

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
КАФЕДРА «ПІДЙОМНО-ТРАНСПОРТНОГО ТА
РОБОТОТЕХНІЧНОГО ОБЛАДНАННЯ»

МЕТОДИЧНІ ВКАЗІВКИ ДО ЛАБОРАТОРНИХ ЗАНЯТЬ

з дисципліни «Сучасні тенденції створення автоматизованих систем
управління вантажопідйомних машин»

Другий (магістерський) рівень вищої освіти
Спеціальність – 133 ГАЛУЗЕВЕ МАШИНОБУДУВАННЯ
Освітня програма – Підйомно-транспортні, будівельні дорожні
машини і обладнання

ОДЕСА 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
КАФЕДРА «ПІДЙОМНО-ТРАНСПОРТНОГО ТА
РОБОТОТЕХНІЧНОГО ОБЛАДНАННЯ»

Михайлов Євген Павлович

МЕТОДИЧНІ ВКАЗІВКИ ДО ЛАБОРАТОРНИХ ЗАНЯТЬ

з дисципліни «Сучасні тенденції створення автоматизованих систем
управління вантажопідійомних машин»

Другий (магістерський) рівень вищої освіти
Спеціальність – 133 ГАЛУЗЕВЕ МАШИНОБУДУВАННЯ
Освітня програма – Підйомно-транспортні, будівельні дорожні
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 дисципліни «Сучасні тенденції створення автоматизованих систем управління вантажопідійомних машин» для здобувачів другого (магістерського) рівня вищої освіти, спеціальності: 133 - Галузеве машинобудування, освітня програма: – Підйомно-транспортні, будівельні дорожні машини і обладнання. / Укл.: Михайлов Є. П. – Одеса: ОП, 2021. - 25 с.

Укладач:

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

В методичних вказівках розглянуті сучасні тенденції автоматизації керування вантажопідійомними машинами. Розглянуті питання проведення досліджень параметрів типових динамічних ланок систем автоматизованого керування, інформаційних та виконавчих пристроїв та сучасних систем автоматизованого керування вантажопідійомними машинами.

Зміст

Вступ	4
Лабораторне заняття 1. Дослідження принципів обчислювання параметрів типових динамічних ланок	5
Лабораторне заняття 2. Дослідження імпульсних датчиків переміщення	7
Лабораторне заняття 3. Дослідження ультразвукових датчиків положення.....	9
Лабораторне заняття 4. Дослідження двигунів постійного струму.....	13
Лабораторне заняття 5. Дослідження апаратних компонент систем програмного керування вантажопідійомними машинами.....	18
Лабораторне заняття 6. Дослідження програмних компонент систем керування вантажопідійомними машинами.....	22
Лабораторне заняття 7. Дослідження засобів захисту від перенавантаження за допомогою ваговимірювальних пристроїв на основі контролерів Arduino.....	23
ЛІТЕРАТУРА	25

Вступ

Засоби автоматизованого керування широко використовуються для автоматизації управління транспортно-складськими процесами в логістиці, гнучких виробничих систем та вантажопідйомних машинах. 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 «Автоматизація управління транспортно-складськими процесами в логістиці», «Автоматизація управління ГВС» та «Сучасні тенденції створення автоматизованих систем управління вантажопідйомних машин» розглянуті питання застосування пристроїв програмного керування на основі мікроконтролерів та програмованих логічних контролерів для вирішування завдань програмованого керування різних засобів автоматизації.

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. 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: середа розробки Arduino IDE, симулятор UnoArduSim, LOGO!Soft Comfort (Demo) [5, 6, 7].

Лабораторне заняття 1. Дослідження принципів обчислювання параметрів типових динамічних ланок

Мета заняття: Продемонструвати розуміння та вміння проведення досліджень принципів обчислювання параметрів типових динамічних ланок.

Для проведення дослідження треба:

- знати особливості визначення системі рівнянь, що описує типові динамічні ланки;
- вміти створювати передавальну функцію типових динамічних ланок;
- навчитися проводити дослідження математичної моделі типових динамічних ланок.

Короткі теоретичні положення

На рис. 1 наведена схема RC -кола інтегруючого типу. Постійна часу кола $\tau = RC$.

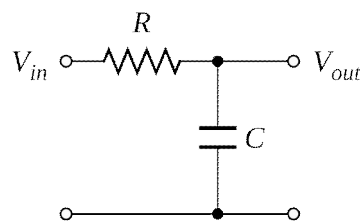


Рис. 1. Схема RC -кола інтегруючого типу

Перехідна функція RC -кола інтегруючого типу.

Реакція кола інтегруючого типу на одиничний вхідний перепад з амплітудою V визначається наступною формулою:

$$V_c(t) = V \left(1 - e^{-t/RC} \right).$$

На рис. 2 наведена схема RC -кола диференціюючого типу. Постійна часу кола $\tau = RC$.

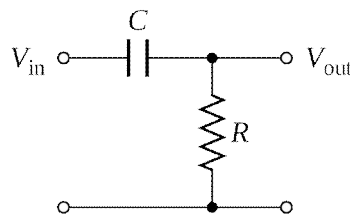


Рис. 2. Схема RC -кола диференціюючого типу

Перехідна функція RC -кола диференціюючого типу.

Реакція кола диференціюючого типу на одиничний вхідний перепад з амплітудою V визначається наступною формулою:

$$V_c(t) = V e^{-t/RC}.$$

Приклади розв'язання задач з теми заняття.

На рис. 3 наведені перехідні функції RC -кола інтегруючого (а) та диференціюючого (б) типу.

Для передачі даних вимірювань в комп'ютер можна використовувати вбудовані інструменти IDE Arduino, а саме, "Монитор порта" ("Монітор порту") для виведення чисельних даних і "Плоттер по послідовальному соединению" ("Плоттер по послідовному з'єднанню") для виведення графіків в режимі реального часу.

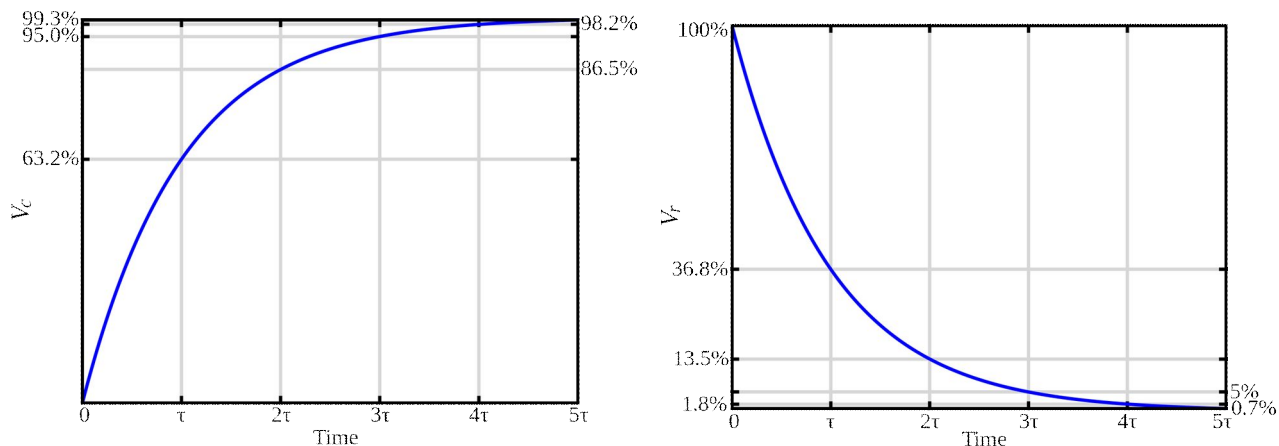


Рис. 3. Перехідні функції RC -кола інтегруючого (а) та диференціюючого (б) типу.

Приклади представлення даних у вигляді послідовності чисел і графіка наведені на рис. 4.

Ось Y показує дані, які передаються по послідовному порту (USB). Ось X складається з 500 позицій, і кожен з яких послідовно виводиться значення, відповідне одному виконання команди `Serial.println ()`.

Частота виведення даних, що вимірюються, на екран відповідає частоті виконання команди `Serial.println ()`.

