

УДК 004.42

## РОЗРОБКА КОМП'ЮТЕРНО-ІНТЕГРОВАНОЇ СИСТЕМИ УПРАВЛІННЯ РОБОТОМ-МАНІПУЛЯТОРОМ

Голубєв Л. П., Вовк Д. М.

Київський національний університет технологій та дизайну

**Мета.** Розробка комп'ютерно-інтегрованої системи керування роботизованим маніпулятором, що навчається.

**Методика.** Дослідження проводилися на системі керування рухом ланок робота-маніпулятора, побудованого на базі персонального комп'ютера (PC) і контролера управління рухом робочих вузлів робота.

**Результати.** Досліджено технологію роботи робота-маніпулятора і виконано розробку програмного і технічного забезпечення комп'ютерно-інтегрованої системи керування роботом-маніпулятором.

**Наукова новизна.** Розроблено новий підхід до створення комп'ютерно-інтегрованої системи керування роботом-маніпулятором, що навчається із застосуванням енергонезалежної пам'яті мікропроцесорної системи.

**Практична значимість.** Розроблений метод керування роботом-маніпулятором може бути застосований в багатьох роботизованих виробничих системах.

**Ключові слова:** робот-маніпулятор, Arduino, серводвигун, самонавчальна система

В даний час роботизація виробництва залишається основним напрямком, який забезпечує підвищення продуктивності праці, поліпшення якості продукції та зниження її собівартості [1, 2].

За способом навчання промислові роботи (ПР) можна розділити на три основні категорії:

- не перепрограмовані – промислові роботи працюють за заздалегідь підготовленою досить простій програмі, що повторює одну й ту ж задану послідовність операцій;
- жорстко програмовані (перенавчальні) промислові роботи із змінним циклом операцій, які містять повний набір інформації, що не змінюється в процесі самої роботи, але піддається коригуванню шляхом «перенавчання» при зміні технологічного процесу;
- гнучко програмовані (самонавчальні) промислові роботи з елементами штучного інтелекту здатні формувати склад і послідовність своїх дій на основі поставленої мети та інформації про навколишнє середовище в умовах неорганізованого робочого простору.

Звичайно ж, найкращим варіантом є використання гнучко програмованих промислових роботів, з елементами штучного інтелекту, які мають розвинену сенсорну систему у вигляді штучних органів зору, слуху і дотику. Однак істотним недоліком цих пристроїв є їх висока вартість.

На підприємствах на сьогоднішній день використовується велика кількість ПР, які непрограмовані або жорстко програмовані. Тому задача розробки автоматизованої системи навчання та управління роботою роботами-маніпуляторами перерахованих вище типів є дуже актуальною.

### **Постановка завдання**

Незважаючи на велике різноманіття систем роботів-маніпуляторів, що застосовуються на виробництві, ми досліджували найбільш універсальну 6-ступеневу систему робота-маніпулятора з електромеханічним захватом [3, 4].

В результаті роботи був створений працюючий макет робота-маніпулятора, проведені дослідження особливостей його роботи і виявлені наступні завдання:

- 1) розробка мікропроцесорної системи управління рухом ланок робота-маніпулятора;
- 2) розробка інтерфейсу роботи оператора з навчання робота-маніпулятора;
- 3) забезпечення роботи системи в ручному режимі навчання;
- 4) забезпечення роботи системи в автоматичному режимі (по записаній в енергонезалежній пам'яті програми).

### **Результати досліджень**

На рис. 1 приведена кінематична схема робота-маніпулятора. Ланки 1 і 5 виконують обертальні рухи, а ланки 2, 3, 4 – поступальні. Ланка 6 – електромеханічний захват.

