні;

- з підпрограмами;
- змішані.

На рис. 3.1 наведено елемент блок-схеми алгоритму пристрою для визначення умов мікроклімату (структурна схема якого була наведена в практичній роботі 2).

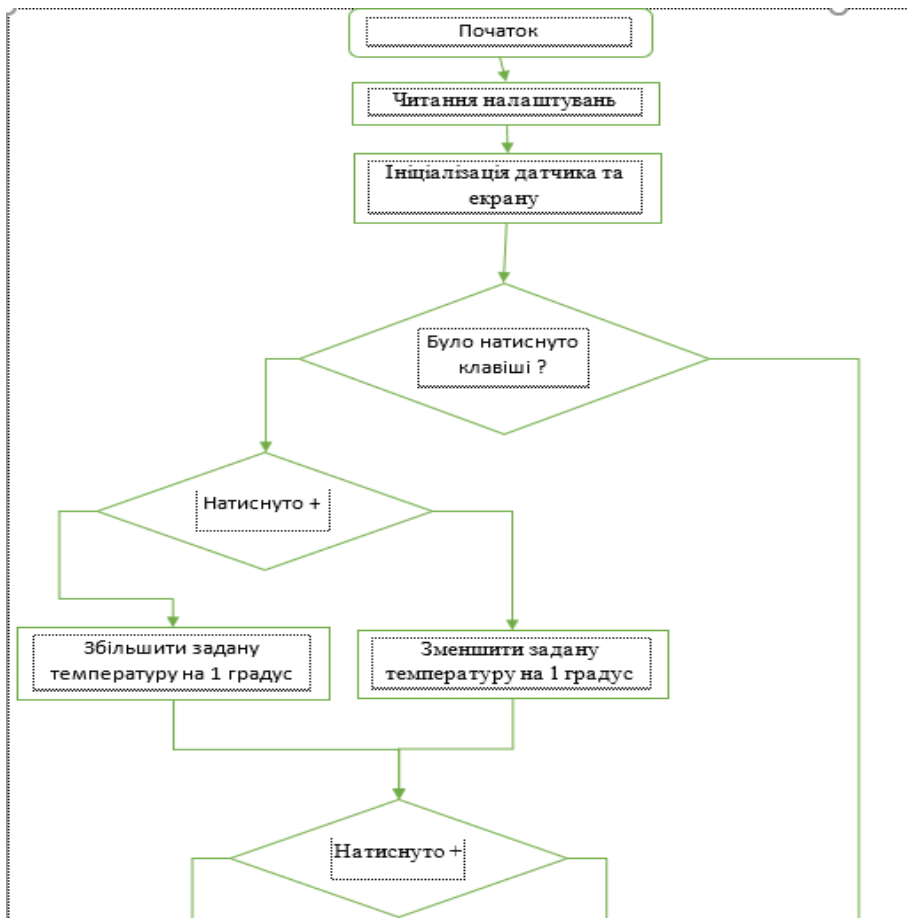


Рис. 3.1. Елемент блок-схеми алгоритму

№ з/п	Найменування	Позначення	Пояснення
1	Початок Кінець		Початок/кінець алгоритму, переривання процесу обробки даних (для початку — тільки вихід, для кінця — тільки вхід)
2	Введення - виведення		Уведення й виведення значень (один вхід, один вихід)
3	Процес		Виконання операції або групи операцій, у результаті яких змінюється значення, форма подання або розташування даних (один вхід, один вихід)
4	Розгалуження		Вибір напрямку виконання алгоритму залежно від деяких змінних умов (один вхід та тільки два виходи — так і ні)
5	Визначений процес		Програма, стандартна підпрограма
6	З'єднувачі		Вказівка зв'язку між перерваними лініями, що з'єднують блоки
7	Міжсторінковий з'єднувач		Вказівка зв'язку між перерваними лініями, що з'єднують блоки, розміщені на різних сторінках

Під час конструювання алгоритмів усі операції можна подати у вигляді комбінацій трьох типів операцій, так званих базових алгоритмічних структур.