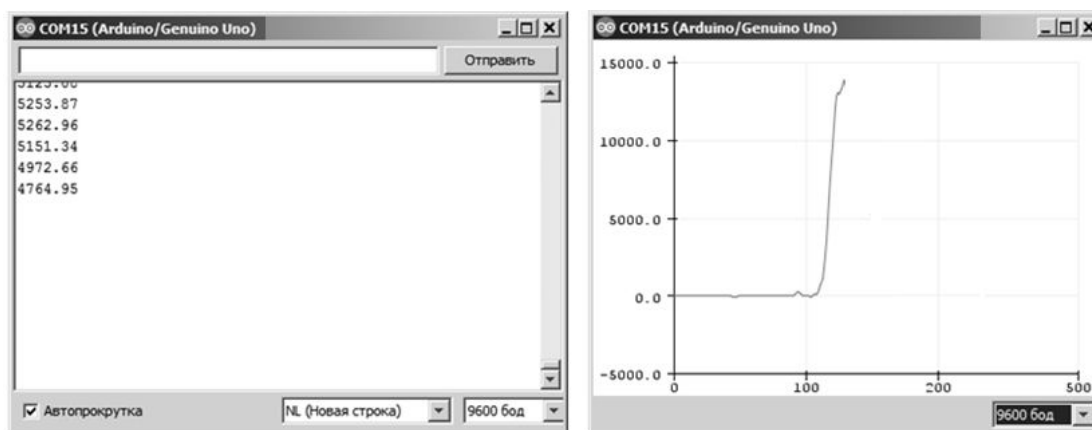


Рис.. Приклади представлення даних у вигляді послідовності чисел і графіка

Завдання

По заданим значенням R та C знайти перехідну характеристику RC -кола інтегруючого та диференціюючого типу.

Вихідні дані для завдання

Варіант	1	2	3	4	5	6	7	8	9	10
$R, кОм$	1,0	1,5	0,5	4,7	10,0	120,0	47,0	1,8	2,4	5,1
$C, мкФ$	10,0	22,0	47,0	68,0	100,0	220,0	470,0	680,0	1000,0	2200,0

Контрольні запитання

1. Які особливості має аперіодична ланка?
2. Які особливості має інтегруюча ланка?
3. Які особливості має диференціююча ланка?

Лабораторне заняття 2. Дослідження імпульсних датчиків переміщення

Мета заняття: Продемонструвати розуміння та вміння проведення досліджень імпульсних датчиків переміщення.

Для проведення дослідження треба:

- знати особливості використання імпульсних датчиків переміщення;
- вміти визначати характеристики імпульсних датчиків переміщення згідно з метою використання;
- навчитися визначати точність визначення положення за допомогою імпульсних датчиків переміщення.

Короткі теоретичні положення

Для визначення шляху переміщення, яке здійснюється за допомогою коліс, часто використовують імпульсні датчики. При цьому визначення шляху переміщення здійснюється за допомогою вимірювання кута переміщення колеса.

Шлях l , що проходить колесо при обертанні на кут φ дорівнює

$$l = d \varphi / 2, \text{ або } l = \pi d \varphi / 360^\circ,$$

де d - діаметр колеса, φ та φ° кут обертання колеса, відповідно, у радіанах або градусах.

Це дає можливість визначити зв'язок кута обертання колеса та шляху, який проходить робот при переміщенні на відстань l .

Для вимірювання кута повороту використовують найчастіше оптичні та індуктивні імпульсні датчики.

Принци дії оптичного датчика для визначення переміщення показаний на рис. 1.



Рис. 1. Використання оптичних датчиків для визначення переміщення

Високу точність забезпечують фотоімпульсні датчики, які дають до 6000 імпульсів за одне обертання.

При цьому імпульсні датчики видають визначену кількість імпульсів n_δ на одне обертання колеса.

Шлях l_δ , який проходить транспортний засіб за одне обертання колеса дорівнює

$$l_\delta = \pi d.$$

Відповідно кількість імпульсів n для переміщення на відстань l , дорівнює

$$n_l = n_\delta l / \pi d.$$

При цьому похибка переміщення (відстань між двома імпульсами) l_n буде складати:

$$l_n = l_\delta / n_\delta = \pi d / n_\delta.$$

Використовуючи цю формулу можна визначити кількість імпульсів, яка потребується для переміщення на відстань l по прямій, або кількість імпульсів для кожного привода при переміщенні по дузі кола транспортного засобу з диференціальним приводом для коліс з діаметром d та кількістю імпульсів на одне обертання колеса n_δ .

Приклади розв'язання задач з теми заняття

Задані такі параметри для переміщення автоматично керованого транспортного засобу:

параметри траєкторії:

переміщення по прямій лінії на відстань $l_V = 10$ м;

характеристики транспортного засобу:

діаметр колеса $d = 0,1$ м;

Кількість імпульсів на одне обертання колеса для встановленого датчика дорівнює 20.
Згідно з наведеними формулами отримаємо

$$n_l = n_d l / \pi d = 20 \cdot 10 / 3,14 \cdot 0,1 = 637;$$

Похибка переміщення буде складати:

$$l_n = \pi d / n_d = 3,14 \cdot 0,1 / 20 = 0,0157 \text{ м.}$$

Вихідні дані для завдань

Для заданих параметрів переміщення автоматично керованих транспортних засобів для вказаних імпульсних датчиків визначити кількість імпульсів для переміщення на вказану відстань та похибку переміщення.

Вихідні дані

Варіант	Діаметр колеса d , м	Переміщення по прямій лінії на відстань l , м	Кількість імпульсів на одне обертання колеса	Кількість імпульсів для переміщення на вказану відстань	Похибка переміщення
1	0,2	10	10		
2	0,3	12	20		
3	0,24	14	40		
4	0,4	15	60		
5	0,35	12,7	50		
6	0,2	15	70		
7	0,4	14	24		
8	0,24	12,7	55		
9	0,35	10	36		
10	0,3	12	14		

Контрольні запитання

1. Як здійснюється визначення положення за допомогою імпульсних датчиків переміщення?
2. Які датчики використовують для визначення положення?
3. Як пов'язати параметри датчика кута обертання колеса з шляхом переміщення?

Лабораторне заняття 3. Дослідження ультразвукових датчиків положення

Мета заняття: Продемонструвати розуміння та вміння проведення досліджень ультразвукових датчиків положення.

Для проведення дослідження треба:

- знати особливості використання ультразвукових датчиків для визначення положення;
- вміти визначати характеристики ультразвукових датчиків для визначення положення згідно з метою використання;
- навчитися визначати положення за допомогою ультразвукових датчиків.

Короткі теоретичні положення

Швидкість розповсюдження ультразвуку у повітрі 343, 1 м/с (при температурі повітря 20°C). При підвищенні або зменшенні температури в діапазоні від 0°C до 40°C швидкість відповідно змінюється на 0,58 м/с на градус. Таким чином при зміні температури від 0°C до 40°C швидкість розповсюдження ультразвуку у повітрі зміниться на 23,2 м/с, що складає 7% від значення температури при 0°C. Тому для мобільних роботів, що працюють на відкритому просторі, при вимірюванні відстані до об'єктів треба враховувати температуру зовнішнього середовища.

Ультразвуковий датчик визначає відстань до перешкоди шляхом вимірювання часу від випромінювання до повернення ультразвукового імпульсу у мікросекундах.

На рис 1 наведено підключення ультразвукового датчика HC-SR04 до контролера.

Датчик має 4 виводи:

VCC: "+" живлення;

TRIG (T): вхідний сигнал, підключений до виходу 9;

ECHO (R): вихідний сигнал, підключений до входу 8;

GND: "-" живлення (земля).

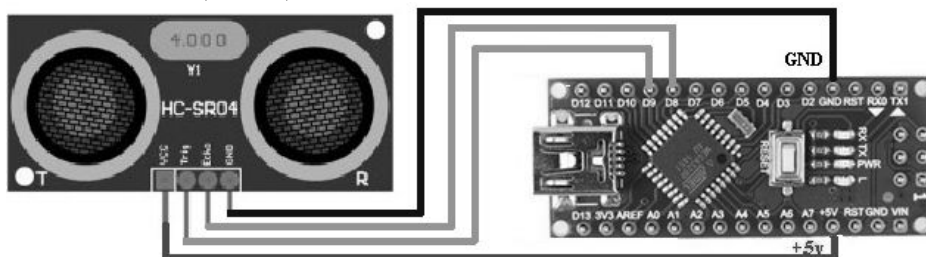


Рис. 1. Підключення далекоміра на основі ультразвукового датчика до контролера

Треба скласти алгоритм вимірювання відстані до перешкоди з урахуванням температури повітря, та програму, що реалізує цей алгоритм за допомогою контролера Arduino (Основи програмування наведені у Додатку).