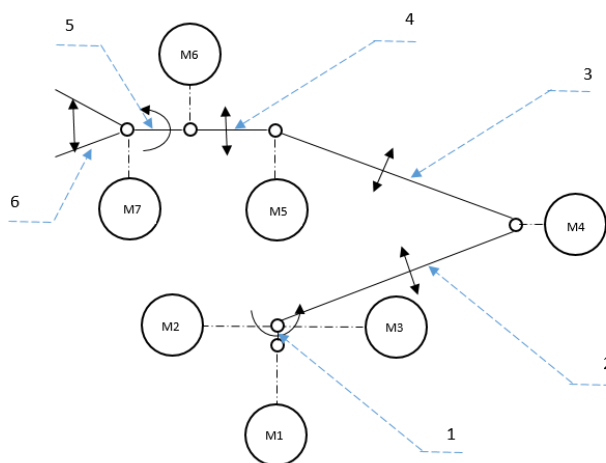


Рис. 1. Кінематична схема робота-маніпулятора

На кінематичній схемі серводвигуни позначені М1-М7. Завдяки обертовим ступенями свободи (ланки 1 і 5) робот може здійснювати обертання вліво і вправо всього механізму, а також і захвату. Для додання лінійного переміщення уздовж поступальних ступенів свободи використовуються ланки 2-4. У кожному з приводів використані серводвигуни MG996R.

Зовнішній вигляд робота-маніпулятора наведено на рис. 2.



Рис. 2. Робот-маніпулятор з 6-ма ступенями свободи

Приводи ланок маніпулятора жорстко посаджені на вали відповідних серводвигунів, що забезпечує точність і надійність з'єднання. Для збільшення зусилля по повороту платформи двигуни М2 і М3 працюють синхронно.

Структурна схема автоматизованої системи управління роботом-маніпулятором що навчається, представлена на рис. 3.

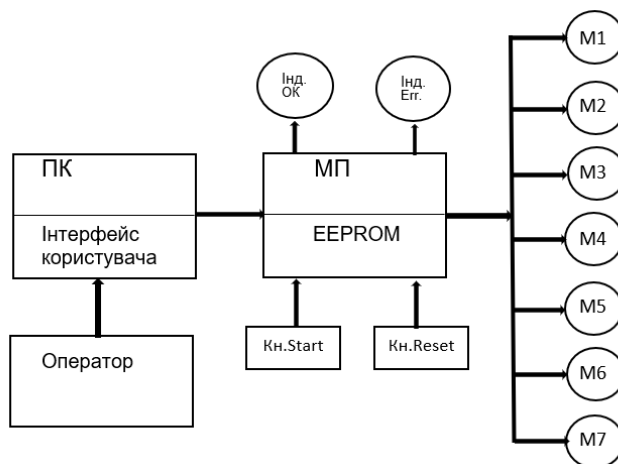


Рис. 3. Структурна схема системи управління роботом-маніпулятором

Основні елементи системи, представлені на схемі:

ПК – персональний комп'ютер з інтерфейсом користувача для управління роботом-маніпулятором;

МП – мікропроцесорна система, що забезпечує управління рухом ланками маніпулятора з енергонезалежною пам'яттю;

М1-М7 – серводвигуни, що забезпечують обертальні і поступальні рухи ланок маніпулятора;

Кн. «Start» – кнопка запуску програми, записаної в енергонезалежній пам'яті;

Кн. «Reset» – кнопка установки ланок маніпулятора в початкове положення;

Інд. «Ок» – індикатор, що підтверджує наявність записаної програми в енергонезалежній пам'яті;

Інд. «Егг» – індикатор, що повідомляє про помилку програми в енергонезалежній пам'яті.

Розроблений інтерфейс користувача представлений рис. 4.

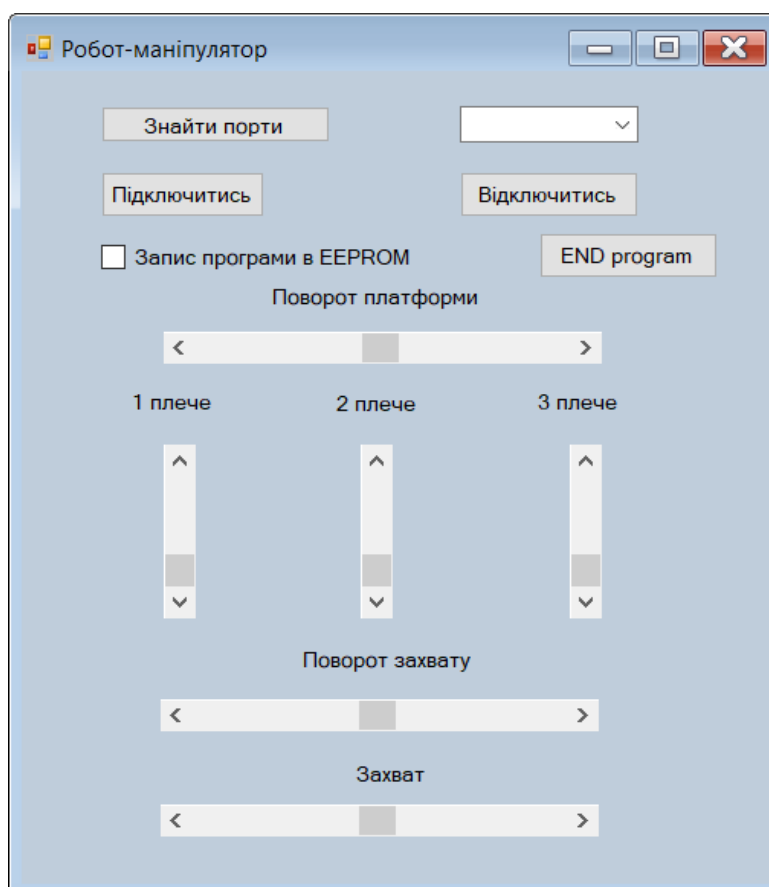


Рис. 4. Інтерфейс програми управління роботою робота-маніпулятора

Послідовність дій оператора при роботі з системою.

- 1) Вибрати вільний СОМ порт.
- 2) Підключитися до вибраного порта. Якщо підключення не вдалося – система видає відповідне повідомлення.
- 3) Виконати програмне скидання системи (встановити в початкове положення ланки робота-маніпулятора – функція RESET ()).
- 4) Включити режим запису команд оператора в область енергонезалежної пам'яті мікропроцесорної системи.
- 5) Виконати послідовність команд по переміщенню ланок робота в потрібне положення.
- 6) Домогтися правильної послідовності дій.
- 7) Відключити від комп'ютера (СОМ-порта).
- 8) Активувати режим автоматичного виконання записаної в EEPROM програми (Натискання кнопки «Start»).
- 9) Після відпрацювання програми скинути систему (Натиснути кнопку «Reset»).

Для забезпечення роботи серводвигунів використовувалася бібліотека <Servo.h>.

Для роботи серводвигунів в системі створено 7 об'єктів типу Servo:

```
Servo myservo1; // Поворот платформи
Servo myservo2_1; // 2 плече поворот
Servo myservo2_2; // 2 плече поворот
Servo myservo3; // 3 плече
Servo myservo4; // 4 плече
Servo myservo5; // 5 поворот зажима
Servo myservo6; // 6 зажим
```

Підключення до контактів Arduino Uno здійснюється за допомогою команд:

```
myservo1.attach(9);
myservo2_1.attach(10);
myservo2_2.attach(11);
myservo3.attach(12);
myservo4.attach(13);
myservo5.attach(4);
myservo6.attach(5);
```

Для роботи з енергонезалежною пам'яттю Arduino використовувалася бібліотека EEPROM.h. Вона надає доступ до основних функцій по роботі з EEPROM.

Функція	Опис
<b>read()</b>	<b>EEPROM.read(<i>address</i>)</b> Зчитує один байт з EEPROM за адресою <i>address</i> . Значення комірки, за замовчуванням, 255. <i>address</i> – адреса комірки що зчитується (int), починається з 0.
<b>write()</b>	<b>EEPROM.write(<i>address, value</i>)</b> Записує один байт із значенням <i>value</i> в EEPROM за адресою <i>address</i> . <i>address</i> – адреса комірки для запису (int), починається з 0. <i>value</i> – значення байта що записується (byte), від 0 до 255.
<b>update()</b>	<b>EEPROM.update(<i>address, value</i>)</b> Аналог функції <i>write</i> ( ), за тим лише винятком, що значення записується тільки тоді, коли воно відрізняється від уже записаного. Оскільки EEPROM має життєвим циклом в 100 000 операцій запису, використання даної функції краще і дозволяє продовжити життя EEPROM.
<b>get()</b>	<b>EEPROM.get(<i>address, data</i>)</b> Зчитує з EEPROM будь-який тип даних за адресою <i>address</i> . <i>address</i> – адреса комірки для запису (int), починається з 0. <i>data</i> – дані для читання, можуть бути будь-яким базовим типом (наприклад, float), або користувальницької структурою <i>struct</i> .
<b>put()</b>	<b>EEPROM.put(<i>address, data</i>)</b> Записує в EEPROM будь-який тип даних за адресою <i>address</i> . Оскільки функція реалізована на функції <i>update</i> ( ), то і працює вона схожим чином – записує тільки відрізняються дані. <i>address</i> – адреса комірки для запису (int), починається з 0. <i>data</i> – дані для запису, можуть бути будь-яким базовим типом (наприклад, float), або користувальницької структурою <i>struct</i> .

На початку роботи виконується навчання маніпулятора. Для цього оператор за допомогою розробленого інтерфейсу користувача виконує дії з ланками маніпулятора, необхідні для виконання технологічного процесу.

Якщо при роботі системи активний CheckBox «Запис програми в EEPROM», то всі команди оператора будуть записані в енергонезалежну пам'ять Arduino. Після запису програми в енергонезалежну пам'ять систему можна відключити від персонального комп'ютера і працювати далі автономно.

Далі, після натискання кнопки «Start», система перевіряє наявність коректної програми в енергонезалежній пам'яті. Якщо така є загоряється індикатор «Ок» і починається автономне виконання програми, записаної в EEPROM. Якщо коректної програми не виявлено загоряється індикатор «Error» і система переходить в стан очікування. У цьому випадку оператор повинен повторити виконання режиму навчання ще раз.

**Висновки**

За допомогою розробленого програмного інтерфейсу оператор, під'єднавшись до мікропроцесорної системи управління роботом, керує положенням ланок робота і виконує захват деталі. При цьому команди оператора записуються в EEPROM. Таким чином оператор виконує процедуру навчання робота і домагається правильної його роботи.

Розроблена система є досить універсальною, тобто може легко здійснювати перехід до випуску нової моделі або виконувати модернізацію поточної, для цього всього-на-всього необхідно виконати навчання автоматизованої системи відповідним операціям. Після успішного навчання робот може виконувати записані в програму дії потрібну кількість разів.

До недоліків розробленої системи можна віднести обмежену кількість команд програми, яке визначається обсягом енергонезалежної пам'яті (для мікропроцесора AtMega 328 – 1024 байт).

**Список використаних джерел**

1. Юревич Е. И. Основы робототехники. – 2-е изд., перераб. и доп. / Е. И. Юревич – СПб.: БХВ – Петербург, 2005. – 416 с.
2. Никитин К. Д. Основы робототехники / К. Д. Никитин, Н. В. Василенко, В. П. Пономарёв, А. Ю. Смолин – ТОМСК МГП «РАСКО» 2013. – 238 с.
3. Можчиль Б. В. Использование микропроцессоров при создании автоматизированных систем управления [Электронный ресурс] / Б. В. Можчиль, С. Ю. Фетисенко, Л. П. Голубев. // Технології та дизайн. – 2016. – № 3. – Режим доступу: [http://nbuv.gov.ua/UJRN/td\\_2016\\_3\\_6](http://nbuv.gov.ua/UJRN/td_2016_3_6)
4. Столяров В. Г. Автоматизированное удаленное управление устройствами при помощи Ардуино [Электронный ресурс] / В. Г. Столяров, Л. П. Голубев. // Технології та дизайн. – 2016. – № 4. – Режим доступу: [http://nbuv.gov.ua/UJRN/td\\_2016\\_4\\_12](http://nbuv.gov.ua/UJRN/td_2016_4_12)

**References**

1. Yurevich, Ye.I. (2005). *Osnovy robototekhniki. – 2-e izd., pererab. i dop. [Basics of robotics]*. SPb. BKhV – Peterburg [in Russian].
2. Nikitin, K.D., Vasilenko, N.V., Ponomarev, V.P. & Smolin, A.Yu. (2013). *Osnovy robototekhniki [Basics of robotics] – TOMSK. MGP «RASKO»* [in Russian].
3. Mozhchil, B.V., Fetisenko, E.Yu. & Golubev, L.P. (2016). *Ispolzovanie mikroprotsessorov pri sozdanii avtomatizirovannykh sistem upravleniya [The use of microprocessors in the creation of automated control systems]. Tekhnologii ta dizayn - Technology and design* – Retrieved from: [http://nbuv.gov.ua/UJRN/td\\_2016\\_3\\_6](http://nbuv.gov.ua/UJRN/td_2016_3_6) [in Ukrainian].
4. Stolyarov, V.G. & Golubev, L.P. (2016). *Avtomatizirovannoe udalennoe upravlenie ustroystvami pri pomoshchi Arduino [Automated remote device control with Arduino] Tekhnologii ta dizayn - Technology and design*. Retrieved from:

[http://nbuv.gov.ua/UJRN/td\\_2016\\_4\\_12](http://nbuv.gov.ua/UJRN/td_2016_4_12)  
[in Ukrainian].

**Golubev Leontiy**

ORCID: <https://orcid.org/0000-0002-2980-8017>

[golubevl@ukr.net](mailto:golubevl@ukr.net)

Kyiv National University of  
Technologies and Design

**Vovk Dmytro**

[usr79@ukr.net](mailto:usr79@ukr.net)

Kyiv National University of  
Technologies and Design

***Разработка компьютерно-интегрированной системы управления роботом-манипулятором***

***Голубев Л. П., Вовк Д. Н.***

*Киевский национальный университет технологий и дизайна*

***Цель.*** Разработка компьютерно-интегрированной обучаемой системы управления роботизированным манипулятором.

***Методика.*** Исследования проводились на системе управления движением звеньев робота-манипулятора, построенного на базе персонального компьютера (ПК) и контроллера управления движением рабочих узлов робота.

***Результаты.*** Выполнено исследование технологии работы робота-манипулятора и разработано программное и техническое обеспечение компьютерно-интегрированной системы управления роботом-манипулятором.

***Научная новизна.*** Разработан новый подход к созданию обучающейся системы управления роботом-манипулятором с применением энергонезависимой памяти микропроцессорной системы.

***Практическая значимость.*** Разработанный метод управления роботом-манипулятором может быть применен во многих роботизированных производственных системах.

***Ключевые слова:*** робот-манипулятор, Arduino, серводвигатель, самообучающаяся система

***Development of a computer-integrated control system for a robot manipulator***

***Golubev L. P., Vovk D. M.***

*Kiev National University of Technology and Design*

***Purpose.*** Development of a computer-integrated learning control system for a robotic manipulator.

***Methodology.*** The research was carried out on the motion control system of the robot-manipulator units constructed on the basis of a personal computer (PC) and a workload control unit controller.

***Findings.*** The research of the robot-manipulator technology and the development of software and technical support for the robot-manipulator control system was performed.

***Originality.*** A new robot-manipulator control technology is developed. A new approach to creating a self-learning robot-manipulator control system is developed using a non-volatile memory of a microprocessor system..

***Practical value.*** The developed robot manipulator control method can be applied in many robotic production systems.

***Keywords:*** robot-manipulator, Arduino, servomotor, self-learning system