

О.М. Поляков, Б.Ю. Жураковський

ПРОТОТИПУВАННЯ ПРИСТРОЇВ КЕРУВАННЯ СИСТЕМ З ПРОМИСЛОВИМИ КОНТРОЛЕРАМИ

Анотація. Скорочення термінів проектування пристрою керування системою залишається актуальним завданням розробників цих систем. Проблемою проектування пристроїв керування на основі програмованих логічних контролерів (ПЛК) є їхня висока вартість і, як правило, недоступність на початковому етапі проектування. Метою дослідження є скорочення термінів та зниження витрат на проектування системи шляхом прототипування пристроїв керування з програмною реалізацією алгоритмів керування мовами стандарту МЕК 61131 3, але виконанням програм у платі Ардуїно. Метод дослідження полягає в декомпозиції проектних моделей пристрою управління та реалізації їх у середовищі програми OpenPLC у вигляді компонентів організації програми (POU) мовами SFC, FBD та LD. Результатом дослідження є методика створення типових POU операційних та керуючих автоматів системи керування, які виконуються у платі Ардуїно. Наведено приклад застосування запропонованої методики проектування прототипу, який може бути корисним для навчання програмуванню ПЛК. Проведено тестування розробленого прототипу за допомогою логічного ПЛК та фізичного макета, яке підтвердило їхню функціональну відповідність оригіналу та зниження, як мінімум на порядок витрат на обладнання.

Ключові слова: програмовані логічні контролери, мови програмування контролерів, прототипування систем управління.

Постановка проблеми. Для керування обладнанням та виробничими процесами широко та ефективно застосовуються промислові контролери [1-3]. Ці контролери ще називають програмованими логічними контролерами (ПЛК).

Слід зазначити, що ПЛК це дороге обладнання. Тому на ранніх стадіях проектування системи доцільно застосування прототипів ПЛК – функціональних аналогів реального ПЛК на базі більш дешевих мікроконтролерів, наприклад плат Ардуїно [4]. Але для програмування ПЛК використовуються спеціальні мови за стандартом МЕК 61131-3 [5], а найбільш поширені середовища програмування, наприклад Arduino IDE [4] підтримують мову програмування C і тому не застосовні при прототипуванні завдань керування систем з ПЛК.

Відомі програмні продукти для трансформації програм, написаних мовами стандарту MEK 61131-3 в код виконаний у мікроконтролері плати Ардуїно. Наприклад, середовище програмування OpenPLC, що вільно розповсюджується [6]. Компонент цього середовища OpenPLC Editor призначений для введення, редагування, компіляції програмного коду мовами стандарту MEK 61131-3 та симуляції виконання цього коду в логічному ПЛК. Компонент OpenPLC Runtime транслює код, створений в OpenPLC Editor у здійснений код і завантажує його в реальний ПЛК, який підключений до комп'ютера через USB-порт.

Водночас використання OpenPLC для прототипування систем керування утруднене через відсутність методики трансформації проектних моделей системи керування в компоненти організації програм (POU – Program Organization Unit, англ. мова) у цьому середовищі. Таким чином, створення методики прототипування систем керування на основі ПЛК за допомогою мікроконтролерних плат типу Ардуїно є актуальним не вирішеним науково-технічним завданням.

Аналіз останніх досліджень і публікацій. Найбільш узагальнена модель системи керування на теоретико-множинному рівні визначає виділення об'єкта керування та керуючого пристрою [7]. Деталізація моделі керуючого пристрою залежить від виду системи керування (безперервна, дискретна, гібридна), розміщення елементів системи в просторі (зосереджена, розподілена), наявності ієрархії елементів, що керують, паралельності в часі різних процесів керування та інших факторів [8]-[10].