Для визначення температури можна використовувати термістори, опір яких залежить від температури. Для обчислення температури шляхом вимірювання опору термістора використовують рівняння Стейнхарта-Харта, яке у спрощеній формі виглядає так:

$$1/T = 1/T_0 + \ln(R/R_0) / B,$$

де T_0 – кімнатна температура, що дорівнює 25°C або 298,15 K;

T – температура, що визначається, за Кельвіном,

B - коефіцієнт, що залежить від типу термістора (у нашому випадку $B = 3950$);

R – опір термістора для температури, що визначається;

R_0 - опір термістора для кімнатної температури (у нашому випадку $R_0 = 10 \text{ КОм}$).

Маємо температуру за Цельсієм:

$$T = 1 / (1/T_0 + \ln(R/R_0) / B) = 1 / (0,003354 + \ln(10/R) / 3950) - 273.15.$$

Так, якщо шляхом вимірювання опору термістора получено значення $R = 6,504$, отримаємо $T = 308,15 \text{ K}$ або $T = 35^\circ\text{C}$.

Практична частина

Скласти алгоритм та програму вимірювання відстані до перешкоди в міліметрах з урахуванням температури повітря.

Приклад вирішування задачі

Вихідні дані:

Температура 20°C ;

Відстань до перешкоди, 300 мм.

Швидкість розповсюдження ультразвуку у повітрі при температурі повітря 20°C складає 343, 1 м/с, або 34310 см/с.

Визначаємо коефіцієнт перерахунку результату опитування датчика в відстань в сантиметрах, на який треба поділити результат опитування датчика:

$$K_{us} = 1000000 \text{ мкс} * 2 / 34310 \text{ см/с} = 58,29 \approx 58.$$

1. Алгоритм роботи можна умовно розділити на 4 етапи:

- 1) подаємо імпульс тривалістю 10 мкс, на вивід Trig;
- 2) датчик перетворює вхідний імпульс в 8 імпульсів частотою 40 КГц і надсилає вперед через випромінювач "Т";
- 3) дійшовши до перешкоди, послані імпульси відбиваються і поступають на приймач "R", внаслідок чого отримуємо вихідний сигнал на виводі Echo, тривалість якого пропорційна відстані до об'єкту;
- 4) функція pulseIn(Echo, HIGH) повертає тривалість імпульсів в мікросекундах
- 5) перетворюємо отриманий сигнал в відстань за формулою: тривалість імпульсу (мкс) / K_{us} = дистанція (см)

2. Програма для вимірювання відстані до об'єкту в сантиметрах. Якщо відстань менше або дорівнює 30 см, включається світлодіод на виході 13.

Код програми

```
#define Trig 9
#define Echo 8
#define ledPin 13          //світлодіод
void setup() {
  pinMode(Trig, OUTPUT); //визначаємо як вихід
  pinMode(Echo, INPUT);  // визначаємо як вхід
  pinMode(ledPin, OUTPUT); //визначаємо як вихід
}
unsigned int Kus=58;
unsigned int impulseTime=0;
unsigned int distance_sm=0;
void loop() {
  digitalWrite(Trig, HIGH); // Подаємо імпульс на вхід trig
  delayMicroseconds(10);    // що дорівнює 10 мікросекунд
  digitalWrite(Trig, LOW);  // відключаємо
  impulseTime=pulseIn(Echo, HIGH); // Вимірюємо тривалість імпульсу
  distance_sm=impulseTime/Kus; // Перераховуємо в сантиметри
  if(distance_sm>30) {      //Якщо відстань більше ніж 30 см
    digitalWrite(ledPin, LOW); // Світлодіод не горить
  }
  else {                    // Якщо відстань менше або дорівнює 30 см
```

```
digitalWrite(ledPin, HIGH); // Світлодіод горить
}
delay(50); } }
```

3. Програма для вимірювання температури за Кельвіном

```
Код програми
float R; //вихідні дані R0
float T; //результат T
void setup() {
  Serial.begin(9600);
  Serial.println("T = 1/(0,003354 + ln (R/10) / 3950)");
}
void loop() {
  // Коли в буфері є дані
  while (Serial.available() > 0)
  {
    R = Serial.parseFloat(); // Отримане число
    if (Serial.read() == '\n') // Кінець передачі
    {
      T = 1 / (0.003354 + (log(R/10))/3950);
      Serial.print("R = \t");
      Serial.println(R, 4); //вихідні дані R
      Serial.print("T = \t");
      Serial.println(T, 4); //вихідні дані T
    } } }
```

Вихідні дані для завдань

1. Скласти алгоритм вимірювання відстані до перешкоди в міліметрах з урахуванням температури повітря.
2. Для заданої температури повітря визначити формулу перерахунку та перерахувати швидкість розповсюдження ультразвуку.
3. Визначити коефіцієнт перерахунку результату опитування датчика в відстань в міліметрах, на який треба поділити результат опитування датчика
4. Скласти програму, що реалізує цей алгоритм за допомогою контролера Arduino для вказаного варіанта значення температури та включає світлодіод, що підключений до виходу 13, якщо відстань до перешкоди менше встановленого варіанта значення. Для визначення змінних використовувати формат **float**.

Варіант	1	2	3	4	5	6
Температура °C	10	15	25	30	35	40
Відстань до перешкоди, мм	250	200	260	100	350	340
Варіант	7	8	9	10	11	12
Температура °C	10	15	25	30	35	40
Відстань до перешкоди, мм	100	350	340	250	200	260

5. У прикладах наведена програма для визначенні температури за Кельвіном. Змінити програму для визначення температури по Цельсію.

Визначити температуру, якщо отримані такі значення для опору термістора R:

Варіант	1	2	3	4	5	6
Опір термістора R	13,6987	11,9441	10,44	9,14743	7,3771	6,23934

Варіант	7	8	9	10	11	12
Опір термістора R	32,96	28,276	24,3274	20,9889	18,1582	15,7511

Контрольні запитання.

1. Визначити, як змінюється швидкість розповсюдження звуку в залежності від температури?
2. Описати, як знайти коефіцієнт перерахунку результату опитування датчика в відстань в міліметрах?
3. Описати, як знайти коефіцієнт перерахунку результату опитування датчика в відстань в сантиметрах?
4. Визначити, як здійснюється вимірювання температури?
5. Назвати, які зміни треба внести в програму для вимірювання відстані до об'єкту для визначення відстані в міліметрах та урахування температури?

Лабораторне заняття 4. Дослідження двигунів постійного струму

Мета заняття: Продемонструвати розуміння та вміння проведення досліджень моделі двигуна постійного струму.

Для проведення дослідження треба:

- знати особливості використання двигунів постійного струму;
- вміти визначити характеристики двигунів постійного струму згідно з метою використання;
- навчитися проводити регулювання швидкості обертання двигунів постійного струму.

Короткі теоретичні положення

Для керування двигуном постійного струму треба вирішувати дві задачі:

- 1) керування швидкістю обертання;
- 2) керування напрямком обертання.

Для керування швидкістю обертання двигуна постійного струму використовують сигнали з широтно-імпульсною модуляцією.

У широтно-імпульсному перетворювачі частота проходження імпульсів постійна, а тривалість імпульсів плавно змінюється, що забезпечує регулювання, як це показано на рис. 1.

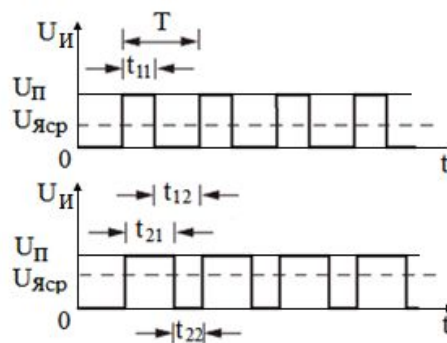


Рис. 1. Сигнал з широтно-імпульсною модуляцією

Для керування напрямком обертання у двигунів постійного струму використовують так званий Н-міст

Схема, що наведена на рис. 2, а, дає можливість здійснювати реверсивне керування.

Якщо замкнуті ключі (контакти реле, транзистори, тиристори тощо) S1 та S4, двигун обертається в одну сторону - прямий напрямок (рис. 2, б), якщо замкнуті ключі S2 та S3, двигун обертається в іншу сторону - зворотний напрямок (рис. 2, в).

