

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
Кафедра “Підйомно-транспортного та робототехнічного обладнання”

МЕТОДИЧНІ ВКАЗІВКИ ДО ЛАБОРАТОРНИХ ЗАНЯТЬ

з дисциплін "Електронні, мікропроцесорні та обчислювальні пристрої ГВС",
"Електронні, мікропроцесорні та обчислювальні пристрої ЛС",
"Електронні, мікропроцесорні та обчислювальні пристрої ПТО"

Перший (бакалаврський) рівень вищої освіти

Спеціальність – 131 ПРИКЛАДНА МЕХАНІКА

Освітня програма – *Інженерія логістичних систем,
Мехатроніка та промислові роботи*

Спеціальність – 133 Галузеве машинобудування

Освітня програма – *Підйомно-транспортні, дорожні,
меліоративні машини і обладнання*

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
Кафедра "Підйомно-транспортного та робототехнічного обладнання"

Михайлов Євген Павлович

МЕТОДИЧНІ ВКАЗІВКИ ДО ЛАБОРАТОРНИХ ЗАНЯТЬ

з дисциплін "Електронні, мікропроцесорні та обчислювальні пристрої ГВС",
"Електронні, мікропроцесорні та обчислювальні пристрої ЛС",
"Електронні, мікропроцесорні та обчислювальні пристрої ПТО"

Перший (бакалаврський) рівень вищої освіти

Спеціальність – 131 ПРИКЛАДНА МЕХАНІКА

Освітня програма – *Інженерія логістичних систем,
Мехатроніка та промислові роботи*

Спеціальність – 133 Галузеве машинобудування

Освітня програма – *Підйомно-транспортні, дорожні,
меліоративні машини і обладнання*

Затверджено
на засіданні кафедри
підйомно-транспортного і
робототехнічного обладнання
Протокол № 1 від 26.08.2021 р.

ОДЕСА 2021

Методичні вказівки до лабораторних занять з дисциплін "Електронні, мікропроцесорні та обчислювальні пристрої ГВС", "Електронні, мікропроцесорні та обчислювальні пристрої ЛС" та "Електронні, мікропроцесорні та обчислювальні пристрої ПТО " для здобувачів першого (бакалаврського) рівня вищої освіти, спеціальності: 131 - Прикладна механіка, освітні програми: – Мехатроніка та промислові роботи, Інженерія логістичних систем, спеціальності: 133 - Галузеве машинобудування, освітні програми: – Підйомно-транспортні, дорожні, меліоративні машини і обладнання. / Укл.: Михайлов Є.П. – Одеса: ОП, 2021. - 61 с.

Укладач:

Михайлов Є.П. доц. кафедри підйомно-транспортного і робототехнічного обладнання

У методичних вказівках наведені лабораторні заняття, метою яких є ознайомлення та дослідження систем керування робототехнічних комплексів, логістичних систем та підйомно-транспортного обладнання на основі електронних, мікропроцесорних та обчислювальних пристроїв. На заняттях розглядаються питання використання програмованих логічних контролерів на основі мікропроцесорної техніки та різних електронних компонент для вирішування завдань керування рухом.

Зміст

Вступ	4
Лабораторне заняття 1. Дослідження принципів обчислювання параметрів пасивних електронних схем	5
Лабораторне заняття 2. Дослідження принципів обчислювання параметрів активних електронних схем	7
Лабораторне заняття 3. Дослідження принципів обчислювання параметрів аналогових інтегральних мікросхем	9
Лабораторне заняття 4. Дослідження принципів визначення параметрів дискретних інтегральних мікросхем	11
Лабораторне заняття 5. Дослідження принципів обчислювання параметрів цифро-аналогових перетворювачів	13
Лабораторне заняття 6. Дослідження принципів обчислювання параметрів аналого-цифрових перетворювачів	15
Лабораторне заняття 7. Дослідження принципів складання програм керування для мікроконтролерів	17
Лабораторне заняття 8. Дослідження принципів складання програм обробки даних для мікроконтролерів	21
Лабораторне заняття 9. Дослідження принципів складання програм у вигляді функціональної схеми	24
Лабораторне заняття 10. Дослідження принципів складання програм циклового керування	26
Лабораторне заняття 11. Дослідження принципів складання програм позиційного керування	28
Література	31
Додаток 1	32
Додаток 2	34
Додаток 3	39

Вступ

Широке використання електронних, мікропроцесорних та обчислювальних пристроїв зв'язано з можливістю широкого застосування засобів комп'ютерного керування в різних галузях промисловості. Одним з завдань, які ефективно вирішуються за допомогою таких пристроїв є програмоване керування рухом, що зокрема широко використовується в підйомно-транспортному обладнанні та в гнучких виробничих системах.

В наведених лабораторних заняттях з дисциплін "Електронні, мікропроцесорні та обчислювальні пристрої ПТО" та "Електронні, мікропроцесорні та обчислювальні пристрої ГВС" розглянуті питання застосування електронних пристроїв, а також використання пристроїв керування на основі мікроконтролерів та програмованих логічних контролерів для вирішення завдань програмованого керування рухом.

Ці пристрої мають прості мови програмування, такі як контактний та функціональний плани, що дозволяє використовувати їх фахівцям, які не мають спеціальної підготовки в галузі обчислювальної техніки. З іншого боку ці пристрої мають вмонтовані функції керування переміщенням. Для використання таких функцій програма складається за допомогою так званих асистентів, які значно спрощують упорядкування програм.

Метою наведених лабораторних занять є знайомство з електронними пристроями та системами керування, а також отримання навиків використання засобів програмування таких систем.

На лабораторних заняттях використовується безкоштовне програмне забезпечення: LOGO!Soft Comfort (Demo), середовище розробки Arduino IDE.

Лабораторне заняття 1. Дослідження принципів обчислювання параметрів пасивних електронних схем

Мета заняття: Дослідження принципів обчислювання параметрів пасивних електронних схем, що включають резистивні подільники напруги. Для проведення дослідження треба:

- знати принципи обчислювання параметрів електронних схем на прикладі подільника напруги та схеми обмежування струму;
- вміти використовувати електронні компоненти;
- навчитися проводити обчислювання параметрів електронних схем.

1. Теоретическая частина

Схема резистивного подільника напруги та залежність вихідної напруги від параметрів подільника наведені на рис. 1

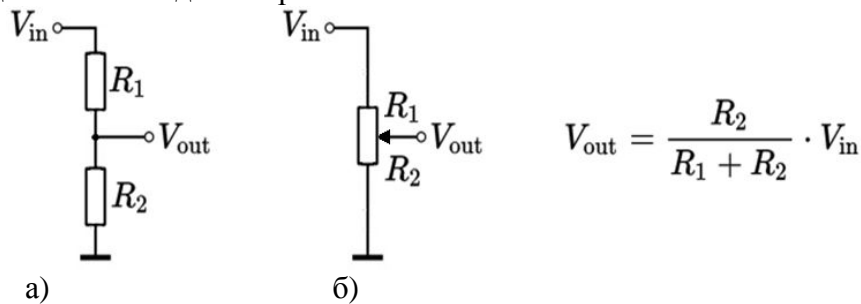


Рис. 1. Схема подільника напруги та залежність вихідної напруги від параметрів подільника

Якщо світлодіод використовується як індикатор, то схему його підключення можна представити у вигляді рис. 2.

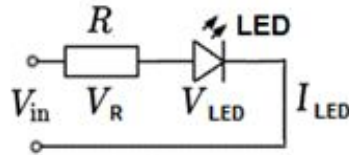


Рис. 2. Схема підключення світлодіода

Опір резистора R визначається падінням напруги на світлодіоді U_{LED} , допустимим струмом світлодіода I_{LED} та напругою живлення U

$$R = (V - V_{LED}) / I_{LED} .$$
$$I_{LED} = (V - V_{LED}) / . R$$

2. Практична частина

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату та відповідні електронні компоненти. Для вимірювання напруги на виході подільника використовується принципова схема, що наведена на рис. 3.

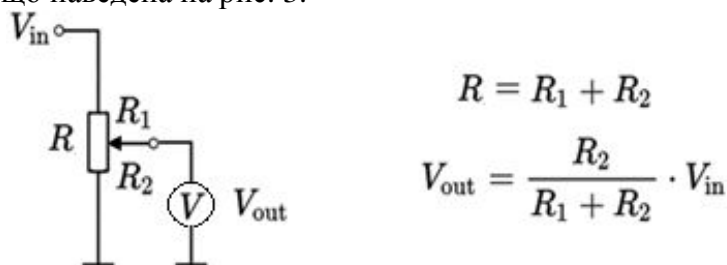
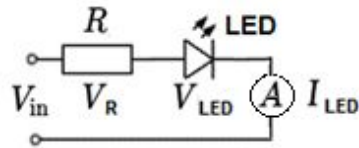


Рис. 3. Схема вимірювання напруги на виході подільника

Для вимірювання струму світлодіода використовується схема, що наведена на рис. 4.



$$R = (U - U_{LED}) / I_{LED} .$$

$$I_{LED} = (U - U_{LED}) / R .$$

Рис. 4. Схема вимірювання струму світлодіода

Приклади розв'язання задач з теми заняття

Якщо треба отримати $V_{out} / V_{in} = 0,2$, а $R1 + R2 = 10 \text{ кОм}$, то виходячи з того, що

$$R2 = (R1 + R2) V_{out} / V_{in}$$

отримаємо $R1 = 8 \text{ кОм}$ та $R2 = 2 \text{ кОм}$.

Якщо використовувати напругу живлення $U = 5 \text{ В}$, то для стандартного світлодіода ($U_{LED} = 2 \text{ В}$, $I_{LED} = 20 \text{ мА}$) отримаємо $R = 150 \text{ Ом}$.

Завдання 1

Знайти параметри подільника напруги $R1$ та $R2$ для заданих значень коефіцієнта передачі V_{out} / V_{in} та загального опору $R1 + R2$.

V_{out} / V_{in}	$R1 + R2, \text{кОм}$	$R1, \text{кОм}$	$R2, \text{кОм}$
0,1	1		
0,5	50		
0,25	270		
0,01	100		

Зібрати схему з визначеними параметрами, перевірити коефіцієнт передачі подільника.

Завдання 2

Знайти V_{out} , для заданих значень V_{in} , $R1$ та $R2$.

$V_{in}, \text{В}$	$R1, \text{кОм}$	$R2, \text{кОм}$	$V_{out}, \text{В}$
24	6,8	2,2	
12	750	5,1	
5	4,7	0,27	
10	1,2	0,1	

Зібрати схему з визначеними параметрами, перевірити напругу на виході подільника.

Завдання 3

Знайти значення обмежувального опору стандартного світлодіода для напруги живлення U , якщо $U_{LED} = 2 \text{ В}$, $I_{LED} = 20 \text{ мА}$.

$U, \text{В}$	$R, \text{Ом}$
12	
24	
10	
220	

Зібрати схему з визначеними параметрами, перевірити струм світлодіода.

Контрольні питання

1. Визначити, для чого використовують подільники напруги?
2. Описати, як здійснюється регулювання напруги за допомогою подільника?
3. Назвати, для чого використовують світлодіоди?
4. Описати, як треба включати світлодіоди, щоб вони світилися?
5. Назвати, чому треба обмежувати струм світлодіода?

Лабораторне заняття 2. Дослідження принципів обчислювання параметрів активних електронних схем

Мета заняття: Дослідження принципів обчислювання параметрів активних електронних схем на основі транзисторів.

Для проведення дослідження треба:

- знати принципи обчислювання параметрів електронних схем на прикладі подільника напруги та схеми обмежування струму.
- вміти використовувати електронні компоненти.
- навчитися проводити обчислювання параметрів електронних схем.

1. Теоретическая частина

Біполярні та польові транзистори використовуються у підсилювачах напруги та струму. На рис 1 наведені схеми підсилювальних каскадів на біполярних транзисторах із спільним емітером (інвертуючий каскад, рис 1, а) та із спільним колектором (неінвертуючий каскад, рис 1, б).

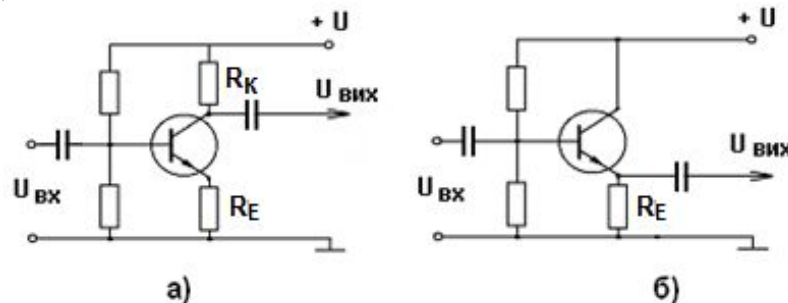


Рис 1. Схеми каскадів підсилювання: а - із спільним емітером, б - із спільним колектором

Струм колектору, бази та емітеру пов'язані такими залежностями:

$$I_E = I_B + I_K, \quad I_K = \beta I_B, \quad I_K = \alpha I_E, \quad \beta = \alpha / (1 + \alpha),$$

де β – коефіцієнт підсилювання по струму, що зазвичай знаходиться у діапазоні 10 - 300.

Схема із спільним емітером використовується як підсилювач напруги, при цьому коефіцієнт підсилювання дорівнює:

$$K_U = R_K / R_E.$$

Схема із спільним колектором використовується як підсилювач струму, при цьому коефіцієнт підсилювання дорівнює:

$$K_I = I_E / I_B = 1 + \beta.$$

2. Практична частина

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату та відповідні електронні компоненти. Для вимірювання коефіцієнта підсилювання використовується принципова схема, що наведена на рис. 2.

Для створення вхідного сигналу зі змінною напругою (послідовність імпульсів) частотою 1 кГц використовується контролер Arduino та подільник напруги для встановлення амплітуди сигналу.