Слідування



Операцію слідування подають у вигляді послідовності двох (або більше) простих операцій, що виконуються одна за одною. Якщо алгоритм складається лише з послідовності простих операцій, то його називають простим. Використовують також іншу назву — лінійний алгоритм.

Базові алгоритмічні конструкції (керівної структури)- способи керування процесами обробки даних.

Алгоритми, в яких використовується тільки структура слідування, називаються лінійними. Лінійний алгоритм описує обчислювальний процес, у якому етапи виконуються послідовно, тобто лінійно (один за одним незалежно від жодних умов).

Алгоритми, в основі яких лежить структура розгалуження, називаються розгалуженими. Такий алгоритм описує обчислювальний процес, у якому порядок обчислень залежить від вихідних умов або від проміжних результатів. Розгалужений алгоритм виконується по одному з кількох, заздалегідь передбачених напрямків, які називаються гілками. У кожному конкретному випадку процес реалізується тільки по одній гілці, тобто виконується одна або інша послідовність дій залежно від того, істинною чи хибною в певний момент є умова, що перевіряється. Є дві форми запису розгалужень — повна й неповна.

Алгоритми, в основі яких лежить структура повторення, називаються циклічними. Циклічний алгоритм описує обчислювальний процес, що містить однотипні, багаторазово повторювані послідовності команд. Виділяють цикли з відомою кількістю повторень та з невідомою кількістю повторень. Циклічні процеси, що допускають вкладеність одних циклів в інші, називають вкладеними циклами.

Операція розгалуження — це вказівка виконати одну з двох команд: команду 1 або команду 2 залежно від істинності чи хибності деякого твердження (логічного виразу). Якщо твердження правильне, то виконується команда 1 і на цьому виконання операції розгалуження закінчується. Якщо ж твердження хибне, то виконується команда 2 і на цьому виконання операції розгалуження закінчується.

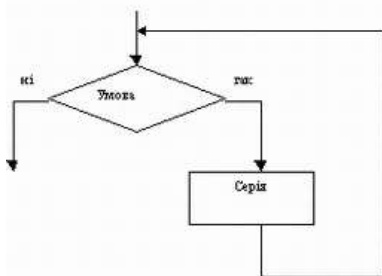
Окремим випадком розгалуження є неповне розгалуження, коли в разі хибного твердження жодна з операцій розгалуження не виконується. Повне розгалуження завжди можна подати у вигляді слідування двох неповних.

## Повторення (цикл)

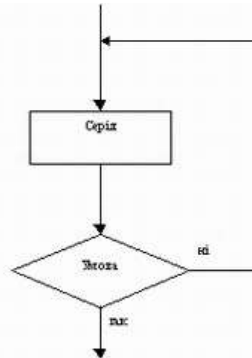
Розрізняють три типи циклів — ПОКИ, ДО та цикл із лічильником.

У структурі циклу ПОКИ для виконання операції повторення спочатку потрібно визначити, істинне чи хибне твердження (логічний вираз). Якщо твердження істинне, то виконується серія команд і відбувається повернення до визначення істинності твердження. Якщо твердження хибне, то виконання команд вважається закінченим. Отже, поки твердження істинне, відбувається повторне виконання команд і повернення до визначення істинності твердження.

У структурі циклу ДО спочатку виконуються команди, а потім визначається істинність твердження. Якщо твердження хибне, то знову виконуються команди і визначається істинність твердження. Якщо твердження істинне, то виконання вказівки вважається закінченим. Отже, виконання команд і визначення істинності твердження повторюються до настання істинності.



Цикл з передумовою



Цикл с післяумовою

У структурі циклу ПОКИ операція може не виконуватись жодного разу. У структурі циклу ДО операція буде виконана принаймні один раз.

Кожну з операцій, наприклад, операцію перевірки істинності твердження, у свою чергу, можна подати як сукупність більш простих операцій, об'єднаних у структури розглянутих типів.