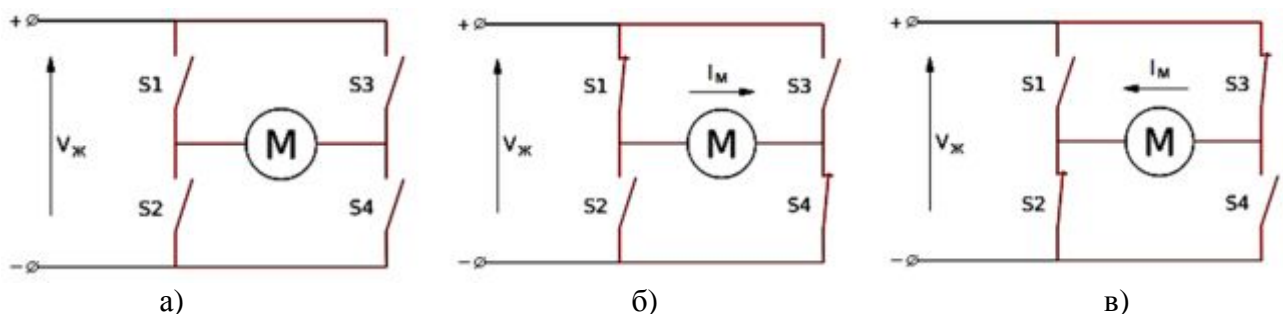


Рис. 2. Керування напрямком обертання двигунів постійного струму за допомогою Н-моста

Схема, що наведена на рис. 3, представляє собою Н-міст на IGBT-транзисторах.

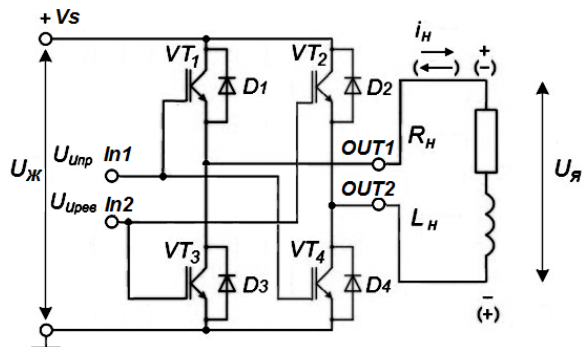


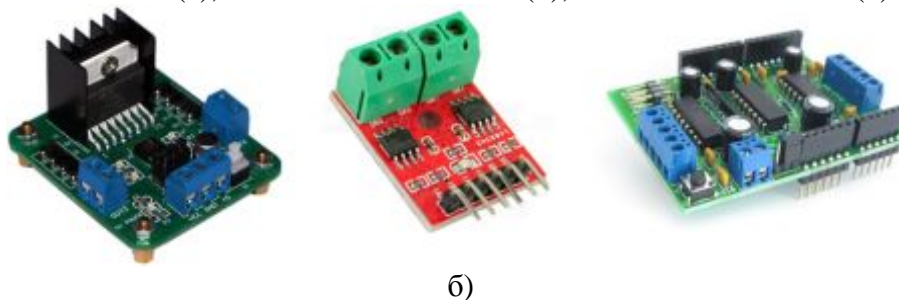
Рис. 3. H-міст на біполярних транзисторах

Імпульсна напруга $U_{\text{іпр}}$ відкриває ключові транзистори VT_1 , VT_4 , і струм через навантаження (двигун) тече в прямому напрямку, імпульсна напруга $U_{\text{ірев}}$ відкриває ключові транзистори VT_2 , VT_3 , і струм через навантаження (двигун) тече в зворотному напрямку (реверс).

Сигнали на виходах $OUT1$ та $OUT2$ мають ті ж значення, що й на відповідних входах $In1$ та $In2$, тому включення обертання двигуна в одну сторону на вхід $In1$ високий рівень сигналу, а на вхід $In2$ низький, а для обертання в іншу сторону, навпаки.

Для регулювання швидкістю обертання двигуна використовують сигнал з широтно-імпульсною модуляцією (ШІМ). Цей сигнал можна подавати на один з входів.

На рис. 4 наведені силові модулі для мікроконтролерів Arduino, що мають виходи з ШІМ Motor Driver L298N (а), Motor Driver L9110S (б), Motor Shield L293D (в).



а)

б)

в)

Рис. 4. Силові модулі для мікроконтролерів Arduino, що мають виходи з ШІМ Motor Driver L298N (а), Motor Driver L9110S (б), Motor Shield L293D (в).

Приклади розв'язання завдання

Розглянемо, як працюють модулі управління електродвигунами на прикладі драйвера двигуна Motor Driver L298N. Модуль виконаний на основі мікросхеми L298N, яка представляє собою два мостових драйвера, і може бути використаний для управління двома двигунами постійного струму (DC моторів). (рис. 5).

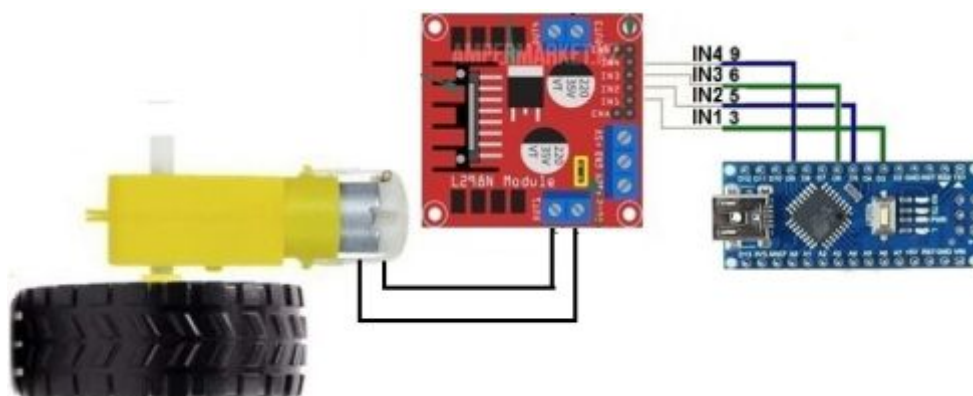


Рис. 5. Приклад підключення DC мотора до контролера

Розглянемо, як здійснюється керування двигуном постійного струму за допомогою контролерів Arduino.

Керування без регулювання швидкості здійснюється за допомогою функції:

```
digitalWrite(pin, value);
```

де:

pin – номер виходу контролера,

value – змінна типу bit, яка задає логічний рівень HIGH, або LOW (включити або виключити) на заданому цифровому виводі pin.

Далі наведений приклад програми керування двигуном постійного струму, що підключений до двох виходів контролера Arduino NANO.

Згідно з програмою двигун виконує такі дії

- обертання на протязі 2 с в одну сторону,
- зупинка на 1 с,
- обертання на протязі 2 с в другу сторону,
- зупинка на 1 с,
- потім все повторюється.

Приклад програми керування двигуном постійного струму

```
const int IN1 = 3;
const int IN2 = 5; //підключення входів модуля до відповідних контактів
void setup() {
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT); }
void loop() {
  digitalWrite(IN1, HIGH); //обертання вперед,
  digitalWrite(IN2, 0);
  delay(2000); //затримка 2 с
  digitalWrite(IN1, LOW); //зупинка
  digitalWrite (IN2, LOW);
  delay(1000); //затримка 1 с
}
```

Для керування швидкістю обертання сигнал з ШІМ можна подати на один з входів.

Розглянемо, як здійснюється керування двигуном постійного струму за допомогою контролерів Arduino, які мають можливість видавати сигнали з широтно-імпульсною модуляцією (ШІМ).

Управління виходом ШІМ контролерів Arduino при використанні програмної оболонки IDE здійснюється 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 та у даній програмі змінює швидкість обертання мотору.

Регулювання швидкості здійснюється за допомогою функції:

```
analogWrite(pin, value);
```

де:

pin – номер виходу контролера, який підтримує ШІМ, при цьому треба враховувати, що у контролерів Arduino UNO та Arduino NANO ШІМ підтримують виходи 3, 5, 6, 9, 10, 11.

value – змінна типу byte, яка задає тривалість імпульсу, та має значення значення від 0 до 255.

Для керування швидкістю обертання двигунів можна використовувати потенціометр, сигнал з яких подається на аналогові входи (рис. 6).