Фрагмент програми, що створює таку послідовність імпульсів на виході 4 має вигляд:

```
digitalWrite(4, HIGH); // Виводимо на вихід 4 сигнал високого рівня
delay(500);           // Затримка 500 мс
digitalWrite(4, LOW); // Виводимо на вихід 4 сигнал низького рівня
delay(500);           // Затримка 500 мс
```

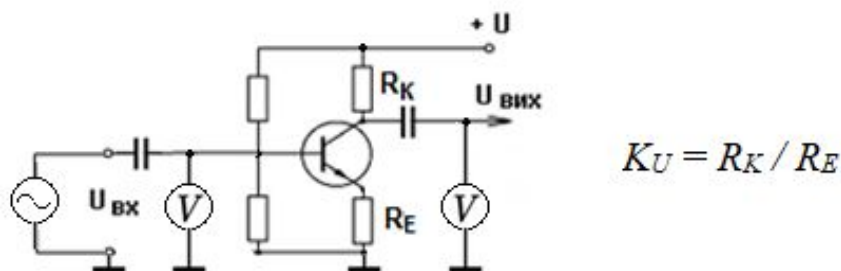


Рис. 2. Схема вимірювання коефіцієнта підсилювання змінного струму для каскаду підсилювання із спільним емітером

Для створення вхідного сигналу зі змінною напругою частотою 1 кГц використовується контролер Arduino та подільник напруги для встановлення амплітуди сигналу.

Приклади розв'язання задач з теми заняття

Якщо треба отримати коефіцієнт підсилювання для схеми із спільним емітером K_U , при $R_K = 100 \text{ кОм}$ та $R_E = 10 \text{ кОм}$, то виходячи з того, що

$$K_U = R_K / R_E$$

отримаємо $K_U = 10$.

Завдання 1

Знайти коефіцієнт підсилювання для схеми із спільним емітером K_U , для заданих значень R_K та R_E .

$R_K, \text{кОм}$	$R_E, \text{кОм}$	K_U
6,8	2,2	
750	5,1	
4,7	0,27	
1,2	0,1	
100	1,0	

Зібрати схему з визначеними параметрами, перевірити роботу підсилювача.

Контрольні питання

1. Описати, як визначити коефіцієнт підсилювання схеми із спільним емітером?
2. Описати, як визначити коефіцієнт підсилювання схеми із спільним колектором?
3. Описати, як визначити коефіцієнт підсилювання змінного струму?
4. Визначити, як здійснити вимірювання напруги на виході підсилювача?
5. Визначити, як здійснити формування вхідного сигналу?

Лабораторне заняття 3. Дослідження принципів обчислювання параметрів аналогових інтегральних мікросхем

Мета заняття: Дослідження принципів обчислювання параметрів аналогових інтегральних мікросхем.

Для проведення дослідження треба:

- знати принципи обчислювання параметрів операційних підсилювачей;
- вміти використовувати аналогові мікросхеми;
- навчитися проводити обчислювання параметрів електронних схем.

1. Теоретическая частина

На рис. 1 наведені інвертуючий (а), неінвертуючий (б) та диференціальний (в) операційні підсилювачі з відповідними коефіцієнтами підсилювання (K_i , K_n та K_d).

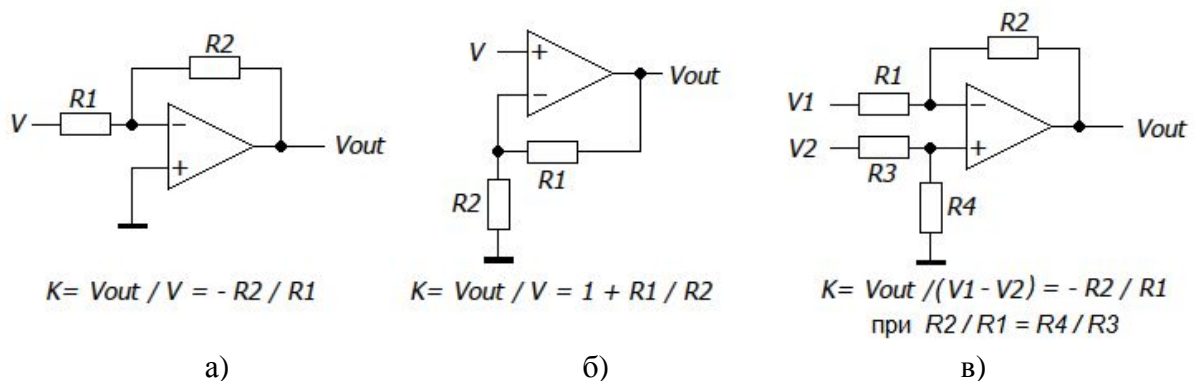


Рис. 1. Операційні підсилювачі: інвертуючий (а), неінвертуючий (б), диференціальний (в)

2. Практична частина

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату та відповідні електронні компоненти. Для вимірювання коефіцієнта підсилювання використовується принципова схема, що наведена у попередній лабораторній роботі.

Приклади розв'язання задач з теми заняття

Для інвертуючого операційного підсилювача у якого $R1 = 100 \text{ кОм}$, $R2 = 10 \text{ кОм}$, виходячи з того, що

$$K = V_{out} / (V1 - V2) = - R2 / R1,$$

отримаємо $K = - 10$.

Завдання 1

Знайти коефіцієнт підсилювання інвертуючого операційного підсилювача, якщо відомі $R1$ та $R2$.

K	$R1, \text{кОм}$	$R2, \text{кОм}$
	150	1
	10	5
	275	25
	100	0,1

Завдання 2

Знайти коефіцієнт підсилювання неінвертуючого операційного підсилювача, якщо відомі $R1$ та $R2$.

<i>K</i>	<i>R1, кОм</i>	<i>R2, кОм</i>
	150	1
	10	5
	275	25
	100	0,1

Завдання 3

Знайти коефіцієнт підсилювання диференціального операційного підсилювача, якщо відомі *R1* та *R2*.

<i>K</i>	<i>R1, кОм</i>	<i>R2, кОм</i>
	150	1
	10	5
	275	25
	100	0,1

Зібрати схему з визначеними параметрами, перевірити роботу підсилювачів.

Контрольні питання

1. Описати, як визначити коефіцієнт підсилювання інвертуючого операційного підсилювача?
2. Описати, як визначити коефіцієнт підсилювання неінвертуючого операційного підсилювача?
3. Описати, як визначити коефіцієнт підсилювання диференціального операційного підсилювача?
4. Визначити, як здійснити вимірювання напруги на виході операційного підсилювача?
5. Визначити, як здійснити формування вхідного сигналу?

Лабораторне заняття 4. Дослідження принципів визначення параметрів дискретних інтегральних мікросхем

Мета заняття: Дослідження принципів визначення параметрів дискретних інтегральних мікросхем.

Для проведення дослідження треба:

- знати принципи створення систем логічного керування;
- вміти використовувати цифрові мікросхеми;
- навчитися складати прості схеми керування з використанням логічних мікросхем.

1. Теоретическая частина

Логічні елементи - це елементарні комбінаційні пристрої з одним виходом, що реалізують, як правило, одну логічну функцію за законами алгебри Буля та використовують для створення систем логічного керування.

На рис. 1 наведені основні логічні елементи:

- логічне І (а), на виході якого маємо 1, якщо на усіх входах 1,
- логічне АБО (б), на виході якого маємо 1, якщо хоч на одному вході 1,
- інверсія входу (в), коли вхідний сигнал змінюється на протилежний,
- інверсія виходу (г), коли вихідний сигнал змінюється на протилежний,
- RS-тригер (д), вихід якого встановлюється в 1, якщо на вхід S поступає 1, та скидається в 0, якщо на вхід R поступає 1.

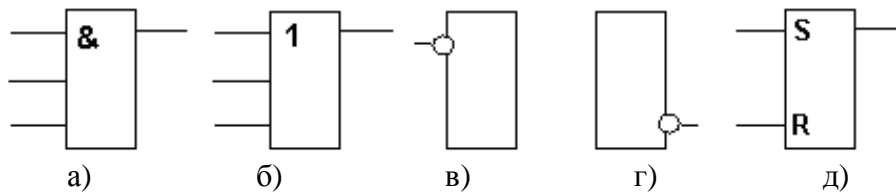


Рис.1. Основні логічні елементи

2. Практична частина

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату та відповідні мікросхеми або, у разі дистанційного навчання, створити відповідну програму за допомогою мови LOGO!Soft Comfort, яка використовує логічні функції (Додаток 1 та).

Приклади розв'язання задач з теми заняття

На рис. 2 наведені приклади використання логічних мікросхем для блокування від одночасного натискання кнопок "SB1" та "SB2" (рис. 2, а), а також включення за допомогою кнопки "SB1" та виключення за допомогою кнопки "SB2" (рис. 2, б).

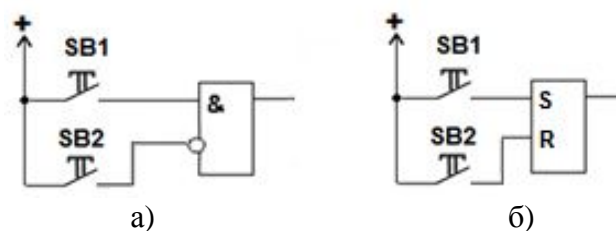


Рис. 2. Приклади використання логічних мікросхем

Завдання 1

Скласти схему керування, яка включає двигун кнопкою "Пуск" та виключає кнопкою "Стоп", з використанням логічних елементів та RS-тригера.

Передбачити блокування від одночасного натискання кнопок "Пуск" та "Стоп" (пріоритет кнопки "Стоп").

Перевірити роботу програми на стенді або, у разі дистанційного навчання, створити та перевірити відповідну програму за допомогою мови LOGO!Soft Comfort.

Контрольні питання

1. Визначити, які функції виконує схема логічного І?
2. Описати, які функції виконує схема логічного АБО?
3. Описати, які функції виконує схема інверсії?
4. Описати, які функції виконує схема RS-тригера?
5. Описати, як перевірити роботу програми в LOGO!Soft Comfort ?

Лабораторне заняття 5. Дослідження принципів обчислювання параметрів цифро-аналогових перетворювачів

Мета заняття: Дослідження принципів обчислювання параметрів цифро-аналогових перетворювачів.

Для проведення дослідження треба:

- знати принципи роботи цифро-аналогових перетворювачів (ЦАП);
- вміти здійснювати перетворення даних в ЦАП;
- навчитися визначати основні параметри ЦАП.

1. Теоретическая частина

До найважливіших параметрів та характеристик ЦАП належать розрядність (кількість двійкових розрядів), діапазон ($U_{\min} \dots U_{\max}$) та рівні вхідних (вихідних) сигналів, точність перетворення, час перетворення та встановлення результату.

Точність перетворення ΔU визначається як відношення діапазону вимірювання ($U_{\max} - U_{\min}$) до максимальної кількості рівнів перетвореного сигналу ($q = 2^N$, де N – кількість двійкових розрядів), при цьому діапазон отриманих значень складає від 0 до $2^N - 1$.

$$\Delta U = (U_{\max} - U_{\min}) / q.$$

Якщо для визначення аналогового сигналу використовується двополярний діапазон, наприклад, $\pm 10 \text{ В}$, що має як позитивні, так і негативні значення, то для визначення негативних значень чисел використовується доповняльний код, який можна отримати так: інвертувати модуль числа у двійковому вигляді («перше доповнення») і додати одиницю («друге доповнення»). Математично доповняльний код $x_{\text{дон}}$ від двійкового числа x дорівнює:

$$x_{\text{дон}} = 2^{N-1} - x$$

Для більш простого запису двійкових чисел використовується шістнадцяткова система числення, що має діапазон чисел від 0 до $2^4 - 1$.

Більш детально різні типи цифро-аналогових перетворювачів розглянуті в [1].

Часто функції цифро-аналогових перетворювачів здійснюються за допомогою сигналу з широтно-імпульсною модуляцією (рис. 1).

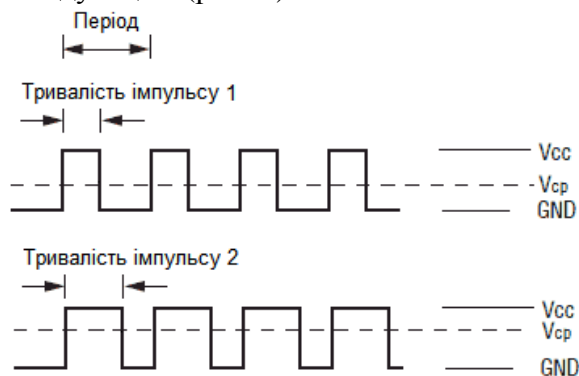


Рис. 1. Вигляд сигналу з широтно-імпульсною модуляцією

Цей сигнал являє собою послідовність імпульсів з постійною частотою та змінною тривалістю. При зміні тривалості імпульсів змінюється середнє значення напруги. Для отримання середнього значення використовують схеми інтегрування.

В системах керування такий сигнал подається на індуктивне навантаження (котушки індуктивності, електродвигуни), струм яких здійснює інтегрування напруги.

Контролери Arduino використовують управління виходом ШІМ за допомогою функції `analogWrite()`.

Ця функція має два аргументи: номер виходу, на який виводиться сигнал ШІМ, і число в діапазоні від 0 до 255 , яке задає пропорційну тривалість імпульсу ШІМ.

Наприклад, функція

analogWrite(7,127);

встановлює на виході 7 напругу на рівні 0,5 від напруги живлення.

2. Практична частина

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату та контролер Arduino, у разі дистанційного навчання, створити відповідну програму за допомогою емулятора UnoArduSim (Додаток 2).

Приклади розв'язання задач з теми заняття

Для ЦАП з розрядністю $N=10$ та діапазоном вимірювання ± 5 В маємо таку точність перетворення коду у вихідний сигнал

$$\Delta U = (U_{\max} - U_{\min}) / q = 10 \text{ В} / 1024 = 0,0097656 \text{ В}.$$

Яке перетворення масштабу K_m треба зробити, щоб отримати значення для діапазону ± 5 В та точності вихідних сигналів $0,01$ В для ЦАП з розрядністю $N=10$.

$$K_m = \Delta U / \Delta U_{\text{вим}} = 0,97656$$

Перелік завдань для розв'язання

Завдання 1

Знайти точність перетворення цифрового коду у напругу для ЦАП з вказаними розрядністю та діапазоном вихідних сигналів.