Об'єкт досліджень: процес прототипування систем керування з на базі ПЛК. У роботі розгляд питань прототипування обмежено дискретними зосередженими системами з програмною реалізацією алгоритму управління. Для таких систем у керуючому пристрої виділяють операційні (ОА) та керуючі (КА) автомати [7]. Моделлю керуючого автомата є кінцевий автомат (FSM – Finite State Machine) типу перетворювач. Класичний FSM описується кортежем виду [7]:

$$A = \langle S, X, Y, s_0, \delta, \lambda \rangle,$$
$$\delta: S \times X \rightarrow S,$$
$$\lambda: S \times X \rightarrow Y \text{ (для автоматів Мілі),}$$
$$\lambda: S \rightarrow Y \text{ (для автоматів Мура),}$$

де S – скінченна непорожня множина (станів); X – скінченна непорожня множина входів (вхідний алфавіт); Y – скінченна непорожня множина виходів (ви-

хідний алфавіт); s_0 — початковий стан; δ — функція переходів; λ — функції виходів автоматів Мілі та Мура.

Розрізняють вхідні (ВхОА), вихідні (ВихОА) операційні автомати. Структурну схему системи керування наведено на рис.1.

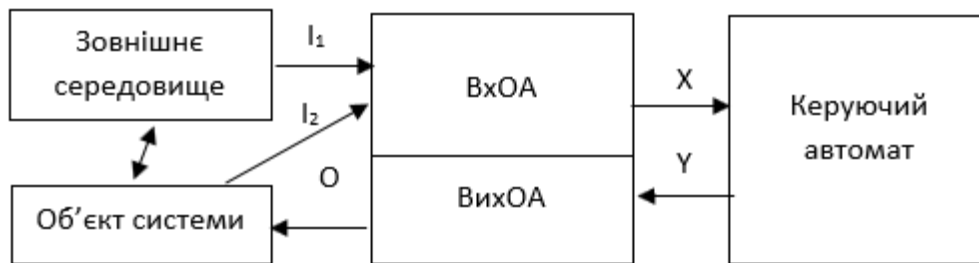


Рисунок 1 – Структура системи керування

Вхідний операційний автомат формує вектор входів X автомата, що управляє, як функцію інформації від об'єкта (I_1) і зовнішнього середовища (I_2) системи. Вихідний операційний автомат формує вектор O впливів на об'єкт системи як функцію виходів Y керуючого автомата. Описана декомпозиція елементів пристрою керування системи спрощує їхню програмну реалізацію в процесі прототипування.

Наведені теоретико-множинні моделі елементів керування є основою для побудови програмно-апаратних моделей прототипу базової системи керування. У складніших системах ці моделі відображають можливості параметричної та структурної адаптації елементів керуючого пристрою, небінарні властивості множин X , Y . І це також вимагає відображення в методиці прототипування.

Мета роботи: скорочення термінів та зниження витрат на проектування системи шляхом прототипування пристроїв керування з програмною реалізацією алгоритмів керування мовами стандарту МЕК 61131-3, але виконанням програм у платі Ардуїно.

Основний матеріал дослідження. Під прототипом зосередженої системи управління на основі ПЛК розумітимемо її функціональний аналог, в якому ПЛК заміщений спрощеною мікроконтролерною платою, при цьому:

- Вихідний код керуючої програми, яка виконується в мікроконтролері плати, написаний мовами програмування за стандартом МЕК 61131-3 та функціонально повторює програмний код системи оригіналу.

- До входів та виходів мікроконтролерної плати підключено всі датчики та виконавчі механізми об'єкта керування, які підключені до ПЛК у реальній сис-

темі. У прототипі системи сигнали від датчиків та до виконавчих механізмів узгоджені за рівнями та полярністю з каналами мікроконтролерної плати.

- Є таблиця відповідності входів/виходів ПЛК, каналів мікроконтролерної плати та їх адреси у програмі.

- Тимчасові константи у програмі мікроконтролерної плати залежні від тактової частоти процесора узгоджені з константами керуючої програми ПЛК.

- Зв'язку мікроконтролерної плати з персональним комп'ютером через віртуальний СОМ-порт достатньо для перевірки функціональності системи за допомогою її прототипу.

У разі невиконання хоча б однієї з перелічених вимог говоритимемо про не повну функціональну відповідність прототипу оригіналу. Але, у багатьох випадках це також корисно, наприклад при навчанні програмування ПЛК.

Можливо два варіанти використання прототипування системи керування:

- Спроектвано систему керування на базі ПЛК, але фізично не доступний обраний ПЛК. І тут прототипування спрощує перевірку правильності алгоритму керування реальним об'єктом чи його макетом.

- Спроектований та протестований прототип системи керування на базі мікроконтролерної плати та макета об'єкта керування. Ця робота спрощує вибір ПЛК.