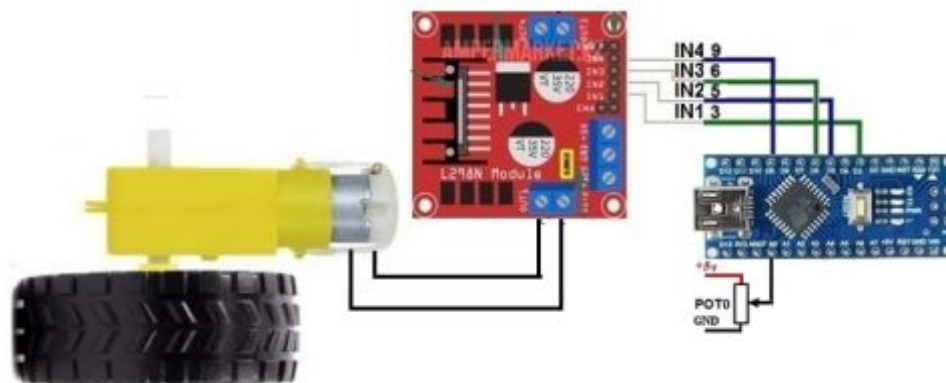


Рис. 6. Керування швидкістю обертання за допомогою потенціометра

Оскільки аналоговий вхід видає значення від 0 до 1023, а для регулювання швидкістю за допомогою ШІМ треба задавати значення від 0 до 255, здійснюється масштабування значення за допомогою функції

`map(value, fromLow, fromHigh, toLow, toHigh),`

яка перетворює значення змінної з одного діапазону в інший, а саме значення змінної `value`, що дорівнює `fromLow`, буде перетворено в число `toLow`, а значення `fromHigh` - в `toHigh`.

Всі проміжні значення `value` масштабуються відносно нового діапазону [`toLow`; `toHigh`].

Програма керування двигунами, яка здійснює обертання в прямому та зворотному напрямку з регулюванням швидкості обертання окремо для обертання в прямому та зворотному напрямках (потенціометр POT0). Значення потенціометрів виводиться на монітор за допомогою функції `Serial.println(var)` значення змінної `speed` після масштабування.

```
const int IN1 = 3;
const int IN2 = 5; //підключення входів модуля до відповідних контактів
int speed = 0; //швидкість обертання
#define POT0 0 //потенціометр

void setup() {
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  Serial.begin(9600); }

void loop() {
  speed = analogRead(POT0);
  speed = map(speed, 0, 1023, 0, 255);
  Serial.println(speed);
  analogWrite(IN1, speed); //обертання вперед
  analogWrite(IN2, 0);
  delay(2000); //затримка 2 с
  analogWrite(IN1, 0); //зупинка
  analogWrite(IN2, 0);
  delay(1000); //затримка 1 с
  analogWrite(IN1, 0); //обертання назад
  analogWrite(IN2, speed);
  delay(2000); //затримка 2 с
  analogWrite(IN1, 0); //зупинка
  analogWrite(IN2, 0);
  delay(1000); //затримка 1 с }
```


Перелік завдань для розв'язання.

Завдання

Скласти програму керування двох двигунів постійного струму з реверсом та регулюванням швидкості обертання з використання чотирьох виходів з ШІМ для наведеної схеми (рис. 7).

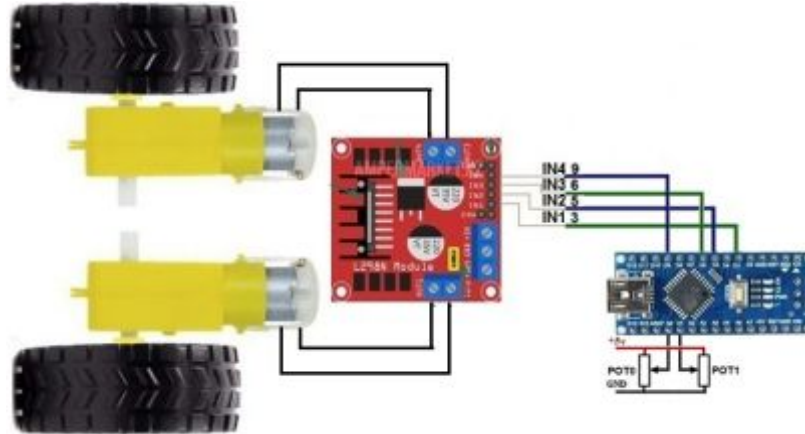


Рис. 7. Схема керування двох двигунів постійного струму з реверсом та регулюванням швидкості обертання з використання чотирьох виходів з ШІМ

Завдання 2 (для дистанційного навчання)

Перевірити роботу програм за допомогою емулятора UnoArduSim.

Контрольні запитання

1. Як визначити систему рівнянь, що описує двигун постійного струму?
2. Як визначити передавальну функцію двигуна постійного струму?
3. Як створити математичну модель двигуна постійного струму?

Лабораторне заняття 5. Дослідження апаратних компонент систем програмного керування вантажопідйомними машинами

Мета заняття: Продемонструвати розуміння та вміння проведення досліджень апаратних компонент систем програмного керування вантажопідйомними машинами.

Для проведення дослідження треба:

- знати особливості створення проекту керування вантажопідйомними машинами;
- вміти створювати алгоритми керування вантажопідйомними машинами;
- вміти визначати апаратні компоненти систем керування вантажопідйомними машинами на основі алгоритму керування.

Короткі теоретичні положення

Розглянемо, як здійснюється керування підйомником, зображеного на рис. 1, за допомогою релейної схеми та програмного керування.

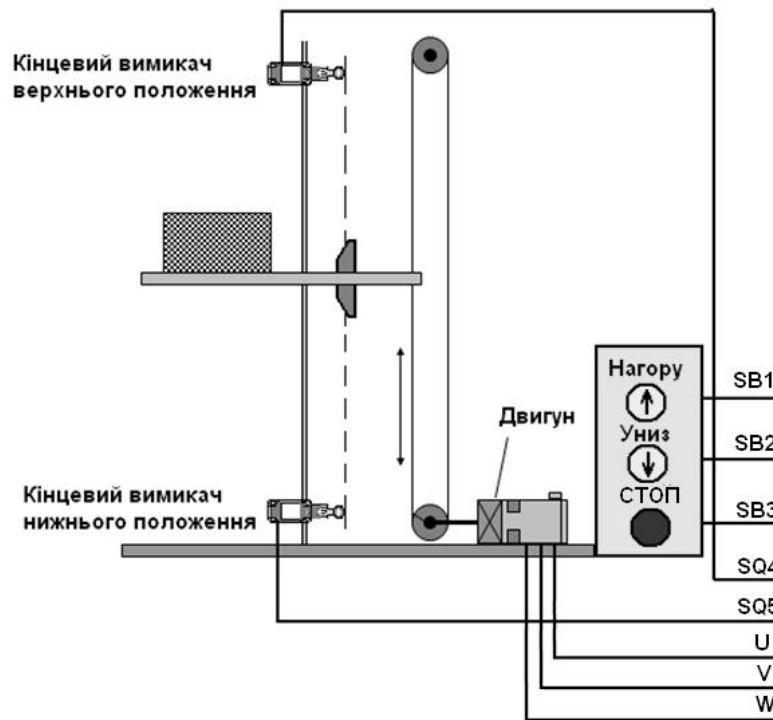


Рис. 1. Схематичне зображення підйомника

Релейна схема керування складається з головного кола, яке підключає двигун до мережі живлення, та допоміжного кола, яке здійснює саме керування (рис. 2).

Керування здійснюється за допомогою трьох кнопок "Нагору" (SB1), "Униз" (SB2), "СТОП" (SB3), кінцевих вимикачів верхнього (SQ4) та нижнього (SQ5) положень. У підйомнику використовується трифазний двигун (фази U, V, W). Кнопки "Нагору" (SB1), "Униз" (SB2) двоколові, вони використовують замикаючий та розмикаючий контакти. Замикаючі контакти вмикають рух у відповідному напрямку, змінюючи напрямок руху за допомогою контакторів КМ2 та КМ3. Ці контактори змінюють підключення двох фаз V та W, що в свою чергу змінює напрям обертання двигуна. Контакт КМ1 підключає двигун до силової мережі трифазного струму. Контакти, що розмикаються, кнопок "Нагору" та "Униз" блокують включення двигуна при одночасному натисканні кнопок "Нагору" та "Униз".