ΔU	$(U_{\max} - U_{\min}), \text{ В}$	N
	10	10
	± 5	12
	1	16

Завдання 2

Представити у двійковій формі такі числа
 $250, -250, 1024, -1024, 27746, -27746$.

Завдання 3

Визначити діапазон представлення даних у двійковій та десятковій формі, які отримує 12-розрядний ЦАП для біполярних вихідних сигналів (представлення даних як цілі числа зі знаком).

Завдання 4

Створити програму для послідовного виведення за допомогою ШІМ сигналу з напругою $U_{\text{вих}} = 0,2 U_{\text{ж}}$, $U_{\text{вих}} = 0,4 U_{\text{ж}}$, $U_{\text{вих}} = 0,6 U_{\text{ж}}$, $U_{\text{вих}} = 0,8 U_{\text{ж}}$ з затримкою 2 с, де $U_{\text{ж}}$ – напруга живлення.

Перевірити роботу програми на стенді або, у разі дистанційного навчання, створити та перевірити відповідну програму за допомогою емулятора UnoArduSim.

Контрольні питання

1. Описати, у яких системі числення задаються значення з ЦАП?
2. Визначити, який діапазон чисел має ЦАП з розрядністю $n=10$?
3. Описати, як знайти коефіцієнт перетворення масштабу?
4. Описати, як визначаються негативні числа у двійковій системі числення?
5. Описати, як створити сигнал змінної напруги за допомогою широтно-імпульсної модуляції?

Лабораторне заняття 6. Дослідження принципів обчислювання параметрів аналого-цифрових перетворювачів

Мета заняття: Дослідження принципів обчислювання параметрів аналого-цифрових перетворювачів (АЦП).

Для проведення дослідження треба:

- знати принципи роботи АЦП;
- вміти здійснювати перетворення даних в АЦП;
- навчитися визначати основні параметри АЦП.

1. Теоретическая частина

До найважливіших параметрів та характеристик АЦП належать розрядність (кількість двійкових розрядів), діапазон ($U_{\min} \dots U_{\max}$) та рівні вхідних (вихідних) сигналів, точність перетворення, час перетворення та встановлення результату.

Точність перетворення ΔU визначається як відношення діапазону вимірювання ($U_{\max} - U_{\min}$) до максимальної кількості рівнів перетвореного сигналу ($q = 2^N$, де N – кількість двійкових розрядів), при цьому діапазон отриманих значень складає від 0 до $2^N - 1$.

$$\Delta U = (U_{\max} - U_{\min}) / q.$$

Якщо для визначення аналогового сигналу використовується двохполярний діапазон, наприклад, $\pm 10 \text{ В}$, що має як позитивні, так і негативні значення, то для визначення негативних значень чисел використовується доповняльний код, який можна отримати так: інвертувати модуль числа у двійковому вигляді («перше доповнення») і додати одиницю («друге доповнення»). Математично доповняльний код $x_{\text{дон}}$ від двійкового числа x дорівнює:

$$x_{\text{дон}} = 2^{N-1} - x$$

Для більш простого запису двійкових чисел використовується шістнадцяткова система числення, що має діапазон чисел від 0 до $2^4 - 1$.

Більш детально різні типи аналого-цифрових перетворювачів розглянуті в [1].

Контролери Arduino мають у своєму складі декілька аналогових входів (залежить від типу контролера), які використовують АЦП з розрядністю $N=10$ та діапазоном вимірювання $0 - U_{\text{ж}}$, де $U_{\text{ж}}$ – напруга живлення (5 В) [1].

Для зчитування у змінну значення сигналу з аналогового входу pin використовується функція analogRead(), яка видають значення від 0 до 1023 .

Функція має такий синтаксис:

```
value = analogRead(pin);
```

де value – змінна, куди записується результат опитування.

У програмах керування часто використовується зміна масштабу.

Таку функцію треба використовувати, наприклад, для ручного керування сервопривода, на який подаються значення від 0 до 180 , за допомогою потенціометра, підключеного до аналогового входу, що видає значення від 0 до 1023 .

Для зміни масштабу можна використати функцію пропорційного перетворення значень від одного діапазону до другого map() та функцію обмеження діапазону constrain().

Функція зміни масштабу map() має такий синтаксис:

```
valNew = map(valOld, fromLow, fromHigh, toLow, toHigh);
```

Змінна valOld з нижньою fromLow і верхньою fromHigh межами діапазону перетворюється у змінну valNew з нижньою toLow і верхньою toHigh межами діапазону.

2. Практична частина

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату та контролер Arduino, у разі дистанційного навчання, створити відповідну програму за допомогою емулятора UnoArduSim (Додаток 2).

Приклади розв'язання задач з теми заняття

Для АЦП з розрядністю $N=10$ та діапазоном вимірювання ± 5 В маємо таку точність перетворювання напруги у цифровий код

$$\Delta U = (U_{\max} - U_{\min}) / q = 10 \text{ В} / 1024 = 0,0097656 \text{ В}.$$

Яке перетворення масштабу K_m треба зробити, щоб отримати значення для діапазону ± 5 В та точності вимірювання $0,01$ В для АЦП з розрядністю $N=10$.

$$K_m = \Delta U / \Delta U_{\text{вим}} = 0,97656$$

Перелік завдань для розв'язання

Завдання 1

Знайти точність перетворювання напруги у цифровий код для АЦП з вказаними розрядністю та діапазоном вимірювання.

ΔU	$(U_{\max} - U_{\min}), \text{ В}$	N
	10	10
	± 5	12
	1	16

Завдання 2

Скласти програму опитування аналогового входу 2, до якого підключений потенціометр (подільник напруги) з періодом 2 с. Отриману інформацію вивести на Монітор порта.

Перевірити роботу програми на стенді або, у разі дистанційного навчання, створити та перевірити відповідну програму за допомогою емулятора UnoArduSim.

Контрольні питання

1. Визначити, у яких системі числення отримаємо значення з АЦП?
2. Описати, який діапазон чисел має АЦП з розрядністю $N=10$?
3. Визначити, як знайти коефіцієнт перетворення масштабу?
4. Визначити, яка функція використовується для опитування аналогового входу у контролера Arduino?
5. Визначити, яка функція використовується для перетворення масштабу?

Лабораторне заняття 7. Дослідження принципів складання програм керування для мікроконтролерів

Мета заняття: Дослідження принципів складання програм керування для мікроконтролерів.

Для проведення дослідження треба:

- знати принципи складання програм для мікроконтролерів;
- вміти використовувати мікроконтролери для керування;
- навчитися складати прості програми мікроконтролерів з використанням мови С.

1. Теоретическая частина

Розглянемо приклади програмування для контролерів Arduino, що використовують мову С [1].

Контролери Arduino часто використовують для керування різними робототехнічними пристроями, тому мають вмонтовані функції керування рухом та обробки даних.

Так для керування швидкості обертання двигунів постійного струму використовується управління виходом ШІМ за допомогою функції `analogWrite()`. Ця функція має два аргументи: номер виходу, на який виводиться сигнал ШІМ, і число в діапазоні від 0 до 255, яке задає пропорційну тривалість імпульсу ШІМ.

Наприклад, функція
`analogWrite(7,127);`

встановлює швидкість обертання на рівні 0,5 від напруги живлення двигуна постійного струму, що підключений до виходу 7.

Контролери Arduino мають багато додаткових модулів і пристрів, призначених для здійснення переміщення, наприклад, для сервоприводів та крокових двигунів.

Для спрощення програмування для таких пристроїв створені бібліотеки з відповідними функціями.

Так для керування сервоприводом використовується бібліотека `Servo`, яка має наступні функції.

Функція підключення сервопривода до вказаного виходу контролера:

`servo.attach(pin)`

де `servo` – змінна типа `Servo`,

`pin` – номер виходу, який здійснює керування приводом.

Функція повернення приводу на вказаний кут:

`servo.write(angle)`

де `servo` – змінна типа `Servo`,

`angle` – кут повороту від 0 до 180, відповідає куту повороту в градусах.

Для керування кроковим двигуном використовується бібліотека `Stepper Library` в Arduino IDE.

Ця бібліотека має наступні функції.

Функція встановлення параметрів крокового двигуна:

`Stepper myStepper = Stepper(steps, pin1, pin2, pin3, pin4);`

де `myStepper` – ім'я крокового двигуна,

`steps` - кількість кроків на одне обертання,

`pin1, pin2, pin3, pin4` - виходи контролера, що використовуються для керування двигуном (у залежності від типа двигуна використовують два або чотири вихода).

Функція встановлення швидкості обертання двигуна `myStepper`:

`myStepper.setSpeed(speed);`

де `speed` - швидкість обертання у кількості обертів за хвилину.

Функція встановлення кількості кроків, на яке буде обертатися двигун myStepper (позначка – означає, що двигун буде обертатися в іншу сторону):

```
myStepper.step(steps);  
де steps - кількість кроків.
```

2. Практична частина

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату та контролер Arduino, у разі дистанційного навчання, створити відповідну програму за допомогою емулятора UnoArduSim (Додаток 2).

Приклади розв'язання задач з теми заняття

Розглянемо, як здійснюється керування двигуном постійного струму та кроковим двигуном контролером Arduino за допомогою драйвера L298N та контролера ArduinoMega.

Приклад програми керування двигуном постійного струму.

H-міст підключений до виходів 8 та 9, що мають можливість видавати сигнали ШІМ.

Схема підключення наведена на рис. 1.

Програма здійснює обертання в одну сторону тривалістю 2 с, потім зупинка на 1 с, після чого здійснюється обертання в іншу сторону тривалістю 2 с та зупинкою на 1 с.

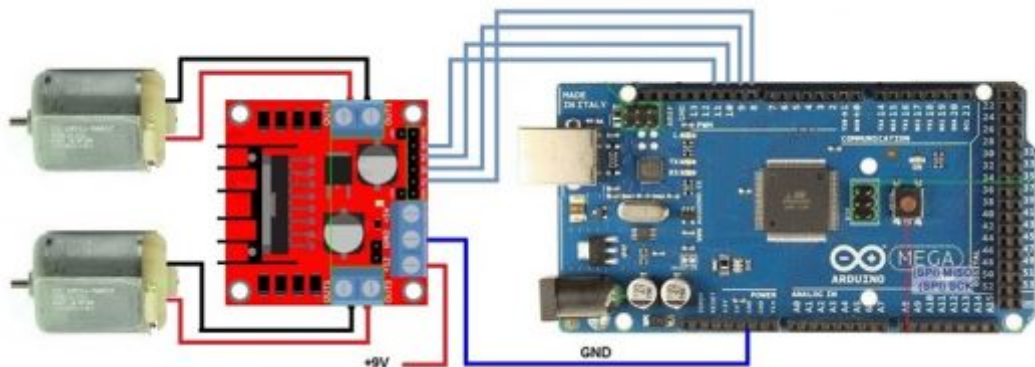


Рис. 1. Схема підключення

Далі наведений приклад програми керування двигуном.

```
const int IN1= 8;  
const int IN2 = 9;    //підключення входів модуля до відповідних контактів  
byte speed = 120;    //швидкість обертання від 0 до 255  
void setup() {  
  pinMode(IN1, OUTPUT);  
  pinMode(IN2, OUTPUT); }  
void loop() {  
  analogWrite(IN1, speed);    //обертання двигуна в одну сторону  
  analogWrite(IN2, 0);  
  delay(2000);                //затримка 2 с  
  analogWrite(IN1, 0);        //зупинка  
  analogWrite(IN2, 0);  
  delay(1000);                //затримка 1 с  
  analogWrite(IN1, 0);        //обертання в іншу сторону, двигун А  
  analogWrite(IN2, speed);  
  delay(2000);                //затримка 2 с  
  analogWrite(IN1, 0);        //зупинка  
  analogWrite(IN2, 0);  
  delay(1000); }             //затримка 1 с
```

Розглянемо, як здійснюється керування кроковим двигуном контролером Arduino. Для спрощення програмування використовується бібліотека Stepper Library в Arduino IDE. Використовується двигун, що має 200 кроків на оборот і може працювати з частотою обертання 60 об / хв., а також контролер ArduinoMega (рис. 2).

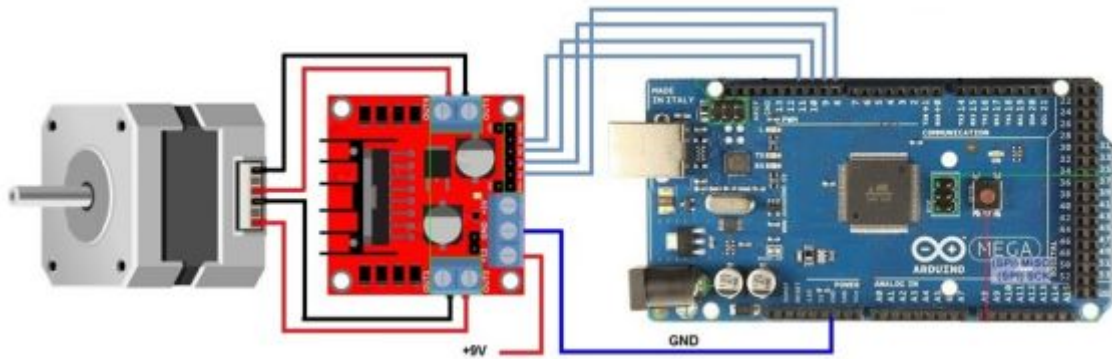


Рис.2. Підключення крокового двигуна до контролера

Контакти модуля L298N IN1, IN2, IN3 и IN4 підключені до відповідних цифрових контактів D8, D9, D10 и D11 у режимі виходів на Arduino.

Далі наведений приклад програми, яка здійснює одне обертання в одну сторону та одне в іншу. Між циклами обертання затримка 0,5 с.

```
#include <Stepper.h>
const int stepsPerRevolution = 200; // кількість кроків на одне обертання
// підключення до виходів 8 - 11:
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);
void setup() {
  myStepper.setSpeed(60); // встановлення швидкості 60 об/хв
}
void loop() {
  myStepper.step(stepsPerRevolution); // обертання в одну сторону
  delay(500); // затримка 0,5 с
  myStepper.step(-stepsPerRevolution); // обертання в іншу сторону
  delay(500); // затримка 0,5 с
}
```

Розглянемо, як здійснюється керування сервоприводом контролером Arduino (рис. 3).