Вихідними даними для прототипування системи керування є електрична схема системи та моделі операційних та керуючого автоматів, які треба реалізувати у програмно-апаратному комплексі прототипу.

На першому етапі проводиться вибір мікроконтролерної плати з урахуванням необхідної кількості цифрових та аналогових каналів введення/виводу, каналів з ШІМ, оцінюється потреба у додаткових вузлах, які відсутні у платі прототипу. Це можуть бути, наприклад, вузли узгодження входів/виходів прототипу за рівнем та потужністю, додаткові джерела живлення. Багато ПЛК працюють у діапазоні напруги 24 В, а мікроконтролерні плати – у діапазоні 5 В.

На другому етапі проводиться вибір структури та елементів РОУ для проекту прототипу керуючої програми. Програма є типом РОУ верхньому рівні організації проекту прототипу. У проекті може бути кілька програм. Операційні автомати реалізуються як функції чи функціональні блоки, а управляючі – як функціональні блоки. Для зв'язку компонентів програми з апаратною частиною мікроконтролерної плати використовуються змінні класів вхід та вихід. Функціональні блоки у програмі описуються як локальні класи.

На третьому етапі обираються мови програмування для POU проекту прототипу програми, що управляє. Різні мови стандарту МЕК 61131-3 мають різні ефективності при реалізації компонентів проекту прототипу системи керування. При виборі мови слід перевірити її доступність у планованому ПЛК. Якщо є можливість вибору, то реалізації ОА вибираються ефективні мови ST, FBD, рідше – мова IL. У той же час, КА простіше реалізувати мовою SFC, але можливо й іншими мовами. Таким чином, типова керуюча програма для прототипу системи є мультимовною.

Приклад прототипування. Розглянемо найпростіший приклад дискретної системи стабілізації температури об'єкта. Структура системи відповідає моделі, наведеній на рис. 1. У цьому випадку об'єкт системи містить датчик температури та нагрівач. За відсутності нагрівання об'єкт остигає, передаючи теплову енергію у зовнішнє середовище. Зовнішнє середовище формує завдання I_1 на температуру об'єкта. Операційний автомат ВхОА порівнює фактичну I_2 і задану I_1 температури та формує події (входи X): «температура менша за завдання», «температура більша за завдання». Ці входи визначають поточний стан керуючого автомата («нагрів» або «охолодження»), в якому формується вихід Y . Операційний автомат ВихОА перетворює цей вихід сигнал керування нагрівачем.

У цій системі як ПЛК, так і мікроконтролерна плата повинні мати по одному аналоговому входу (температура) та по два дискретні виходи (ввімкнення/вимкнення нагрівача). Цим вимогам задовольняють контролер Micrologix 1000 [3] та плата Arduino Uno [4]. Схема підключення датчика та виконавчих механізмів до цієї плати наведена на рис.2.

Датчик приєднано до аналогового входу А0. Двійковий код температури надано змінній *temp* (адреса %IW0). Управління нагрівачем виконується через цифрові виходи 7 і 8. Управляюча дія «включити нагрівання» виконується змінною *outon* (вихід 7, адреса %QX0.0), а вплив «вимкнути нагрівання» – змінною *outoff* (вихід 8, адреса %QX0.1).

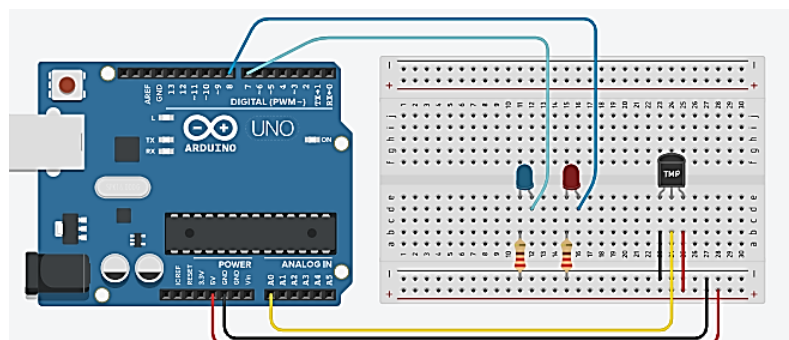


Рисунок 2 – Схема підключення об'єкта до мікроконтролерної плати прототипу

Елементи POU проекту прототипу системи керування у середовищі програми OpenPLC Editor наведено на рис. 3.