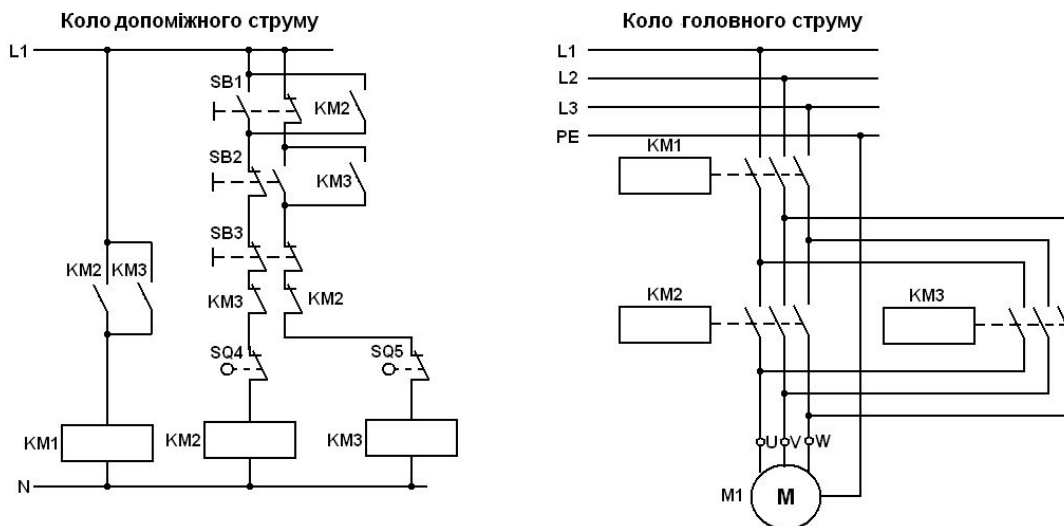


Рис. 2. Схема керування підйомником

Допоміжні контакти, що розмикаються, контакторів КМ2 та КМ3 блокують включення одного напрямку руху, якщо включений другий. Тому при натисканні кнопок "Нагору" або "Униз" розмикаючий контакт спочатку відключить рух у іншому напрямку та блокування допоміжного розмикаючого контакту, а тільки потім спрацює контактор руху у відповідному напрямку.

Допоміжні замикаючі контакти контакторів КМ2 та КМ3, що включені паралельно замикаючим контактам кнопок "Нагору" та "Униз", замикають коло живлення контакторів і дозволяють продовжити рух, якщо відпустити ці кнопки.

Кнопка "СТОП" теж двоколова, вона має два розмикаючих контакти. Це дозволяє підключити її до кіл живлення рухів в обох напрямках і відключити двигун незалежно від напрямку руху.

Контактор КМ1 підключає двигун до силової мережі, якщо включений один з напрямків руху.

Наведена схема включає різні функції логічного керування, реалізованих за допомогою релейних схем, а саме функції І, АБО та НІ, а також функції пам'яті. Релейні пристрої керування у певний час, а у деяких випадках і досі, знайшли дуже широке використання.

Недоліком таких систем керування є неможливість швидкої зміни алгоритму керування, бо це потребує зміни електричної схеми, яка саме і реалізує цей алгоритм.

Заміна релейної логіки на напівпровідникові логічні схеми, а потім і мікросхеми не зняла цей недолік, бо залишилась апаратна реалізація алгоритму керування.

Суттєва зміна підходу до будування систем керування здійснилась завдяки використанню цифрових обчислювальних пристроїв, які дають змогу створення алгоритму керування будь-якої складності за допомогою програм. Головною перевагою такого програмного, або комп'ютерного керування є можливість швидкої зміни програми, що в свою чергу змінює алгоритм керування та робить систему керування гнучкою.

Ця перевага знайшла своє втілення, наприклад, у так званих програмованих логічних контролерах (ПЛК), які мають модульну структуру та являють собою основу для створення гнучких промислових систем керування.

Розглянемо, як здійснюється керування підйомника, наведеного на рис. 1, за допомогою програмованого логічного контролера (рис. 3).

Треба зазначити, що комп'ютерне керування у даному випадку замінює коло допоміжного струму, бо саме там закладений алгоритм керування. Коло головного струму, наведене на рис. Д1.5, залишається, бо воно здійснює підключення двигуна до мережі живлення та виконує функції силового перетворювача, який за допомогою вихідних сигналів ПЛК підключає двигун до мережі живлення та змінює його напрям обертання.

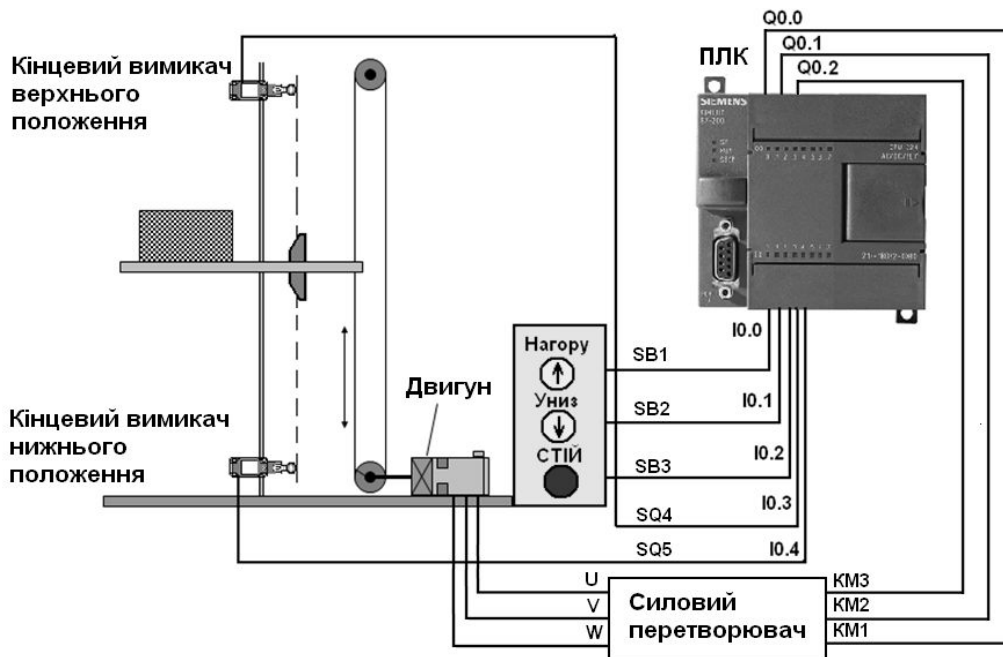


Рис. 3. Керування підйомника за допомогою програмованого логічного контролера

Одна з форм представлення програм у сучасних ПЛК, які більш детально будуть розглянуті далі, є форма у вигляді контактної плану (релейно-контактної схеми), що близька по своєму вигляду до електричної схеми, тому використаємо цю форму для програми керування підйомником.

На рис. 4 наведене порівняння програми керування з колом допоміжного струму (див. рис. 3), яке здійснює алгоритм керування.

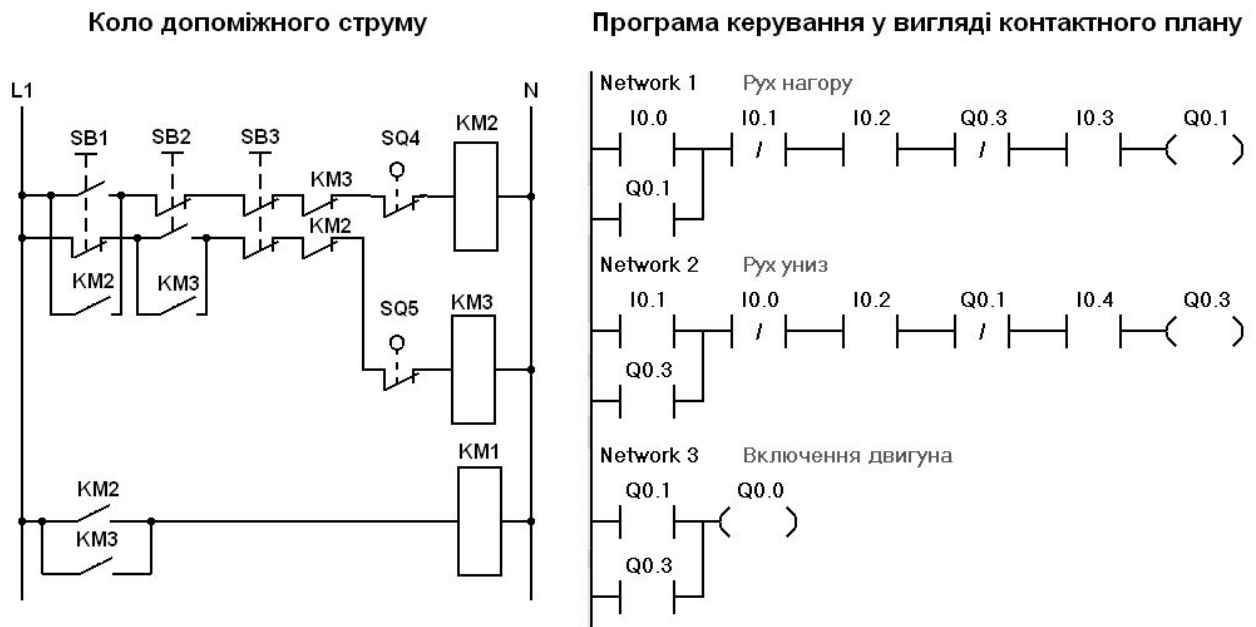


Рис. 4. Порівняння програми керування у вигляді контактної плану з колом допоміжного струму