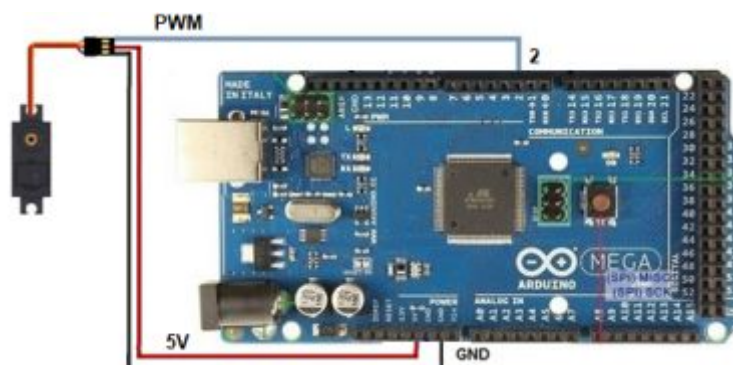


Рис.3. Підключення сервопривода до контролера

Для спрощення програмування використовується бібліотека Servo або VarSpeedServo, яка має більше можливостей, а саме:

- підтримує до 8 сервоприводів

- дозволяє одночасний, асинхронний рух всіх сервоприводів;
- може бути встановлена швидкість руху;
- функція write () ініціює переміщення і може додатково дочекатися завершення переміщення.

Далі наведений приклад програми, яка здійснює послідовне переміщення на 0, 90, 180, 90 градусів.

Приклад програми

```
#include <VarSpeedServo.h>
VarSpeedServo myservo; // складає servo object
void setup() {
  myservo.attach(2); // підключає серво до pin 2 контролера
  myservo.write(0, 10, true); // переміщення на вихідне положення
                               //(0 градусів) зі швидкістю 10, поки не завершиться рух
}
void loop() {
  myservo.write(0, 50, true); // переміщення на 0 градусів,
  delay(500); //затримка 0,5 с
  myservo.write(90, 50, true); // переміщення на 90 градусів,
  delay(500); //затримка 0,5 с
  myservo.write(180, 50, true); // переміщення на 180 градусів,
  delay(500); //затримка 0,5 с
  myservo.write(90, 50, true); // переміщення на 90 градусів,
  delay(500); //затримка 0,5 с
}
```

Завдання 1

Скласти програму для керування швидкості двигуна постійного струму за допомогою потенціометра для контролера Arduino.

Перевірити роботу програми на стенді або, у разі дистанційного навчання, створити та перевірити відповідну програму за допомогою емулятора UnoArduSim.

Завдання 2

Скласти програму для керування послідовністю переміщень сервопривода за допомогою контролера Arduino з встановленням таких кутів повороту:

0°, 40°, 60°, 100°, 120°, 100°, 60°, 40°, 0°,

з затримкою між окремими переміщеннями 4 с.

Перевірити роботу програми на стенді або, у разі дистанційного навчання, створити та перевірити відповідну програму за допомогою емулятора UnoArduSim (використовується бібліотека Servo.

Контрольні питання

1. Визначити, як здійснюється керування двигуном постійного струму?
2. Описати, яка функція використовується в Arduino для управління виходом ШІМ?
3. Назвати, яка бібліотека використовується для керування кроковим двигуном?
4. Назвати, яка бібліотека використовується для керування сервоприводом?
5. Описати, як здійснюється регулювання швидкості сервоприводів?

Лабораторне заняття 8. Дослідження принципів складання програм обробки даних для мікроконтролерів

Мета заняття: Дослідження принципів складання програм обробки даних для мікроконтролерів.

Для проведення дослідження треба:

- знати принципи складання програм для мікроконтролерів;
- вміти використовувати мікроконтролери для керування та обробки даних;
- навчитися складати прості програми мікроконтролерів з використанням мови С.

1. Теоретическая частина

Розглянемо приклади програмування для контролерів Arduino, що використовують мову С [1].

Контролери Arduino часто використовують для керування різними робототехнічними пристроями, тому мають вмонтовані функції обробки даних.

Для спрощення програмування для таких пристроїв створені бібліотеки з відповідними функціями.

Функція зчитування значення сигналу з аналогового входу `analogRead()`, яка видає значення від 0 до 1023.

Функція має такий синтаксис:

```
value = analogRead(pin);
```

де `value` – змінна, куди записується результат опитування;

`pin` – номер аналогового входу.

Для вимірювання тривалості імпульсів, наприклад, у ультразвукових датчиків вимірювання відстані (рис. 1), використовується функція `pulseIn()`.

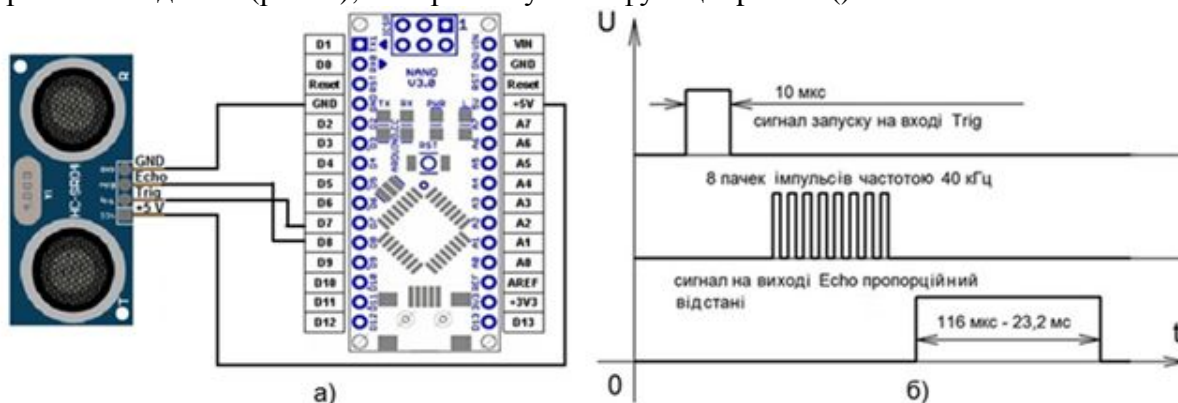


Рис. 1. Схема підключення ультразвукового датчика HC-SR04 до контролера Arduino Nano (а) та діаграми роботи (б)

Ця функція зчитує імпульс (HIGH чи LOW) на визначеному вході (`pin`). Наприклад, якщо значення є високим (HIGH), `pulseIn()` чекає, поки стан `pin` зміниться на HIGH, починає підрахунок часу у мікросекундах, потім чекає, поки стан входу `pin` не зміниться на LOW, і зупиняє підрахунок.

```
pulseIn(pin, value)
```

```
pulseIn(pin, value, timeout)
```

де `pin` - номер піна для зчитування імпульсу (*int*)

`value` - тип імпульсу для зчитування HIGH чи LOW (*int*)

(опціонально)

`timeout` - час очікування у мікросекундах, стандартне значення 1 секунда

(*unsigned long*)

Повертає тривалість імпульсу (у мікросекундах) або 0, якщо не було повного імпульсу за час очікування (*unsigned long*)

Розглянемо алгоритм вимірювання відстані за допомогою ультразвукового датчика HC-SR04.

Датчик має 4 вивода:

VCC: “+” живлення;

TRIG (T): вивід вхідного сигналу;

ECHO (R): вивід вихідного сигналу;

GND: “-” живлення (земля).

Тривалість вихідного сигналу залежить від відстані об'єкту до датчика.

Принцип роботи датчика можна умовно поділити на 4 етапи.

1. Подаємо імпульс тривалістю 10 мкс, на вивід Trig.
2. Датчик перетворює вхідний імпульс у 8 імпульсів частотою 40 КГц та відправляє вперед через передавач T.
3. Дійшовши до перешкоди, послані імпульси відбиваються і приймаються приймачем R. Отримуємо вихідний сигнал на виводі Echo.
4. Визначаємо тривалість часу до отримання відбитого сигналу (ширина імпульсу t_i).
5. Безпосередньо в контролері переводимо отриманий сигнал довжини імпульсу t_i (мкс) в відстань S (см) за формулою:

$$S = t_i / 58.$$

2. Практична частина

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату, ультразвуковий датчик HC-SR04 та контролер Arduino, у разі дистанційного навчання, створити відповідну програму за допомогою емулятора UnoArduSim (Додаток 2).

Приклади розв'язання задач з теми заняття

Приклад програми вимірювання відстані за допомогою ультразвукового датчика HC-SR04 та передачі отриманого значення по послідовному каналу. При дистанції менше 30 сантиметрів включається світлодіод, що підключений до піну 13.

```
sketch_echo
#define Trig 9
#define Echo 8
#define ledPin 13

void setup() {
  pinMode(Trig, OUTPUT); // вихід
  pinMode(Echo, INPUT); // вхід
  pinMode(ledPin, OUTPUT); //світлодіод
  Serial.begin(9600); } // задаємо швидкість послідовного каналу
  unsigned int impulseTime=0;
  //визначаємо змінну для часу до отримання відбитого сигналу
  unsigned int distance_sm=0; //визначаємо змінну для дистанції
  void loop() {
    digitalWrite(Trig, HIGH); // Подаємо імпульс на вхід trig
    delayMicroseconds(10); // що дорівнює 10 мікросекунд
    digitalWrite(Trig, LOW); // відключаємо
    // визначаємо тривалість часу отримання відбитого сигналу
    impulseTime=pulseIn(Echo, HIGH);
    distance_sm=impulseTime/58; // Перераховуємо в сантиметри
    Serial.println(distance_sm); // Виводимо на послідовний канал
```

```
if (distance_sm<30) // Якщо відстань менше ніж 30 сантиметрів
{
digitalWrite(ledPin, HIGH); // включаємо світлодіод
}
else {
digitalWrite(ledPin, LOW); // у протилежному випадку виключаємо
}
delay(100); }
```

Завдання 1

Скласти програму вимірювання відстані за допомогою ультразвукового датчика HC-SR04 з таким підключенням: Trig - пін 2, Echo - пін 3.

Ультразвуковий датчик керує переміщенням візка.

Візок їде прямо з максимальною швидкістю, якщо відстань до перешкоди більше 40 см.

Візок їде прямо з швидкістю 0,5 від максимальної, якщо відстань до перешкоди більше 20 см.

Якщо відстань до перешкоди менше ніж 20 см, візок зупиняється. При зникненні перешкоди рух продовжується.

Двигун підключений до виходу 5 Arduino.

Перевірити роботу програми на стенді або, у разі дистанційного навчання, створити та перевірити відповідну програму за допомогою емулятора UnoArduSim.

Контрольні питання

1. Назвати, яку функцію має контролер Arduino для перетворення масштабу?
2. Описати, як здійснюється формування вихідного імпульсу у контролера Arduino?
3. Описати, як здійснюється вимірювання відстані за допомогою ультразвукового датчика HC-SR04?
4. Описати, як здійснюється вимірювання тривалості імпульсів у контролерів Arduino?
5. Описати, як здійснюється перерахування дистанції отриманої за допомогою ультразвукового датчика у сантиметри?

Лабораторне заняття 9. Дослідження принципів складання програм у вигляді функціональної схеми

Мета заняття: Дослідження принципів складання програм у вигляді функціональної схеми.

Для проведення дослідження треба:

- знати принципи складання програм у вигляді функціональної схеми;
- вміти використовувати програмовані логічні контролери для керування підйомником;
- навчитися складати програми у вигляді функціональної схеми на прикладі алгоритму керування підйомником.

1. Теоретическая частина

Для вивчення принципів складання програм у вигляді функціональної схеми з метою керування простими механізмами розглянемо як можна здійснити керування підйомником (рис. 1) з функціями захисту за допомогою програмованого логічного контролера LOGO! (рис. 2).

Підйомник здійснює підйом та опускання вантажу за допомогою двигуна постійного струму, напрямом обертання якого здійснюється шляхом переключення полярності напруги живлення. При досягненні верхнього та нижнього положення спрацьовують кінцеві вимикачі, які здійснюють захист від подальшого переміщення. Керування здійснюється у ручному режимі за допомогою кнопок "Вгору" та "Вниз". Переміщення здійснюється, поки натиснута відповідна кнопка. Схема керування повинна здійснити захист від одночасного натиснення обох кнопок.

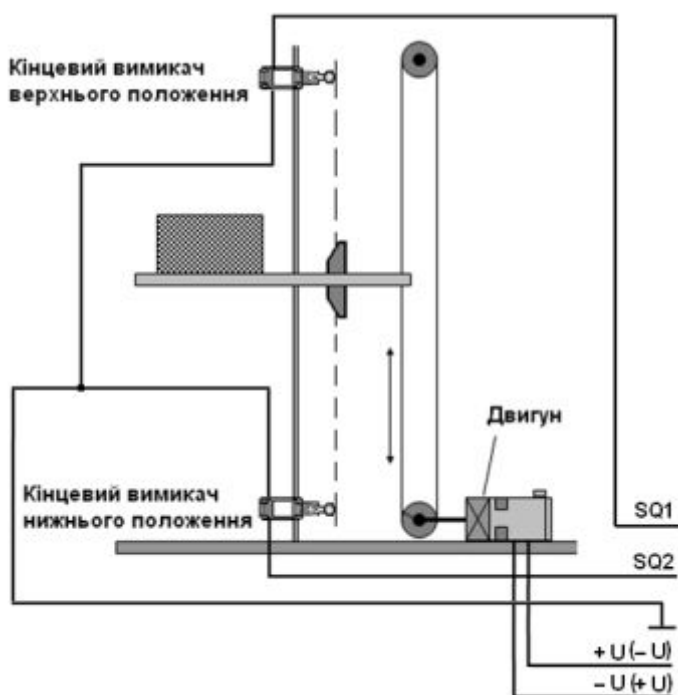


Рис.1. Підйомник

Керування здійснюється за допомогою програмованого логічного контролера LOGO! з релейними виходами, які мають навантажувальну здатність 10 А. Контролер LOGO! можна віднести до найбільш простих пристроїв керування, який має логічні функції, функції пам'яті (тригер), функції часу (таймери) та функції ліку (лічильники). Ці функції дозволяють здійснити прості функції логічного керування.

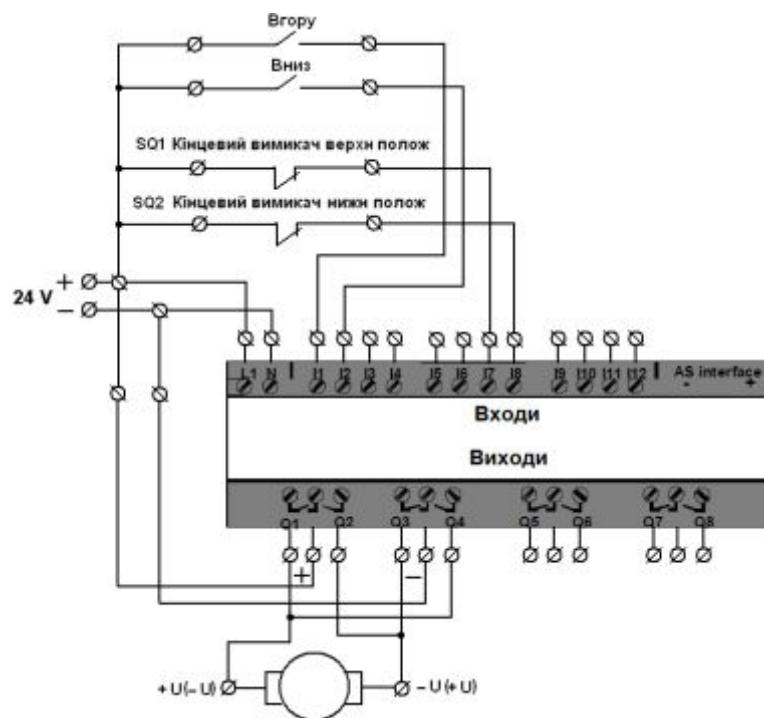


Рис. 2. Схема підключення двигуна підйомника та кнопок керування до контролера LOGO!

Мова програмування LOGO!Soft Comfort [1] дозволяє створити програму та перевірити її роботу на програмному емуляторі.

2. Практична частина

Опис лабораторної установки

Для перевірки роботи треба створити відповідну програму за допомогою мови LOGO!Soft Comfort, яка використовує логічні функції (Додаток 1).

Приклади розв'язання задач з теми заняття

Приклади програм керування підйомником з використання LOGO!Soft Comfort наведені у Додатку 1.

Завданн 1

Скласти блок-схему алгоритму та програму керування підйомником за допомогою мови LOGO!Soft Comfort.

Підйомник переміщається у двох напрямках: догори та униз. Для прямування нагору необхідно короткочасно натиснути кнопку "нагору". Для прямування униз необхідно короткочасно натиснути кнопку "униз".

Прямування нагору або униз припиняється при натисненні кнопки "стій" або при досягненні кінцевих вимикачів "верх" або "низ".

Для здійснення вмикання та вимикання двигуна використати функцію RS-тригера.

Передбачити захист від одночасного натискання кнопок догори та униз.

Перевірити роботу програму на емуляторі LOGO!Soft Comfort.

Контрольні питання

1. Описати, як здійснюється зміна напрямку обертання двигунів постійного струму?
2. Описати, як здійснюється захист від одночасного натискання кнопок?
3. Назвати, яку функцію здійснює кінцевий вимикач верхнього положення?
4. Назвати, яку функцію здійснює кінцевий вимикач нижнього положення?
5. Описати, як працює кінцевий вимикач?

Лабораторне заняття 10. Дослідження принципів складання програм циклового керування

Мета заняття: Дослідження принципів складання програм циклового керування.

Для проведення дослідження треба:

- знати принципи використання команд пам'яті;
- вміти використовувати логічні функції;
- навчитися складати програми послідовного виконання дій.

1. Теоретическая частина

Створення програм за допомогою мови LOGO!Soft Comfort розглянуті у Додатку 1 та в [1].

2. Практична частина

Опис лабораторної установки

Для перевірки роботи треба створити відповідну програму за допомогою мови LOGO!Soft Comfort, яка використовує логічні функції (Додаток 1).

Приклади розв'язання задач з теми заняття

Програма послідовного переміщення

Програма, що здійснює послідовне переміщення трьох приводів, "Привод_1", "Привод_2" та "Привод_3", має вигляд? наведений на рис. 1.

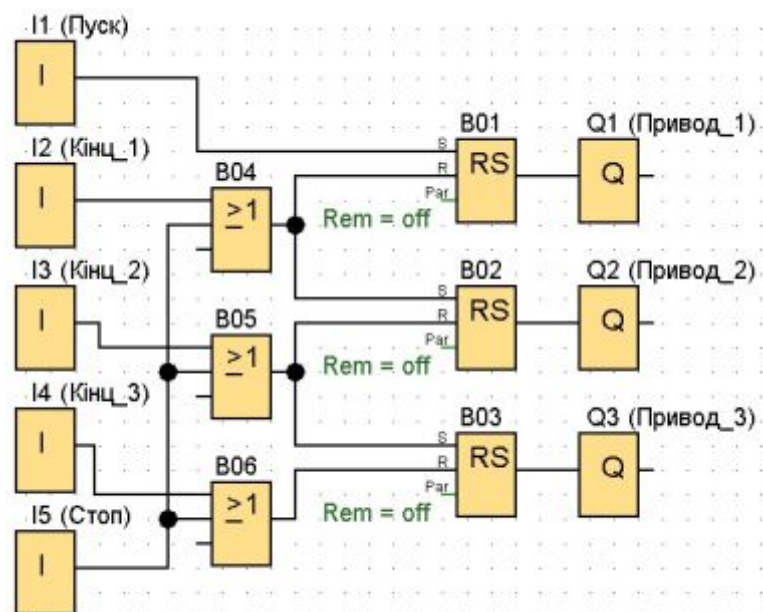


Рис. 1. Приклад програми послідовного переміщення

Переміщення приводу 1 запускається кнопкою "Пуск".

Кінцевий вимикач першого переміщення зупиняє привод 1 та запускає привод 2.

Аналогічно здійснюється переміщення приводу 2 та приводу 3.

Завдання 1

Скласти блок-схему алгоритму та програму циклового керування, що складається з послідовного переміщення з поверненням у вихідне положення.

Маніпулятор здійснює такі пересування: ВВЕРХ, ВНИЗ, ВПРАВО, ВЛІВО, ВПЕРЕД, НАЗАД.

Кожному пересуванню відповідає один вихід на виконавчий пристрій переміщення та один вихід для опитування кінцевого вимикача.

Треба скласти програму пересування, використовуючи функції пам'яті відповідно такій послідовності переміщень:

ВВЕРХ, ВПЕРЕД, ВПРАВО, ВНИЗ, ВВЕРХ, ВЛІВО, НАЗАД, ВНИЗ.

Перевірити роботу програму на емуляторі LOGO!Soft Comfort.

Контрольні питання

1. Описати, як здійснюється послідовне переміщення за допомогою функції пам'яті?
2. Назвати, у яке положення переміщується робочий орган в кінці циклу?
3. Описати, як здійснюється послідовне включення та виключення двигуна?
4. Назвати, який елемент включає двигун?
5. Назвати, який елемент виключає двигун?

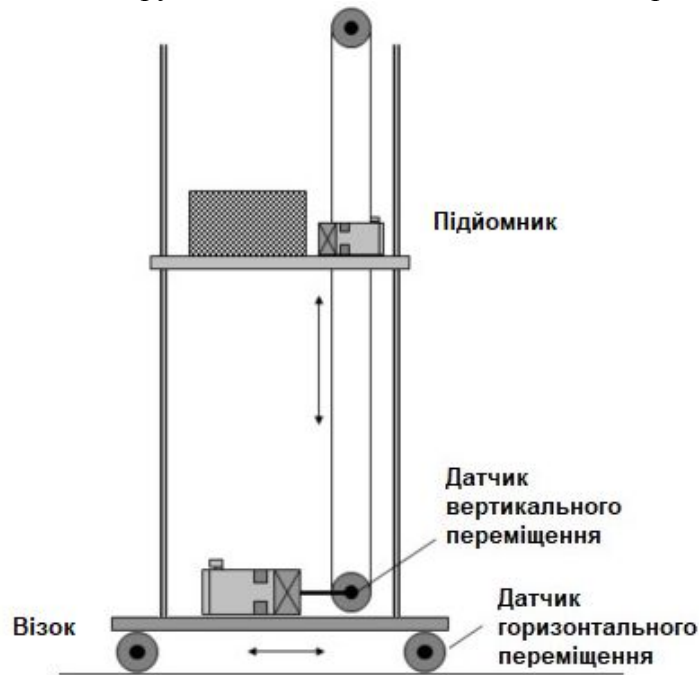
Лабораторне заняття 11. Дослідження принципів складання програм позиційного керування

Мета заняття: Дослідження принципів складання програм позиційного керування. Для проведення дослідження треба:

- знати принципи використання функцій підрахунку;
- вміти використовувати датчики переміщення;
- навчитися складати програми позиційного керування.

1. Теоретическая частина

Розглянемо позиційне керування візка та підйомника штабелера (рис. 1).



Для позиційного керування визначення шляху переміщення можна здійснити за допомогою імпульсних датчиків швидкості (переміщення) FC-03 (рис. 1).

При обертанні диска отвори перетинає щілину датчика та видають послідовність імпульсів. Підраховуючи кількість імпульсів та знаючи кількість імпульсів на одне обертання колеса та його діаметр можна визначити відстань, пройдену візком. Аналогічно здійснюється керування підйомником.



2. Практична частина

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату, датчик швидкості FC-03, встановлений на двигун постійного струму, та контролер Arduino, у разі дистанційного навчання, створити відповідну програму за допомогою емулятора UnoArduSim (Додаток 2).

Приклади розв'язання задач з теми заняття
Програма переміщення до вказаної позиції.
Переміщення до позиції 5 наведено на рис. 1.

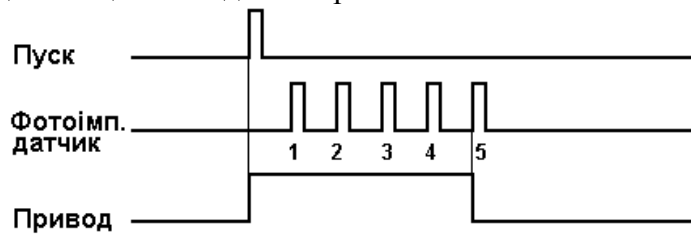


Рис. 1. Переміщення до позиції 5

Сигнал "Пуск" запускає привод.

Під час переміщення фотоімпульсний датчик видає імпульси.

На 5-ому імпульсі спрацьовує лічильник та зупиняє привод (рис. 3).

Сигнал "Стоп" зупиняє привод та перезавантажує лічильник.

```
Програма позиціонування для переміщення вправо, вліво
// підключення драйвера двигунів до цифрових пінів Arduino
#define in1 5 // двигун 1 вправо
#define in2 6 // двигун 1 вліво
#define in3 9 // двигун 2 уверх
#define in4 10 // двигун 2 униз
#define dsR 2 // Встановлення контакта 2 для переривання
volatile unsigned int pulses;
//функція підрахунку імпульсів
void counter()      {
  pulses++; }

void setup() {
  // усі піни для керування двигунами як outputs
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  Serial.begin(9600);
  pinMode(dsR, INPUT);
  pulses = 0;
  // дозвіл переривання від датчика переміщення
  attachInterrupt(digitalPinToInterrupt(dsR), counter, FALLING); }

void loop()      {
  // запуск двигуна 1
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  if (pulses >= 100) //якщо кількість імпульсів перевищує 100, зупинка
  {
    // заборона переривання
    detachInterrupt(digitalPinToInterrupt(dsR));
    Serial.println(pulses);
    //Двигуни Стоп
    digitalWrite(in1, LOW);
```

```
digitalWrite(in2, LOW);  
delay(2000);  
pulses = 0;  
// дозвіл переривання  
attachInterrupt(digitalPinToInterrupt(dsR), counter, FALLING); } }
```

Завдання 1

Скласти блок-схему алгоритму та програму переміщення до визначеної позиції з використанням датчика на основі контролера Arduino.

Маніпулятор з позиційним керуванням здійснює такі переміщення: УВЕРХ, УНИЗ, ВПРАВО, ВЛІВО.

Вихідне положення визначається кінцевими вимикачами.

Кожному пересуванню відповідає один вихід на виконавчий пристрій переміщення та один вихід для опитування датчика пересування.

Позиціонування здійснюється під час руху УВЕРХ, УНИЗ та ВПРАВО, ВЛІВО.

Треба скласти програму пересування відповідно такій послідовності переміщень:

Переміщення ВПРАВО до позиції 40.

Переміщення УВЕРХ до позиції 20.

Перевірити роботу програми на стенді або, у разі дистанційного навчання, створити та перевірити відповідну програму за допомогою емулятора UnoArduSim.

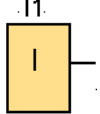
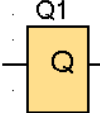
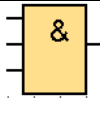
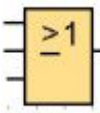
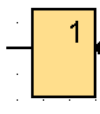
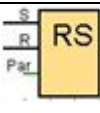
Контрольні питання

1. Описати, як здійснюється позиційне переміщення?
2. Назвати, яке положення є вихідним?
3. Описати, яку функцію використовують для підрахунку дій?
4. Назвати, який елемент використовується як датчик положення?
5. Назвати, який елемент визначає вихідне положення?

Література

1. Михайлов Є. П. Навчальний посібник з дисциплін «Електронні, мікропроцесорні та обчислювальні пристрої ГВС, ПТМ та ЛС» для студентів за фахом 131 – Прикладна механіка – спеціалізації – Мехатроніка та промислові роботи, Інженерія логістичних систем, 133 – Галузеве машинобудування – спеціалізація – Підйомно-транспортні, будівельні, дорожні машини і обладнання / Укладач: Михайлов Є. П. Одеса: ОНПУ. – 2018. 167 с. Рег. ном. НПО9356 19.03.18 №5682-РС-2018 (електронна версія)
2. Блум Джереми. Изучаем Arduino: инструменты и методы технического волшебства: Пер. с англ. - СПб.: БХВ-Петербург, 2015. - 336 с.: ил. (електронна версія)
3. Управляющие системы и автоматика / Дитмарт Шмид, Альбрехт Бауман, Ханс Кауфман, Бернгард Зиппель. -М.: Техносфера, 2007. – 584 с.
4. Руководство LOGO!. Справочник по аппарату Siemens AG, Industry Sector A5E01248543-03 04/2009 (електронна версія)
5. Программное обеспечение LOGO!Soft Comfort V6.0 Siemens AG, Industry Sector (електронна версія)

Елементи та функції ПЛК LOGO!

	Вхід Видає результат опитування відповідних входів 0 або 1.
	Вихід Сигнал 0 або 1 видає на відповідний вихід.
	Функція логічного І На виході 1, коли на усіх входах 1. У протилежному випадку на виході 0.
	Функція логічного АБО На виході 1, коли хоча б на одному вході 1. У протилежному випадку, коли на усіх входах 0, на виході 0.
	Функція інверсії На виході 0, коли на вході 1. На виході 1, коли на вході 0.
	Функція Реле з блокуванням (RS-тригер) Вхід S. Якщо на воді S сигнал 1, вихід встановлюється в 1. Вхід R. Якщо на воді R сигнал 1, вихід скидається в 0. Пріоритет входу R. Par. Параметри (у нашому випадку не використовуються)

Порядок складання програми для ПЛК LOGO! за допомогою програми LOGOComfort наведений в [1].

Приклади програм

Наведена програма керування підйомником, що здійснює підйом та опускання вантажу за допомогою двигуна постійного струму та виконує такі функції.

Напрямок обертання двигуна здійснюється шляхом переключення полярності напруги живлення.

При досягненні верхнього та нижнього положення спрацьовують нормально замкнуті кінцеві вимикачі, які здійснюють захист від подальшого переміщення.

Керування здійснюється у ручному режимі за допомогою кнопок "Вгору" та "Вниз". Переміщення здійснюється, поки натиснута відповідна кнопка. Схема керування повинна здійснити захист від одночасного натиснення обох кнопок.

Напругу для живлення двигуна підвести до таких клем:

+U до групи клем виходів Q1 та Q2,

- U до групи клем виходів Q3 та Q4.

При натисканні кнопки "Вгору" (вхід I1) підключаються виходи Q1 та Q3 та здійснюється підйом, а при натисканні кнопки "Вниз" (вхід I2) підключаються виходи Q2 та Q4 та здійснюється опускання.

Для блокування від одночасного натиснення обох кнопок можна використати підхід, наведений на рис. 3.

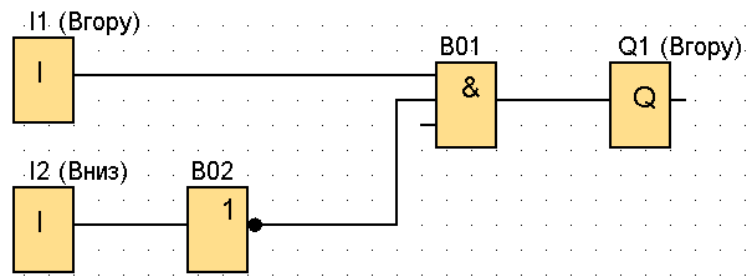


Рис. 3. Блокування від одночасного натиснення кнопок

Вхід I1 поступає на перший вхід функції I (V01), а вхід I2 поступає спочатку на функцію інверсії (V02), а потім на другий вхід функції I (V01). Коли на вході I2 немає сигналу (на вході сигнал 0), то після інверсії (сигнал 1) він дає сигнал дозволу для функції I. Якщо на вході I2 є сигнал (1), то після інверсії (0) він блокує функцію I.

Третій вхід функції I використовується для блокування переміщення при спрацюванні кінцевого вимикача.

Кінцеві вимикачі треба підключити до входів I3 (кінцевий вимикач верхнього положення) та I4 (кінцевий вимикач нижнього положення).

Програма керування підйомником у вигляді функціональної, що здійснює вказане керування підйомником, має вигляд, наведений на рис. 4.

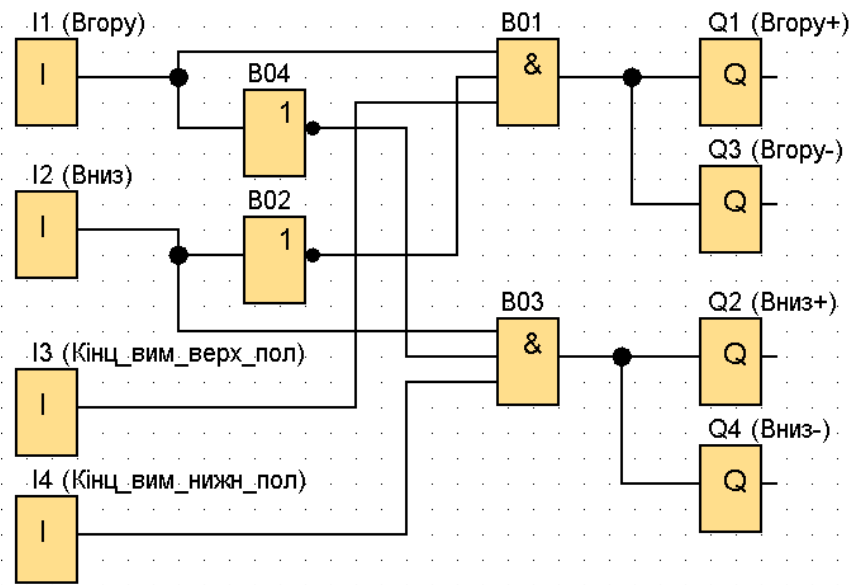


Рис. 4. Програма керування підйомником

Функція вмикання (кнопка Пуск) та вимкання (кнопка Стоп) здійснюється за допомогою функції RS-тригера (рис. 5).

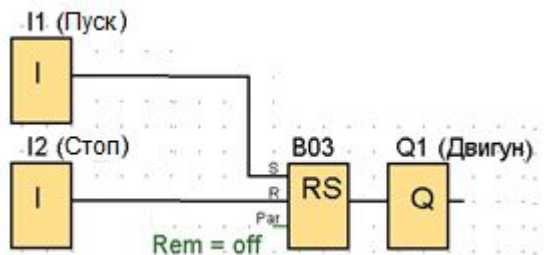


Рис. 5. Функція RS-тригер

Створення та перевірка програм для контролера Arduino за допомогою емулятора UnoArduSim

Додаток UnoArduSim є безкоштовним емулятором (симулятором) контролера Arduino, який дає можливість здійснити виконання програми в реальному часі без наявності самої плати Arduino.

При цьому є можливість перегляду ходу виконання програми.

UnoArduSim призначений для полегшення налагодження Arduino програм і містить набір віртуальних пристроїв вводу / виводу ('I / O' Пристроїв), які можна налаштовувати і підключати до віртуального Arduino.

UnoArduSim не потребує установки. Його треба завантажити за посиланням <https://www.sites.google.com/site/unoardusim/home>, після чого запустити файл UnoArduSim.exe і працювати з ним.

Симулятор працює в операційній системі Windows починаючи з Windows XP.

Після завантаження UnoArduSim для спрощення запуску можна створити ярлик на робочому столі.



Після запуску програми з'являється вікно (рис. 1) з вихідним прикладом програми:

```

/* This is a default program--
Use File->Load Prog to load a different program
*/
int count;
void setup()
{
count=0;
}
void loop()
{
count=count+1;
delay(100);
}

```



Рис. 1. Вихідне вікно

Подвійним кліком зони, де наведена програма, запускається вікно редагування Arduino програм (Edit / View Program), яке наведено на рис.2.

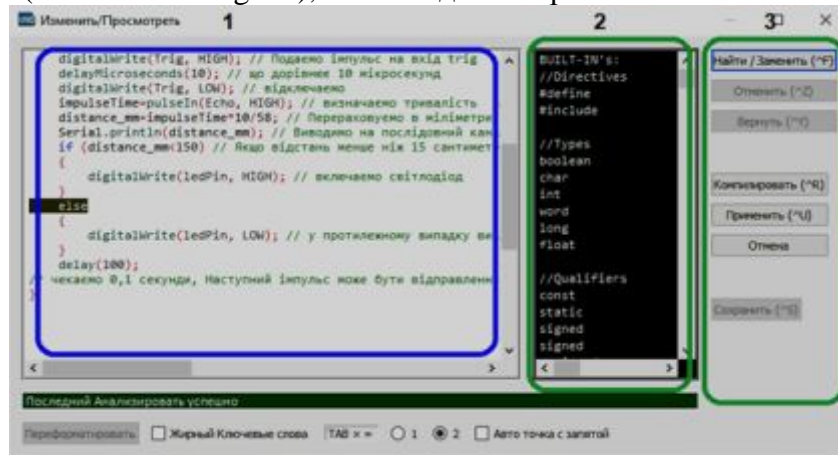


Рис. 2. Вікно редагування програми

Зона 1 призначена для внесення та редагування Arduino програм (скетчів). Вона містить невеликий код для емуляції скетчу, цей код можна видалити, потім він додається автоматично. Тут можна вставити свій код або скопійований з програми Arduino IDE.

Права кнопка мишки тут не працює, так що копіювати і вставляти доводиться через команди Control C і Control V.

Зона 2 містить каталог мови Ардуіно, в якому наведені команди та функції, який працює наступним чином – треба встановити курсор в потрібному місці в зоні 1 і двічі клікнути на потрібну функцію в зоні 2.

Зона 3 включає кнопки:

Найти / Заменить (^ H) - для виклику пошуку у програмі,

Отменить (^ Z) Вернуть (^ Y) - крок назад чи крок вперед,

Компилировать (^ R) - скомпіювати (перевірити на наявність помилок),

Применить (^ U) - прийняти,

Отмена - скасування,

Сохранить (^ S) - зберегти.

На рис. 3 наведено вікно змінних.

Тут під час емуляції можна спостерігати стан і значення всіх змінних, які містяться в скетчі.

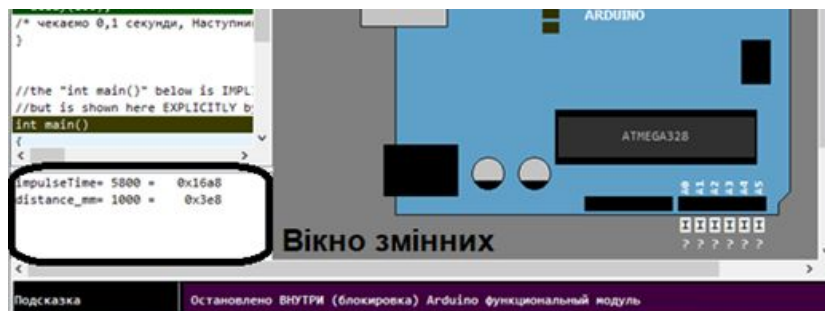


Рис. 3. Вікно змінних

Скорочену та повну інформацію про UnoArduSim можна знайти у закладці «Помощь» - «Быстрый помощь», «Полный помощь».

На рис. 4 наведений приклад виконання програми, що здійснює миготіння контакту 13 та підрахунок кількості миготінь.

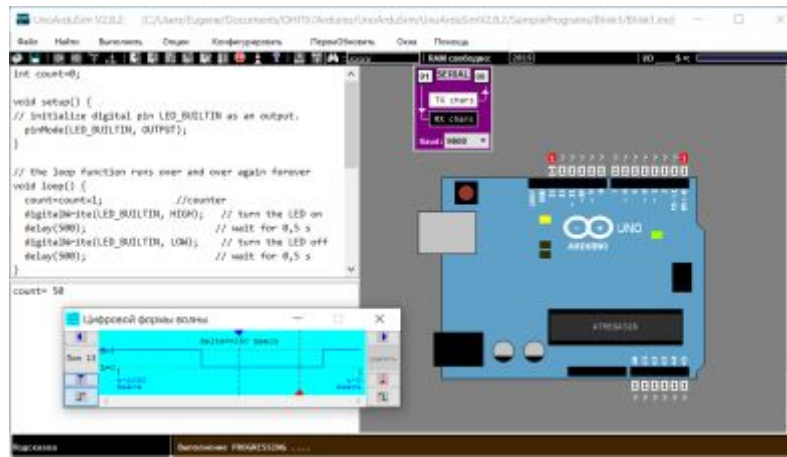


Рис. 4. Відображення виконання програми на лабораторній панелі

На лабораторній панелі можна спостерігати стан контактів віртуального Arduino UNO.

Знак питання означає що контакт не використовується, нуль на синьому фоні це сигнал низького рівня LOW, одиниця на червоному фоні це сигнал високого рівня HIGH, стрілка вгору на рожевому фоні, це сигнал ШІМ. На платі є функціонуючі світлодіоди: ON, RX, TX і pin 13.

Також, якщо клікнути лівою кнопкою по активному контакту, то запуститься вікно цифрового осцилографа (видає значення 0 та 1).

Права кнопка запускає аналоговий осцилограф, що дозволяє проглянути сигнали на аналогових входах.

Синя і червона мітки параметра delta виставляються шляхом перетягування їх мишкою, а параметр t (час) налаштовується коліщатком мишки.

Змінна count у вікні змінних показує кількість миготінь.

Додаток UnoArduSim зберігається у папці UnoArduSimV2.8.2 (остання версія).

У папці translations знаходяться переклади файлів допомоги UnoArduSim_FullHelp та UnoArduSim_QuickHelp на різні мови. (російською мовою UnoArduSim_QuickHelp_ru.pdf та UnoArduSim_FullHelp_ru.pdf).

У папці SamplePrograms знаходяться приклади програм. Цю папку можна використовувати для зберігання своїх програм у папках з відповідними назвами.

Для перевірки отриманих даних можна використовувати Монитор 'SERIAL', який дає можливість введення та відображення даних (рис. 5).

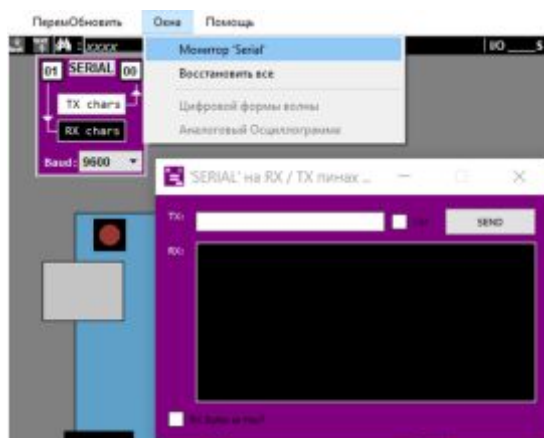


Рис. 6. Монитор 'SERIAL',

Монитор 'SERIAL' (вікно 'SERIAL' на RX / TX пинах...) можна визвати подвійним кліком на інструмент 'SERIAL', або через закладку «Окна» - «Монитор 'SERIAL'».

Справа на додатку UnoArduSim в запущеному стані розташована лабораторна панель з різними інструментами розташованими по периметру, які можна налаштувати і підключати до віртуального Arduino UNO.

Всі ці інструменти можна додавати, вказавши потрібну кількість у вікні I/O Устройства, яке знаходиться у вкладці **Конфигурировать – I/O Устройства** (рис. 6).

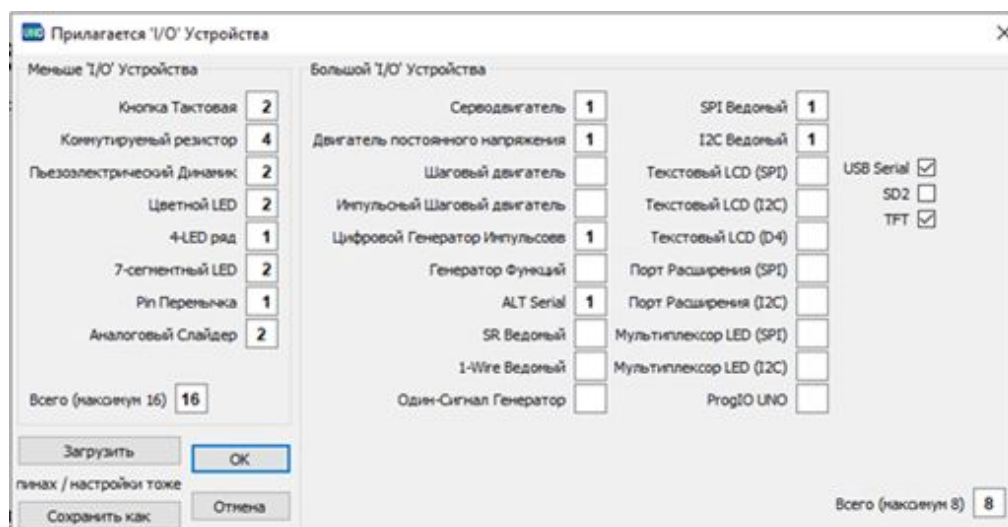


Рис. 6. Вкладка **Конфигурировать – I/O Устройства**

Видалити інструменти можна видаливши ці значення у вікні кількості.

Детальну інформацію про інструменти можна знайти у вказаних закладках «Быстрый помощь», «Полный помощь».

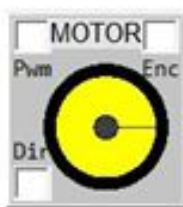
Для збереження налагодження треба натиснути кнопку «Сохранить как» та вказати папку, де знаходиться відповідна програма.

Розглянемо деякі інструменти.

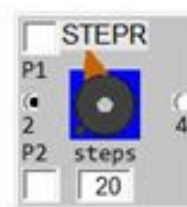
У наявності є три модулі моторів (рис.7), а саме, серводвигуни, двигуни постійного струму та крокові двигуни.



а)



б)



в)

Рис. 7. Серводвигун (а), двигун постійного струму (б), кроковий двигун (в)

Серводвигун SERVO, що наведений на рис. 7, а, управляється ШІМ сигналом, в залежності від шпаруватості ШІМ сигналу вал приймає певний кут від 0 до 180 градусів. Використовується, коли треба здійснити поворот на певний кут.

Двигун постійного струму DC MOTOR наведений на рис. 7. б.

Цей інструмент вводу-виводу здійснює емуляцію електродвигуна постійного струму.

Швидкість обертання встановлюється за допомогою сигналу з широтно-імпульсною модуляцією, що подається на вхід Pwm, напрямком обертання встановлюється на вході Dir.

На вихід Enc поступають сигнали з умонтованого енкодера (імпульсного датчика переміщення), що видає 8 імпульсів на одне обертання.

Кроковий двигун STEPR (Stepper) наведений на рис. 7, в. У вікні steps виставляється кількість кроків на один оборот ротора. Є два варіанти: двохфазний і чотирьохфазний які перемикаються шляхом установки галочки над двійкою або над четвіркою. В даний момент

активний двофазний варіант з підключенням до контактів P1 і P2, якщо вибрати чотирьохфазний то відповідно додадуться ще два контакти P3 і P4. Обертається за рахунок зміни полярності, при кожній зміні полярності відбувається переміщення ротора на один крок.

На рис. 8 наведені інструменти для введення цифрових та аналогових даних.

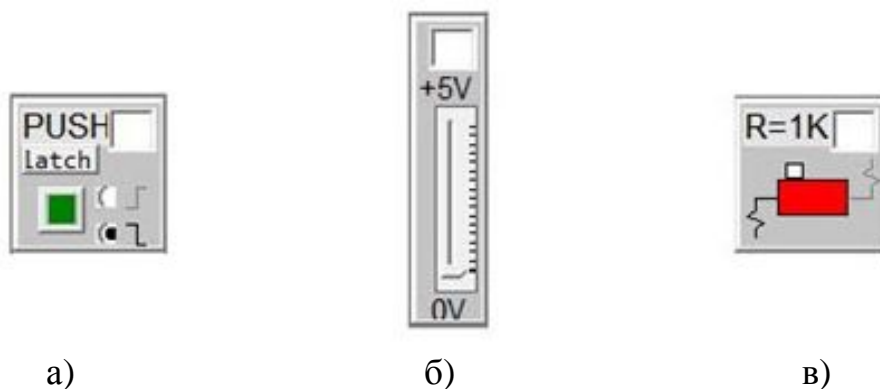


Рис. 8. Кнопка (а), повзунковий потенціометр (б), перемикач (в)

Кнопка PUSH (рис. 8, а). На рисунку кнопка має на виході одиницю, а при натисканні на неї на виході буде нуль. Якщо переставити галочку вище то на виході буде нуль, а при натисканні буде одиниця, тобто навпаки.

Повзунковий потенціометр (рис. 8, б). Управляється шляхом перетягування повзунка мишкою (вгору-вниз). На виході формується аналоговий сигнал від 0 до 5 вольт, в програмному вигляді це значення від 0 до 1023.

Перемикач R = 1K (рис. 8, в) представле собою подтягівуюшій резистор з номіналом 1 Ком, який за замовчуванням підключений до мінуса. Якщо клікнути на нього то він підключиться до плюса.

Інструмент Одна дія ('1SHOT'), наведений на рис. 8.

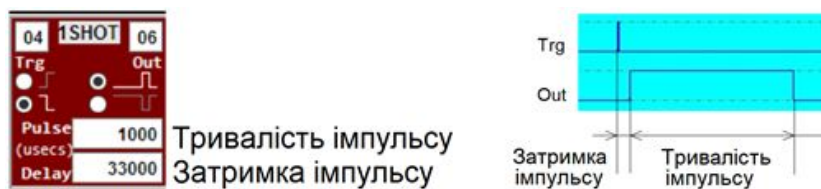


Рис. 8. Інструмент Одна дія ('1SHOT')

Цей інструмент вводу-виводу здійснює емуляцію цифрової одноразової дію, яка може генерувати імпульс обраної полярності та тривалості Pulse на контакті "Out".

Імпульс виникає після заданої затримки Delay від фронту запуску, який надходить на вхідний контакт Trg (тригер).

Параметри Pulse та Delay надаються у мікросекундах.

Протоколи до лабораторних занять

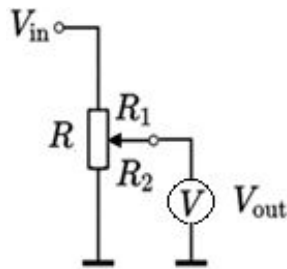
ПРОТОКОЛ
до лабораторного заняття 1
Дослідження принципів обчислювання параметрів пасивних електронних схем

Студент _____ група _____

Мета заняття: Дослідження принципів обчислювання параметрів пасивних електронних схем, що включають резистивні подільники напруги.

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату та відповідні електронні компоненти. Для вимірювання напруги на виході подільника використовується принципова схема, що наведена на рис. 1.

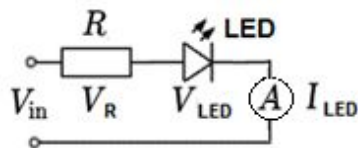


$$R = R_1 + R_2$$

$$V_{out} = \frac{R_2}{R_1 + R_2} \cdot V_{in}$$

Рис. 1. Схема вимірювання напруги на виході подільника

Для вимірювання струму світлодіоду використовується схема, що наведена на рис. 2.



$$R = (U - U_{LED}) / I_{LED} .$$

$$I_{LED} = (U - U_{LED}) / R.$$

Рис. 2. Схема вимірювання струму світлодіоду

Завдання 1

Знайти параметри подільника напруги R_1 та R_2 для заданих значень коефіцієнта передачі V_{out}/V_{in} та загального опору $R_1 + R_2$.

V_{out}/V_{in}	$R_1 + R_2, \text{кОм}$	$R_1, \text{кОм}$	$R_2, \text{кОм}$
0,1	1		
0,5	50		
0,25	270		
0,01	100		

Зібрати схему з визначеними параметрами, перевірити коефіцієнт передачі подільника.

Завдання 2

Знайти V_{out} , для заданих значень V_{in} , R_1 та R_2 .

$V_{in}, \text{В}$	$R_1, \text{кОм}$	$R_2, \text{кОм}$	$V_{out}, \text{В}$
24	6,8	2,2	
12	750	5,1	
5	4,7	0,27	
10	1,2	0,1	

Зібрати схему з визначеними параметрами, перевірити напругу на виході подільника.

Завдання 3

Знайти значення обмежувального опору стандартного світлодіода для напруги живлення U , якщо $U_{LED} = 2 \text{ В}$, $I_{LED} = 20 \text{ мА}$.

$U, \text{В}$	$R, \text{Ом}$
12	
24	
10	
220	

Зібрати схему з визначеними параметрами, перевірити струм світлодіода.

Контрольні питання

1. Визначити, для чого використовують подільники напруги?

2. Описати, як здійснюється регулювання напруги за допомогою подільника?

3. Назвати, для чого використовують світлодіоди?

4. Описати, як треба включати світлодіоди, щоб вони світилися?

5. Назвати, чому треба обмежувати струм світлодіода?

Висновки по роботі

ПРОТОКОЛ
до лабораторного заняття 2
Дослідження принципів обчислювання параметрів активних електронних схем

Студент _____ група _____

Мета заняття: Дослідження принципів обчислювання параметрів активних електронних схем на основі транзисторів.

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату та відповідні електронні компоненти. Для вимірювання коефіцієнта підсилювання використовується принципова схема, що наведена на рис. 1.

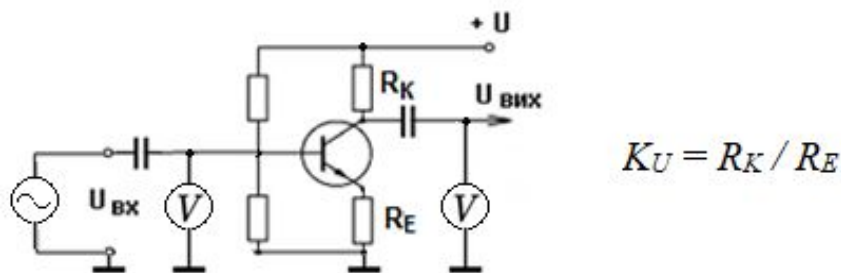


Рис. 1. Схема вимірювання коефіцієнта підсилювання каскаду підсилювання із спільним емітером

Завдання 1

Знайти коефіцієнт підсилювання для схеми із спільним емітером K_U , для заданих значень R_K та R_E .

$R_K, \text{кОм}$	$R_E, \text{кОм}$	K_U
6,8	2,2	
750	5,1	
4,7	0,27	
1,2	0,1	
100	1,0	

Зібрати схему з визначеними параметрами, перевірити роботу підсилювача.

Контрольні питання

1. Описати, як визначити коефіцієнт підсилювання схеми із спільним емітером?

2. Описати, як визначити коефіцієнт підсилювання схеми із спільним колектором?

3. Описати, як визначити коефіцієнт підсилювання змінного струму?

4. Визначити, як здійснити вимірювання напруги на виході підсилювача?

5. Визначити, як здійснити формування вхідного сигналу?

Висновки по роботі

ПРОТОКОЛ
до лабораторного заняття 3
Дослідження принципів обчислювання параметрів аналогових інтегральних мікросхем

Студент _____ група _____

Мета заняття: Дослідження принципів обчислювання параметрів аналогових інтегральних мікросхем.

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату та відповідні електронні компоненти. Для вимірювання коефіцієнта підсилювання використовується принципова схема, що наведена у попередній лабораторній роботі.

Домашнє завдання

Використовуючи матеріали попереднього заняття представити схеми вимірювання коефіцієнта підсилювання інвертуючого та неінвертуючого операційних підсилювачів.

Схеми вимірювання коефіцієнта підсилювання

Завдання 1

Знайти коефіцієнт підсилювання інвертуючого операційного підсилювача, якщо відомі $R1$ та $R2$.

K	$R1, кОм$	$R2, кОм$
	150	1
	10	5
	275	25
	100	0,1

Завдання 2

Знайти коефіцієнт підсилювання неінвертуючого операційного підсилювача, якщо відомі $R1$ та $R2$.

K	$R1, кОм$	$R2, кОм$
	150	1
	10	5
	275	25
	100	0,1

Завдання 3

Знайти коефіцієнт підсилювання диференціального операційного підсилювача, якщо відомі $R1$ та $R2$.

K	$R1, \text{кОм}$	$R2, \text{кОм}$
	150	1
	10	5
	275	25
	100	0,1

Зібрати схему з визначеними параметрами, перевірити роботу підсилювачів.

Контрольні питання

1. Описати, як визначити коефіцієнт підсилювання інвертуючого операційного підсилювача?

2. Описати, як визначити коефіцієнт підсилювання неінвертуючого операційного підсилювача?

3. Описати, як визначити коефіцієнт підсилювання диференціального операційного підсилювача?

4. Визначити, як здійснити вимірювання напруги на виході операційного підсилювача?

5. Визначити, як здійснити формування вхідного сигналу?

Висновки по роботі

ПРОТОКОЛ
до лабораторного заняття 4
Дослідження принципів обчислювання параметрів дискретних інтегральних мікросхем

Студент _____ група _____

Мета заняття: Дослідження принципів обчислювання параметрів дискретних інтегральних мікросхем.

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату та відповідні мікросхеми або, у разі дистанційного навчання, створити відповідну програму за допомогою мови LOGO!Soft Comfort, яка використовує логічні функції.

Завдання 1

Скласти схему керування, яка включає двигун кнопкою "Пуск" та виключає кнопкою "Стоп", з використанням логічних елементів та RS-тригера.

Передбачити блокування від одночасного натискання кнопок "Пуск" та "Стоп" (пріоритет кнопки "Стоп").

Перевірити роботу програми на стенді або, у разі дистанційного навчання, створити та перевірити відповідну програму за допомогою мови LOGO!Soft Comfort.

Навести складену схему керування.

Схема керування з використанням дискретних інтегральних мікросхем

Контрольні питання

1. Визначити, які функції виконує схема логічного І?

2. Описати, які функції виконує схема логічного АБО?

3. Описати, які функції виконує схема інверсії?

4. Описати, які функції виконує схема RS-тригера?

5. Описати, як перевірити роботу програми в LOGO!Soft Comfort ?

Висновки по роботі

ПРОТОКОЛ
до лабораторного заняття 5
Дослідження принципів обчислювання параметрів цифро-аналогових перетворювачів

Студент _____ група _____

Мета заняття: Дослідження принципів обчислювання параметрів цифро-аналогових перетворювачів.

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату та контролер Arduino, у разі дистанційного навчання, створити відповідну програму за допомогою емулятора UnoArduSim.

Завдання 1

Знайти точність перетворення цифрового коду у напругу для ЦАП з вказаними розрядністю та діапазоном вихідних сигналів.

ΔU	$(U_{\max} \square U_{\min}), B$	N
	10	10
	± 5	12
	1	16

Завдання 2

Представити у двійковій формі такі числа

250	- 250	1024	- 1024

Завдання 3

Визначити діапазон представлення даних у двійковій та десятковій формі, які отримує 12-розрядний ЦАП для біполярних вихідних сигналів (представлення даних як цілі числа зі знаком).

Завдання 4

Створити програму для послідовного виведення за допомогою ШІМ сигналу з напругою $U_{\text{вих}} = 0,2 U_{\text{ж}}$, $U_{\text{вих}} = 0,4 U_{\text{ж}}$, $U_{\text{вих}} = 0,6 U_{\text{ж}}$, $U_{\text{вих}} = 0,8 U_{\text{ж}}$ з затримкою 2 с, де $U_{\text{ж}}$ – напруга живлення.

Перевірити роботу програми на стенді або, у разі дистанційного навчання, створити та перевірити відповідну програму за допомогою емулятора UnoArduSim.

Програма

Контрольні питання

1. Описати, у яких системі числення задаються значення з ЦАП?

2. Визначити, який діапазон чисел має ЦАП з розрядністю $n=10$?

3. Описати, як знайти коефіцієнт перетворення масштабу?

4. Описати, як визначаються негативні числа у двійковій системі числення?

5. Описати, як створити сигнал змінної напруги за допомогою щиротно-імпульсної модуляції?

Висновки по роботі

ПРОТОКОЛ
до лабораторного заняття 6
Дослідження принципів обчислювання параметрів аналого-цифрових перетворювачів

Студент _____ група _____

Мета заняття: Дослідження принципів обчислювання параметрів аналого-цифрових перетворювачів.

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату та контролер Arduino, у разі дистанційного навчання, створити відповідну програму за допомогою емулятора UnoArduSim.

Завдання 1

Знайти точність перетворювання напруги у цифровий код для АЦП з вказаними *розрядністю та діапазоном вимірювання*.

ΔU	$(U_{\max} \square U_{\min}), B$	N
	10	10
	± 5	12
	1	16

Завдання 2

Скласти програму опитування аналогового входу 2, до якого підключений потенціометр (подільник напруги) з періодом 2 с. Отриману інформацію вивести на Монітор порта.

Перевірити роботу програми на стенді або, у разі дистанційного навчання, створити та перевірити відповідну програму за допомогою емулятора UnoArduSim.

Програма

Контрольні питання

1. Описати, у яких системі числення задаються значення з АЦП?

2. Визначити, який діапазон чисел має АЦП з розрядністю $n=10$?

3. Описати, як знайти коефіцієнт перетворення масштабу?

4. Визначити, яка функція використовується для опитування аналогового входу у контролера Arduino?

5. Визначити, яка функція використовується для перетворення масштабу?

Висновки по роботі

ПРОТОКОЛ
до лабораторного заняття 7
Дослідження принципів складання програм керування для мікроконтролерів

Студент _____ група _____

Мета заняття: Дослідження принципів складання програм керування для мікроконтролерів.

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату та контролер Arduino, у разі дистанційного навчання, створити відповідну програму за допомогою емулятора UnoArduSim.

Завдання 1

Скласти програму для керування швидкості двигуна постійного струму за допомогою потенціометра для контролера Arduino.

Завдання 2

Скласти програму для керування послідовністю переміщень сервопривода за допомогою контролера Arduino Nano з встановленням таких кутів повороту:

0°, 40°, 60°, 100°, 120°, 100°, 60°, 40°, 0°,

з затримкою між окремими переміщеннями 2 с.

Перевірити роботу програми на стенді або, у разі дистанційного навчання, створити та перевірити відповідну програму за допомогою емулятора UnoArduSim.

Програма 1

Програма 1

Контрольні питання

1. Визначити, як здійснюється керування двигуном постійного струму?

2. Описати, яка функція використовується в Arduino для управління виходом ШІМ?

3. Назвати, яка бібліотека використовується для керування кроковим двигуном?

4. Назвати, яка бібліотека використовується для керування сервоприводом?

5. Описати, як здійснюється регулювання швидкості обертання крокових двигунів?

Висновки по роботі

ПРОТОКОЛ
до лабораторного заняття 8
Дослідження принципів складання програм обробки даних для мікроконтролерів

Студент _____ група _____

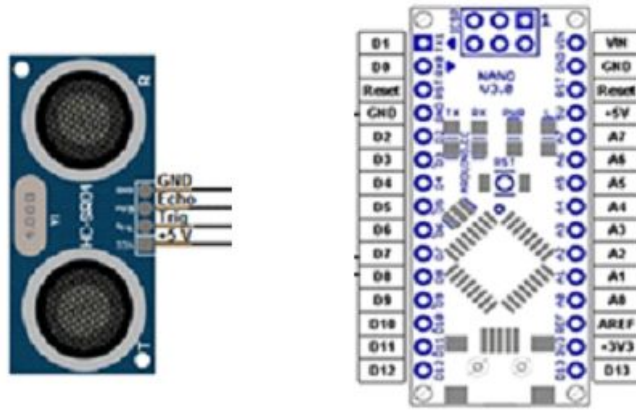
Мета заняття: Дослідження принципів складання програм обробки даних для мікроконтролерів.

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату, ультразвуковий датчик HC-SR04 та контролер Arduino, у разі дистанційного навчання, створити відповідну програму за допомогою емулятора UnoArduSim.

Завдання 1

Доповнити схему підключення датчика.



Скласти програму вимірювання відстані за допомогою ультразвукового датчика HC-SR04 з таким підключенням: Trig - пін 2, Echo - пін 3.

Ультразвуковий датчик керує переміщенням візка.

Візок їде прямо з максимальною швидкістю, якщо відстань до перешкоди більше 40 см.

Візок їде прямо з швидкістю 0,5 від максимальної, якщо відстань до перешкоди більше 20 см.

Якщо відстань до перешкоди менше ніж 20 см, візок зупиняється. При зникненні перешкоди рух продовжується.

Двигун підключений до виходу 5 Arduino.

Перевірити роботу програми на стенді або, у разі дистанційного навчання, створити та перевірити відповідну програму за допомогою емулятора UnoArduSim.

Програма 1

Контрольні питання

1. Назвати, яку функцію має контролер Arduino для перетворення масштабу?

2. Описати, як здійснюється формування вихідного імпульсу у контролера Arduino?

3. Описати, як здійснюється вимірювання відстані за допомогою ультразвукового датчика HC-SR04?

4. Описати, як здійснюється вимірювання тривалості імпульсів у контролерів Arduino?

5. Описати, як здійснюється перерахування дистанції отриманої за допомогою ультразвукового датчика у сантиметри?

Висновки по роботі

ПРОТОКОЛ
до лабораторного заняття 9
Дослідження принципів складання програм у вигляді функціональної схеми

Студент _____ група _____

Мета заняття: Дослідження принципів складання програм у вигляді функціональної схеми.

Опис лабораторної установки

Для перевірки роботи треба створити відповідну програму за допомогою мови LOGO!Soft Comfort, яка використовує логічні функції.

Завдання 1

Скласти блок-схему алгоритму та програму керування підйомником за допомогою мови LOGO!Soft Comfort.

Підйомник переміщається у двох напрямках: догори та униз. Для прямування нагору необхідно короткочасно натиснути кнопку "нагору". Для прямування униз необхідно короткочасно натиснути кнопку "униз".

Прямування нагору або униз припиняється при натисненні кнопки "стій" або при досягненні кінцевих вимикачів "верх" або "низ".

Для здійснення вмикання та вимикання двигуна використати функцію RS-тригера.

Передбачити захист від одночасного натискання кнопок догори та униз.

Перевірити роботу програму на емуляторі LOGO!Soft Comfort.

Програма 1

Контрольні питання

1. Описати, як здійснюється зміна напрямку обертання двигунів постійного струму?

2. Описати, як здійснюється захист від одночасного натиснення кнопок?

3. Назвати, яку функцію здійснює кінцевий вимикач верхнього положення?

4. Назвати, яку функцію здійснює кінцевий вимикач нижнього положення?

5. Описати, як працює кінцевий вимикач?

Висновки по роботі

ПРОТОКОЛ
до лабораторного заняття 10
Дослідження принципів складання програм циклового керування

Студент _____ група _____

Мета заняття: Дослідження принципів складання програм циклового керування.

Опис лабораторної установки

Для перевірки роботи треба створити відповідну програму за допомогою мови LOGO!Soft Comfort, яка використовує логічні функції.

Завдання 1

Скласти блок-схему алгоритму та програму циклового керування, що складається з послідовного переміщення з поверненням у вихідне положення.

Маніпулятор здійснює такі пересування: ВВЕРХ, ВНИЗ, ВПРАВО, ВЛІВО, ВПЕРЕД, НАЗАД.

Кожному пересуванню відповідає один вихід на виконавчий пристрій переміщення та один вихід для опитування кінцевого вимикача.

Треба скласти програму пересування, використовуючи функції пам'яті відповідно такій послідовності переміщень:

ВВЕРХ, ВПЕРЕД, ВПРАВО, ВНИЗ, ВВЕРХ, ВЛІВО, НАЗАД, ВНИЗ.

Перевірити роботу програму на емуляторі LOGO!Soft Comfort.

Програма 1

Контрольні питання

1. Описати, як здійснюється послідовне переміщення за допомогою функції пам'яті?

2. Назвати, у яке положення переміщується робочий орган в кінці циклу?

3. Описати, як здійснюється послідовне включення та виключення двигуна?

4. Назвати, який елемент включає двигун?

5. Назвати, який елемент виключає двигун?

Висновки по роботі

ПРОТОКОЛ
до лабораторного заняття 11
Дослідження принципів складання програм позиційного керування

Студент _____ група _____

Мета заняття: Дослідження принципів складання програм позиційного керування.

Опис лабораторної установки

Для перевірки роботи схеми використовувати монтажну плату та контролер Arduino, у разі дистанційного навчання, створити відповідну програму за допомогою емулятора UnoArduSim.

Завдання 1

Скласти блок-схему алгоритму та програму переміщення до визначеної позиції з використанням фотоімпульсного датчика на основі контролера Arduino.

Маніпулятор з позиційним керуванням здійснює такі переміщення: УВЕРХ, УНИЗ, ВПРАВО, ВЛІВО.

Вихідне положення визначається кінцевими вимикачами.

Кожному пересуванню відповідає один вихід на виконавчий пристрій переміщення та один вихід для опитування датчика пересування.

Позиціонування здійснюється під час руху УВЕРХ, УНИЗ та ВПРАВО, ВЛІВО.

Треба скласти програму пересування відповідно такій послідовності переміщень:

Переміщення ВПРАВО до позиції 4.

Переміщення УВЕРХ до позиції 2.

Перевірити роботу програми на стенді або, у разі дистанційного навчання, створити та перевірити відповідну програму за допомогою емулятора UnoArduSim.

Програма 1

Контрольні питання

1. Описати, як здійснюється позиційне переміщення?

2. Назвати, яке положення є вихідним?

3. Описати, яку функцію використовують для підрахунку дій?

4. Назвати, який елемент використовується як датчик положення?

5. Назвати, який елемент визначає вихідне положення?

Висновки по роботі