Проект *Prototype* включає функціональні блоки OA і CA, які виконуються в програмі *LimTemp*. Ця програма реалізує алгоритм керування нагрівом об'єкта системи за якого нагрівання відбувається за умови, що температура об'єкта менша за граничну. У проекті *Prototype* значення коду температури прийнято рівним 40.

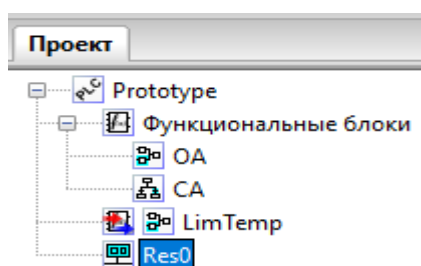


Рисунок 3 – Елементи POU проекту прототипу системи керування у середовищі програми OpenPLC Editor

Змінні та логіка функціонального блоку OA наведені на рис. 4. Цей блок формує входи *outheat* і *outcold* керуючого автомата CA, тобто $X = \{outheat, outcold\}$.

Функціональний блок CA реалізує керуючий автомат обмеження нагріву об'єкта, граф якого представлений на рис. 5. Він містить два стани та два переходи. Початковий стан автомата – *Cooling*. У цьому стані виконується дія *outoff* – вимкнути нагрівання об'єкта.

#	Имя	Класс	Тип	Исходное значение
1	temperature	Вход	DINT	
2	outheat	Выход	BOOL	
3	outcold	Выход	BOOL	

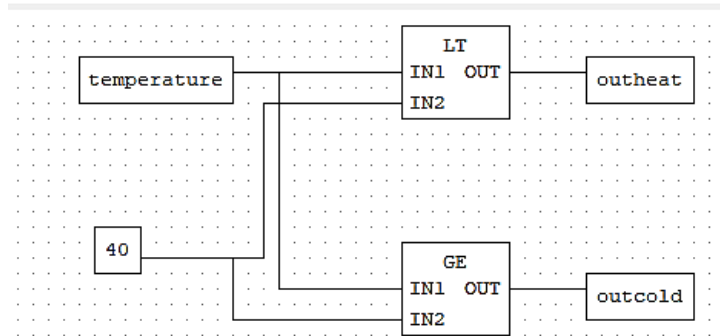


Рисунок 4 – Графічне уявлення функціонального блоку ОА

Графічний варіант реалізації функціонального блоку КА мовою SFC наведено на рис. 6. Описані вище функціональні блоки сприймаються програмою *LimTemp* як користувальницькі ROU. Графічний варіант цієї програми на мові FBD наведено на рис. 7.

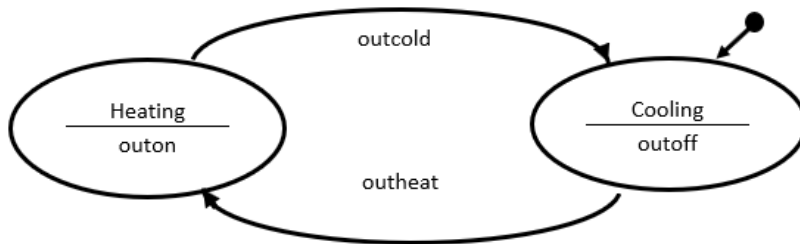


Рисунок 5 – Граф автомата обмеження нагріву

#	Имя	Класс	Тип	Исходное значение	Квалификатор
1	heat	Вход	BOOL		
2	cold	Вход	BOOL		
3	outon	Выход	BOOL		
4	outoff	Выход	BOOL		