Якщо треба змінити алгоритм керування системи релейного керування, це потребує зміни електричної схеми кола допоміжного струму. При використанні комп'ютерного керування зміна алгоритму здійснюється лише шляхом зміни програми, а електрична схема не змінюється.

Приклади розв'язання задач з теми заняття.

Перелік завдань для розв'язання.

Спроекувати апаратні компоненти для системи керування з використанням програмованого контролера LOGO!, наведеної на рис. 5.

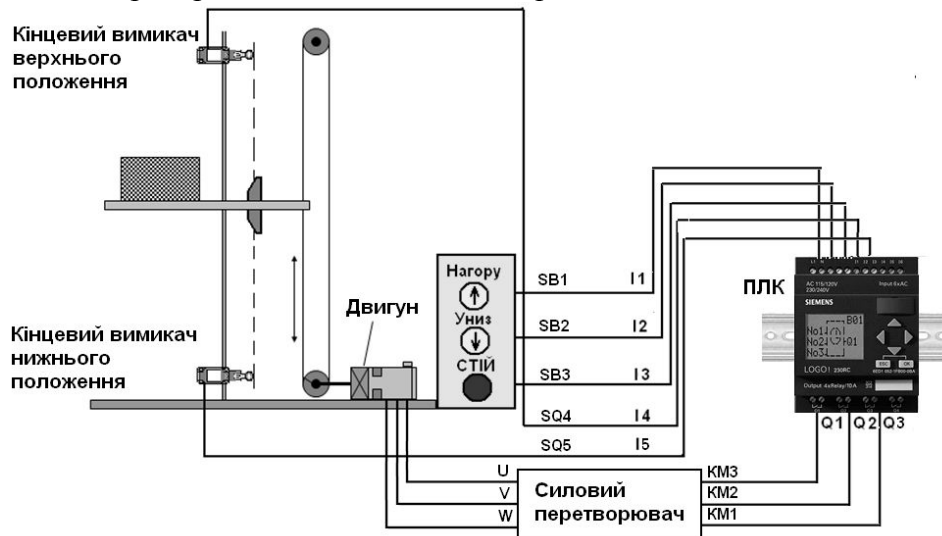


Рис. 5. Структура системи керування

Приклади розв'язання задач з теми заняття.

Програма у вигляді релейно-контактної схеми (LAD) для системи керування, що на рис. 5.7, наведена на рис. 6.

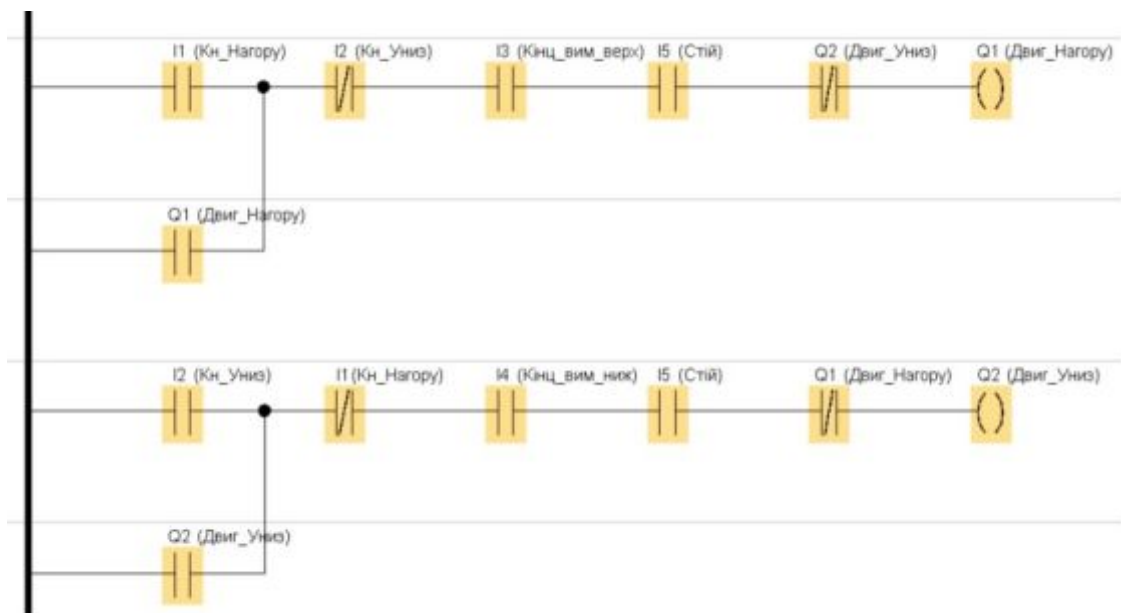


Рис. 6. Програма у вигляді релейно-контактної схеми (LAD)

Створити аналогічну програму у вигляді функціональної схеми (FBD).

Контрольні запитання.

1. Як створити проект керування?
2. Як визначити апаратних компонент згідно з поставленою задачею?
3. Як здійснити налагодження апаратних компонент?

Лабораторне заняття 6. Дослідження програмних компонент систем керування вантажопідйомними машинами

Мета заняття: Продемонструвати розуміння та вміння проведення досліджень програмних компонент систем керування вантажопідйомними машинами.

Для проведення дослідження треба:

- знати особливості створення програмних компонент систем керування вантажопідйомними машинами;
- вміти створювати алгоритм керування вантажопідйомними машинами;
- навчитися здійснювати налагодження програмних компонент систем керування вантажопідйомними машинами.

Короткі теоретичні положення

Теоретичні положення викладені у попередній лабораторній роботі.

Приклади розв'язання задач з теми заняття.

На рис. 1 наведена програма у вигляді релейно-контактної схеми (LAD) з використанням функції Реле фіксації (RS-тригер). Кнопка Пуск включає Двигун, кнопка Стоп виключає Двигун.

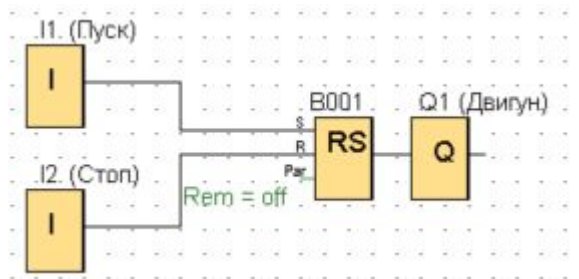


Рис. 1. Приклад використання лічильника

Перелік завдань для розв'язання.

У попередній лабораторній роботі використовувались базові логічні функції. Скласти програму у вигляді релейно-контактної схеми (LAD) та функціональної схеми (FBD), яка здійснює керування підйомником, наведеним на рис. 5 у попередній лабораторній роботі, з використанням Реле фіксації (RS-тригер).

Контрольні запитання

1. Які операції виконують основні логічні функції?
2. Як створити програму логічного керування?
3. Які засоби існують для налагодження програм?

Лабораторне заняття 7. Дослідження засобів захисту від перенавантаження за допомогою ваговимірювальних пристроїв на основі контролерів Arduino

Мета заняття: Продемонструвати розуміння та вміння проведення досліджень засобів захисту від перенавантаження за допомогою ваговимірювальних пристроїв та алгоритмів обробки отриманих даних на основі контролерів Arduino

Для проведення дослідження треба:

- знати особливості створення апаратних компонент ваговимірювальних пристроїв;
- вміти створювати програмні компоненти ваговимірювальних пристроїв;
- навчитися здійснювати налагодження ваговимірювальних пристроїв.