### 3.3. Програма роботи

1. Вивчити вимоги до запропонованого процесу.
2. Ознайомитись з існуючими методиками розробки алгоритмів.
3. Розробити алгоритм функціонування пристрою та зобразити

його у вигляді блок-схеми.

### **3.4. Контрольні запитання**

1. Що таке алгоритм?
2. Чим регламентується зображення блок-схем?
3. Які є основні види блок-схем?
4. Чим особливі алгоритми для мікроконтролерів?
5. Що є основою для розробки алгоритму?
6. Які бувають алгоритми?
7. Які основні властивості алгоритму?
8. Які є основні способи зображення алгоритмів?
9. Які переваги графічного зображення алгоритмів?
10. Які є компоненти побудови алгоритмів?

### **3.5. Завдання для самостійного виконання**

Розробити блок-схему алгоритму функціонування для запропонованого варіанту мікропроцесорного пристрою.

1. Розробка мікропроцесорного пристрою вимірювання та стабілізації температури.
2. Розробка мікропроцесорного пристрою вимірювання та стабілізації вологості.
3. Розробка мікропроцесорного пристрою вимірювання та стабілізації мікроклімату.
4. Розробка мікропроцесорного пристрою вимірювання та стабілізації рівня силучого матеріалу.
5. Розробка мікропроцесорного пристрою вимірювання та стабілізації рівня рідини.
6. Розробка мікропроцесорного пристрою визначення кута відхилення об'єкта від вертикалі.
7. Розробка мікропроцесорного пристрою визначення положення об'єкта в просторі.
8. Розробка мікропроцесорного пристрою вимірювання та стабілізації рівня рідини.
9. Розробка мікропроцесорного пристрою охоронної сигналізації рухомого об'єкта.

10. Розробка мікропроцесорного пристрою виведення інформації з клавіатури на дисплей.
11. Розробка мікропроцесорного пристрою розпізнавання об'єктів (система «свій-чужий»).
12. Розробка мікропроцесорного пристрою контролю доступу до об'єкта (можливі кілька варіантів).
13. Розробка мікропроцесорного пристрою вимірювання та тиску.
14. Розробка мікропроцесорного пристрою керування кроковим двигуном.
15. Розробка мікропроцесорного пристрою збереження вимірної інформації на зовнішній носій.
16. Розробка мікропроцесорного пристрою передачі вимірної інформації на web-сервер.
17. Розробка мікропроцесорної системи пожежної сигналізації.
18. Розробка мікропроцесорного пристрою регулювання освітленням в приміщенні.
19. Розробка мікропроцесорного пристрою обліку руху об'єктів.
20. Розробка мікропроцесорного пристрою «електронна панель приладів».
21. Розробка мікропроцесорного пристрою сортування дрібних об'єктів за
22. Розробка мікропроцесорного пристрою дистанційного регулювання поливу.
23. Розробка мікропроцесорного пристрою дистанційного блокування двигуна середньої потужності.
24. Розробка мікропроцесорного пристрою вимірювання та стабілізації витрати рідини.
25. Розробка мікропроцесорного пристрою контролю руху сипучого матеріалу.
26. Розробка мікропроцесорного пристрою вимірювання маси.
27. Розробка мікропроцесорного пристрою вимірювання та стабілізації швидкості обертання двигуна.
28. Розробка мікропроцесорного пристрою опитування давачів по протоколу RS485.
29. Розробка мікропроцесорного пристрою передачі вимірної інформації по протоколу RS485.
30. Розробка мікропроцесорного пристрою візуалізації інформації.
31. Розробка мікропроцесорного пристрою «ехолот».
32. Розробка мікропроцесорного пристрою «калькулятор».

33. Розробка мікропроцесорного пристрою «вольтметр».
  34. Розробка мікропроцесорного пристрою «кораблик».
  35. Розробка мікропроцесорного пристрою «автономна машинка».
  36. Розробка мікропроцесорного пристрою «плоттер».
  37. Розробка мікропроцесорного пристрою «металошукач».
- Розробка мікропроцесорного пристрою «контроль викиду CO».