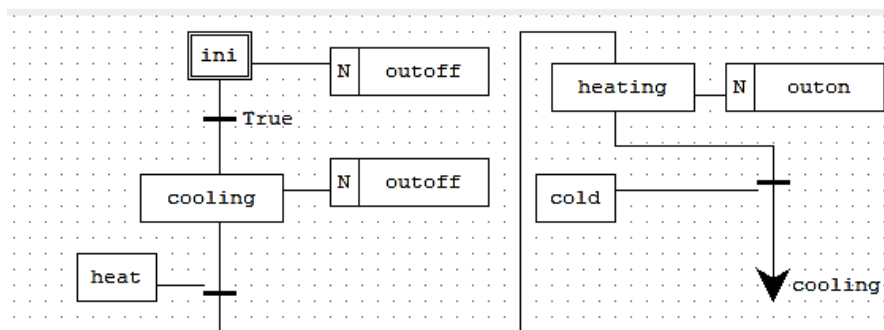


Рисунок 6 – Графічне представлення функціонального блоку КА мовою SFC

«Системні технології» 2 (151) 2024 «System technologies»

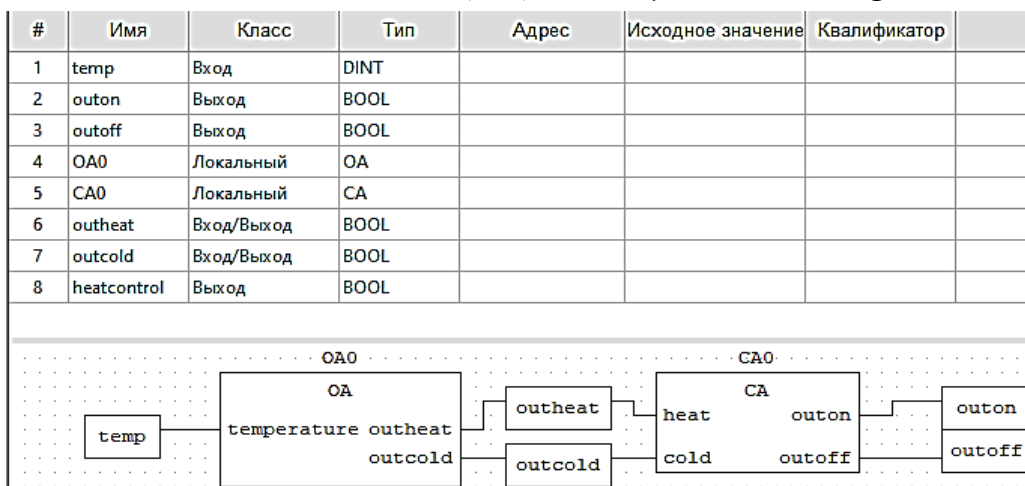


Рисунок 7 – Графічне представлення програми *LimTemp* мовою FBD

Функціональний блок КА був спроектований мовою LD. Програму цього блоку наведено на рис. 8. Вона містить 6 рангів.

#	Имя	Класс	Тип	Исходное значение	Квалификатор	Описание
1	firstscan	Локальный	BOOL	1		
2	heat	Вход	BOOL	0		
3	cold	Вход	BOOL	0		
4	outon	Выход	BOOL	0		
5	outoff	Выход	BOOL	0		
6	cooling	Локальный	BOOL	0		

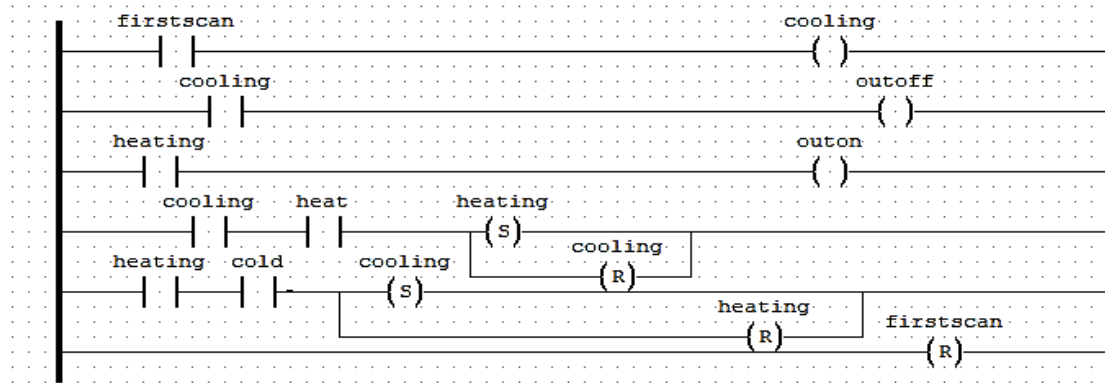


Рисунок 8 – Графічне представлення функціонального блоку КА мовою LD

Перший та останній ранги забезпечують перехід у стан *cooling* при ініціалізації. Другий та третій ранги визначають дії (виходи автомата) в активному стані, а третій та четвертий – переходи з одного стану до іншого.