Теоретична частина

Для захисту від перенавантаження становить інтерес використання компактних систем зважування на основі апаратно-програмного комплексу Arduino, в складі якого є модуль зважування HX711. Цей модуль призначений для вимірювання сигналу тензOMETричних датчиків виконаних у вигляді моста Уитстона.

Схема підключення тензOMETричного датчика до контролера Arduino за допомогою модуля зважування HX711 наведена на рис. 7.1.

Основою цієї системи зважування є високоточний двоканальний модуль, побудований на мікросхемі HX711, яка має аналого-цифровий перетворювач з дозволом 24 біта, що забезпечує високу точність вимірювання.

Модуль має 2 канали А і В, до яких можна підключити два тензодатчика. Для каналу А коефіцієнт посилення дорівнює 64 або 128 (для сигналів, відповідно, ± 40 мВ ± 20 мВ), для каналу В коефіцієнт посилення дорівнює 32 (для сигналів ± 80 мВ). Модуль працює з частотою вимірювання 10 або 80 Гц.

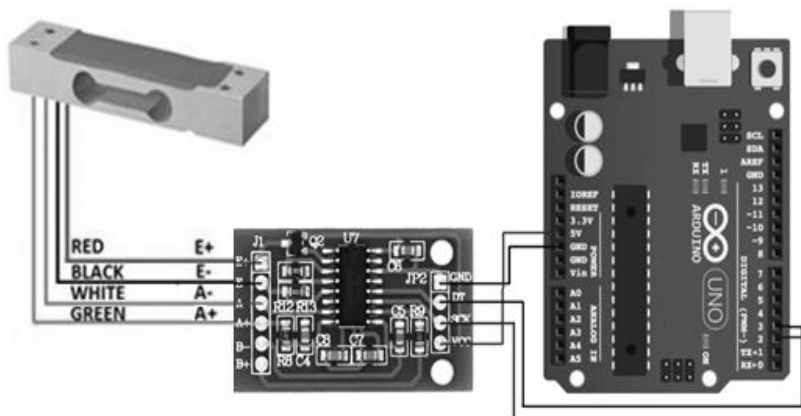


Рис. 7.1. Схема підключення тензOMETричного датчика до контролера Arduino за допомогою модуля зважування HX711

Для спрощення програмування є бібліотеки Arduino, наприклад, бібліотека HX711-master, що має в своєму складі функції вимірювання з можливістю усереднення декількох значень, а також функції тарування та калібрування.

Для передачі даних вимірювань в комп'ютер можна використовувати вбудовані інструменти IDE Arduino, а саме, "Монитор порта" ("Монітор порту") для виведення чисельних даних і "Плоттер по послідовному соединенію" ("Плоттер по послідовному з'єднанню") для виведення графіків в режимі реального часу.

Приклади представлення даних у вигляді послідовності чисел і графіка наведені на рис. 7.2.

Ось Y показує дані, які передаються по послідовному порту (USB). Ось X складається з 500 позицій, і кожна з яких послідовно виводиться значення, відповідне одному виконання команди Serial.println ().

Частота виведення даних, що вимірюються, на екран відповідає частоті виконання команди Serial.println ().

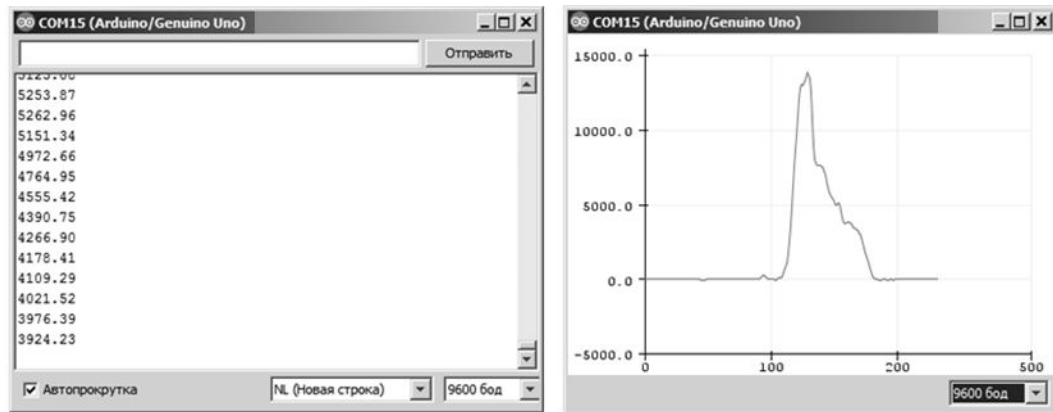


Рис. 7.2. Приклади представлення даних у вигляді послідовності чисел і графіка

Завдання до лабораторних занять

Ознайомитися з програмою виведення даних вимірювання ваги у вигляді графіка, що наведена у прикладі. Наведена програма виводить 500 значень з періодом 10 с.

Змінити програму для постійного виведення даних та подання сигналу на світлодіод у разі перевищення встановленого значення вантажу.

Приклад розв'язання завдань

програма виведення даних у вигляді графіка. При запуску програми здійснюється тарировка за допомогою функції scale.tare (), калібрування за допомогою функції scale.set_scale (calibration), а також установка коефіцієнта посилення 128 (діапазон ± 20 мВ) за допомогою функції scale.set_gain (128) або 64 (діапазон ± 40 мВ) за допомогою функції scale.set_gain (64). Зчитування даних здійснюється за допомогою функції scale.get_units ().

У програмі використовувалась бібліотека HX711 Arduino Library (HX711-master.zip).

sketch_HX711_2019_4_18

```
#include <HX711.h> // підключаємо бібліотеку HX711
HX711 scale;
float calibration = 3.7; // калібрувальний коефіцієнт
float units, gramm;
void setup() {
  Serial.begin(9600);
  scale.begin(3, 2);
  scale.set_scale();
  scale.tare(); // скидаємо ваги на нуль
  scale.set_scale(calibration); // застосовуємо калібрування
  scale.set_gain (128);
}
void loop() {
  for(int i = 0; i < 500; i++) { // 500 вимірювань
    units += scale.get_units(), 1;
```



```
gramm = units * 0.035274; // переводимо унції в грами
Serial.println(gramm);
}
//delay(10000); // період повторення виведення графіка 10 с
}
```

Контрольні запитання

ЛІТЕРАТУРА

1. Конспект лекцій з дисципліни «Сучасні тенденції створення автоматизованих систем управління вантажопідйомних машин» для здобувачів другого (магістерського) рівня вищої освіти, спеціальності: 133 - Галузеве машинобудування, освітня програма: – Підйомно-транспортні, будівельні дорожні машини і обладнання. / Укл.: Михайлов Є. П. – Одеса: ОП, 2021. - 88 с.

2. Попович М. Г., Ковальчук О. В. Теорія автоматичного керування: Підручник. — 2-е вид., перероб. і доп. — К.: Либідь, 2007. — 656 с. URL: http://pdf.lib.vntu.edu.ua/books/Popovich_2007_656.pdf

3. 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. – 171 с. Рег. ном. НП09356 19.03.18 №5682-РС-2018. URL: <http://dspace.opu.ua/jspui/handle/123456789/8143>

4. Blum J. Exploring Arduino: Tools and Techniques for Engineering Wizardry John Wiley & Sons, Inc., 2013. — 385 с. — ISBN: 978-1-118-54936-0. URL: <https://www.twirpx.com/file/1371963/>

ІНФОРМАЦІЙНІ РЕСУРСИ

5. Arduino. *Download the Arduino IDE* : веб-сайт. URL: <https://www.arduino.cc/en/Main/Software> (дата звернення: 20.08.2021).

6. Queen's University. *UnoArduSim*: веб-сайт. URL: <https://www.sites.google.com/site/unoardusim/home> (дата звернення: 20.08.2021).

7. Siemens AG. *LOGO! Web editor for the one in LOGO! integrated web server. Download a demo version of LOGO! Soft Comfort* : веб-сайт. URL: <http://w3.siemens.com/mcms/programmable-logic-controller/en/logic-module-logo/demo-software/pages/default.aspx> (дата звернення: 20.08.2021).