Експерименти Перевірка проекту Prototype у середовищі програми OpenPLC Editor з використанням логічного ПЛК для обох версій автомата КА підтвердила відповідність проекту вимогам системи керування. Бажане зна-

чення змінної *temp* встановлювалося у налагоджувачі у вікні "форсувати значення змінної".

Перевірка проекту *Prototype* із виконанням програмного коду в платі Ардуїно, до якої підключено за схемою рис. 2 прототипи датчика температури та нагрівача проводилася з використання програм OpenPLC Editor та OpenPLC Runtime. Фактична температура задавалася шляхом нагрівання датчика. При цьому сигнали керування нагріванням індикували світлодіодами і відповідали вимогам системи керування.

Оцінимо зниження витрат під час використання прототипу керуючого пристрою системи керування. Прототип - плату Arduino Uno, на момент написання статті, можна було придбати приблизно за 10 у. о. У той час як промисловий контролер з аналогічним набором входів/виходів оцінюється від 300 до 500 у.о. Наприклад, ПЛК Micrologix 1000 1761-L20BWB-5A (компанія Rockwell Automation [3]) має 20 дискретних входів/виходів та 4 аналогові входи. Його можна придбати в Україні за 480 євро.

Висновки. Основні проблеми проектування та навчання проектуванню систем керування на базі ПЛК – висока вартість та недоступність ПЛК на початковому етапі проектування.

Запропонована методика прототипування дозволяє трансформувати операційні та керуючі автоматів проектних моделей системи керування на компоненти організації програм проекту прототипу системи.

Додаток OpenPLC дозволяє ефективно розробляти та тестувати проекти прототипів у мультимовному середовищі сімейства мов програмування ПЛК за стандартом MEK 61131-3.

Приклад прототипування системи обмеження нагріву виконано з використанням доступних мікроконтролерних плат Ардуїно та реалізації операційних та керуючих автоматів графічними мовами LD, SFC, FBD. Проведене тестування розробленого прототипу за допомогою логічного ПЛК та фізичного макета підтвердило його функціональну відповідність оригіналу та зниження, як мінімум, на порядок витрат на обладнання.

Подальше дослідження передбачається проводити у бік прототипування адаптивних ієрархічних систем керування.

ЛІТЕРАТУРА

1. Parr, E. A. Programmable Controllers. An engineer's guide / E. A. Parr. 3rd ed. – Oxford : Newnes, 2003. – 429 p.
2. Віддалений та віртуальний інструментарій в інжинірингу : монографія / за заг. ред. Хенке К. – Запоріжжя: Дике поле, 2015. – 250 с.
3. 1761-UM003B-EN-P MicroLogix 1000 Programmable Controllers User Manual [Electronic resource] – Access mode:
https://literature.rockwellautomation.com/idc/groups/literature/documents/um/1761-um003_-en-p.pdf
4. Arduino - Home [Electronic resource] – Access mode: <https://www.arduino.cc/>.
5. IEC 61131-3, Revision 3.0, February 2013 - Programmable controllers – Part 3: Programming languages. Published By: International Electrotechnical Commission (IEC). – 468 p.
6. OpenPLC Overview – Autonomy. [Electronic resource] – Access mode: <https://autonomylogic.com/docs/openplc-overview/>.
7. Глушков В. М. Синтез цифрових автоматів/В. М. Глушков. – М. Фізматіздат, 1962. - 456 с.
8. Поляков М. А. Комплекс математичних моделей функціональних елементів та структур інтегрованих та когнітивних систем /М. А. Поляков // Матеріали міжнародної науково-технічної конференції «Інформаційні технології в металургії та машинобудуванні» імені професора Михальова А.В. І. (НМетАУ, 17 – 19 березня 2020 року) . – Дніпро, 2020. – с. 228–233.
9. Behavioral Types in Programming Languages Foundations and Trends in Programming Languages / [D. Ancona et al.]. – 2016. – Vol. 3. – No. 2–3. – p. 95–230.
10. Poliakov, O. Performance indicators of models of non-binary control automates / M. Poliakov, S. Subbotin, O. Poliakov // Proceeding of 2021 IEEE 16th International Conference on the Experience of Designing and Application of CAD Systems (CADSM) (22-26 February, 2021 Lviv, Ukraine). – P. 38–42.

REFERENSIS

1. Parr, E. A. Programmable Controllers. An engineer's guide / E. A. Parr. 3rd ed. – Oxford: Newnes, 2003. – 429 p.
2. Viddalenyu ta virtual'nyy instrumentariy v inzhynirynhu: monohrafiya / za zah. ed. Khenke K. - Zaporizhzhya: Dyke pole, 2015. - 250 с.
3. 1761-UM003B-EN-P MicroLogix 1000 Programmable Controllers User Manual [Electronic resource]– Access mode:

https://literature.rockwellautomation.com/idc/groups/literature/documents/um/1761-um003_-en-p.pdf

4. Arduino - Home [Electronic resource]– Access mode: <https://www.arduino.cc/>
5. IEC 61131-3, Revision 3.0, February 2013 - Programmable controllers – Part 3: Programming languages. Published By: International Electrotechnical Commission (IEC). – 468 p.
6. OpenPLC Overview – Autonomy (autonomylogic.com) [Electronic resource]– Access mode: <https://autonomylogic.com/docs/openplc-overview/>
7. Hlushkov V. M. Syntez tsyfrovyykh avtomativ/V. M. Hlushkov. - M. Fizmatizdat, 1962. -456 p.
8. Polyakov, M. O. Kompleks matematychnykh modeley funktsional'nykh elementiv ta struktur intehrovanykh ta kohnityvnykh system / M. O. Polyakov // Materialy mizhnarodnoyi naukovo-tekhnichnoyi konferentsiyi "Informatsiyeni tekhnolohiyi u metalurhiyi ta mashynobuduvanni" imeni profesora Mikhal'ova O. I. (NMetAU, 17-19 bereznya 2020). - Dnipro, 2020. - S. 228-233.
9. Behavioral Types in Programming Languages Foundations and Trends R in Programming Languages / [D. Ancona et al.]. – 2016. – Vol. 3. – No. 2–3. – P. 95–230.
10. Poliakov, O. Performance indicators of models of non-binary control automates / M. Poliakov, S. Subbotin, O. Poliakov // Proceeding of 2021 IEEE 16th International Conference on the Experience of Designing and Application of CAD Systems (CADSM) (22-26 February, 2021 Lviv, Ukraine). – P. 38–42.

Received 04.03.2024.
Accepted 06.03.2024.

Prototyping of control units for systems with industrial controllers

Reducing the design time of the system control unit remains an urgent task for the developers of these systems. The problem of designing control units based on programmable logic controllers (PLCs) is their high cost and, as a rule, unavailability at the initial design stage. The aim of the research is to reduce the time and cost of designing the system by creating prototypes of control units with the software implementation of the control algorithms of the languages of the IEC 61131-3 standard and the execution of programs in the Arduino board. The research method consists in the decomposition of project models of operating and control automata of the control device and their implementation in the OpenPLC application environment in the form of program organization components (POU) in Ladder Diagram, Function Block Diagram and Sequential Function Chart languages. The result of the study is a method of creating typical POU operating and control automata of the control system, which are executed in the Arduino board. An

example of the application of the proposed methodology for the design of a prototype of the object's temperature control system, which can be useful for teaching PLC programming, is given. The developed prototype was tested using a logical PLC and a physical prototype, which confirmed their functional compliance with the original and a reduction in the cost of the equipment by at least an order of magnitude.

Key words: programmable logic controllers, controller programming languages, control system prototyping.

Поляков Олексій Михайлович – інж. програміст компанії LineUp, студент каф. «Інформаційні системи та технології», Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Жураковський Богдан Юрійович – проф. каф. «Інформаційні системи та технології», Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Poliakov Oleksii – Software Developer LineUp company, student dep. "Information systems and technology", National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute".

Zhurakovskiyi Bogdan – professor, dep. "Information systems and technology", National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute".