

МЕТОДИКА ПРОТОТИПУВАННЯ ПРИСТРОЇВ КЕРУВАННЯ НА БАЗІ ДОДАТКУ OPENPLC

Поляков О. М.¹, Жураковський Б. Ю.²

¹ Інженер-програміст компанії LineUp, студент, Національний технічний
університет України «Київський політехнічний
інститут імені Ігоря Сікорського», Україна

² Професор, кафедри «Інформаційні системи та технології», Національний
технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського», Україна

Анотація. *Проблемою проектування пристроїв керування на основі програмованих логічних контролерів (ПЛК) є їх висока вартість і, як правило, недоступність на початковому етапі проектування. Метою дослідження є скорочення часу та вартості проектування системи шляхом створення прототипів блоків керування з програмною реалізацією алгоритмів керування мовами стандарту MEK 61131-3 та виконанням програм у платі Ардуїно. Метод дослідження полягає в декомпозиції проектних моделей операційних та керуючих автоматів пристрою керування та їх реалізації в програмному середовищі додатку OpenPLC у вигляді компонентів організації програм (POU) на мовах Ladder Diagram (LD), Function Block Diagram (FBD), Sequential Function Chart (SFC), Structured Text (ST) та Instruction List (IL). Результатом дослідження є методика створення типових POU операційних та керуючих автоматів системи керування, які виконуються у платі Arduino. Наведено приклад застосування запропонованої методики для проектування прототипу системи керування температурою об'єкту. Розроблений прототип пройшов випробування з використанням логічного ПЛК та фізичної прототипу, що підтвердило їх функціональну відповідність оригіналу та зниження вартості обладнання як мінімум на порядок.*

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

Вступ. Для керування обладнанням та виробничими процесами широко та ефективно застосовуються ПЛК, які є досить дорогим обладнанням [1]. Тому на ранніх стадіях проектування системи доцільно застосування прототипів ПЛК – функціональних аналогів реального ПЛК на базі більш дешевих мікроконтролерів, наприклад плат Ардуїно. Але для програмування ПЛК використовуються спеціальні мови за стандартом MEK 61131-3 [2], а найбільш

поширені середовища програмування, наприклад Arduino IDE 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 [3]. 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 ROU 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.

Основний матеріал. 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 MEK 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 системі. 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 прототипа оригіналу. Але, у багатьох випадках, це також корисно, наприклад при навчанні програмуванню ПЛК.

Можливо два варіанти використання прототипування системи керування:

- Спроектвано систему керування на базі ПЛК, але обраний ПЛК фізично не доступний. У цьому випадку 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 на 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 описуються у програмі як локальні класи.

На третьому етапі обираються мови програмування для ROU проекту прототипу програми, що управляє. Різні мови стандарту MEK 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 і іншими мовами. Таким чином, типова керуюча програма для прототипу системи є мультимовною.

Розглянемо приклад прототипування найпростішої дискретної системи стабілізації температури об'єкта. 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.

Проект *Prototype* включає функціональні блоки ОА і КА, 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. Графічне уявлення функціональних блоків ОА, КА та програми *LimTemp* наведено на рис. 1.

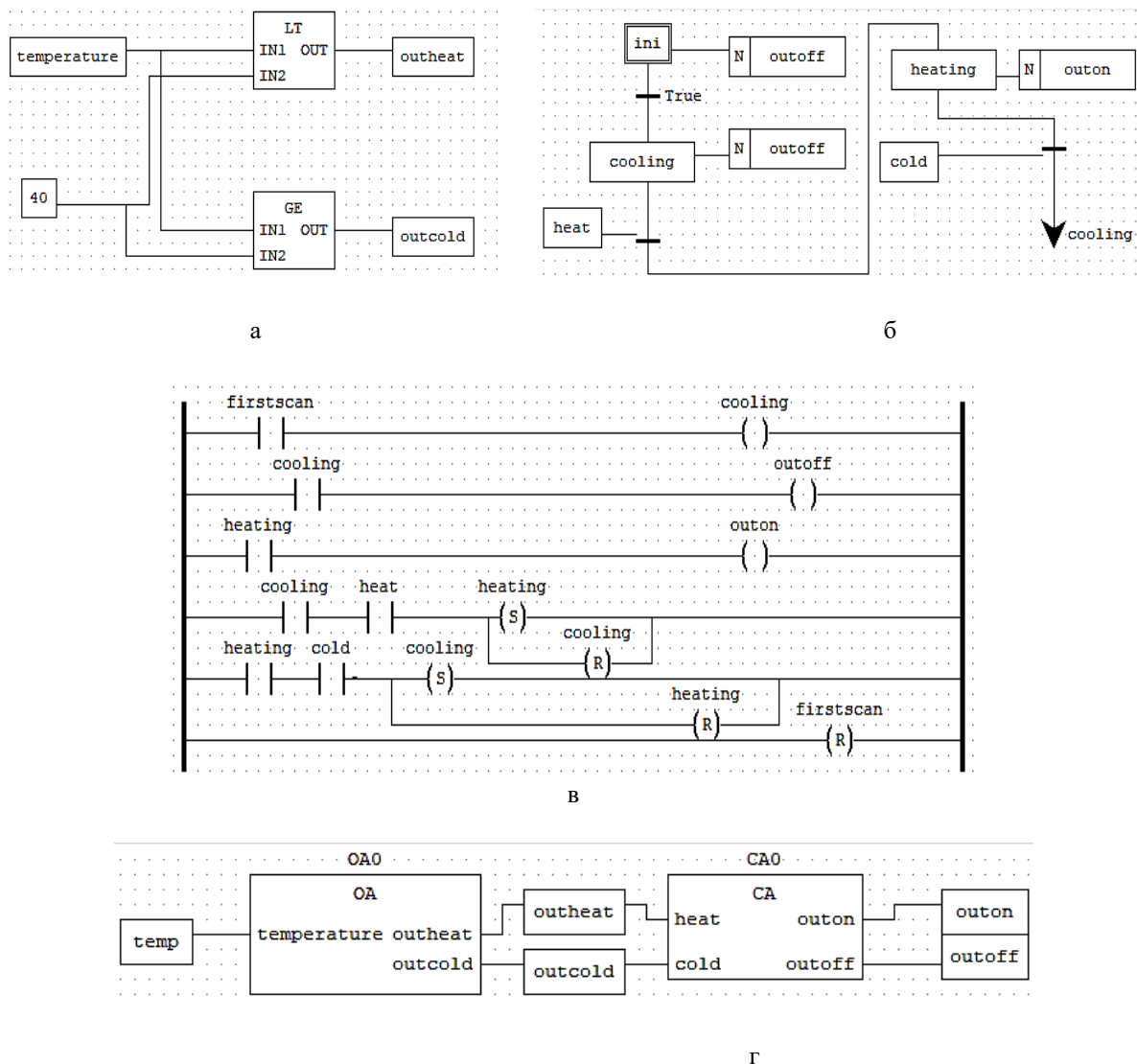


Рисунок 1 – Графічне уявлення POU: а – ОА; б – КА мовою SFC; в – КА мовою LD; г – програми *LimTemp*.

Перевірка проекту Prototype у середовищі додатку OpenPLC Editor з використанням логічного ПЛК для обох версій автомата КА підтвердила відповідність проекту вимогам системи керування. Бажане значення змінної *temp* встановлювалося у налагоджувачі у вікні "форсувати значення змінної".

Перевірка проекту *Prototype* із виконанням програмного коду в платі Ардуїно, до якої підключено 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.

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 у компоненти організації програм проекту прототипу системи.

Додаток OpenPLC дозволяє ефективно розробляти та тестувати проекти прототипів у мультимовному середовищі 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.

ЛІТЕРАТУРА

1. Parr, E. A. Programmable Controllers. An engineer's guide / E. A. Parr. 3rd ed. – Oxford: Newnes, 2003. – 429 p.
2. IEC 61131-3, Revision 3.0, February 2013 - Programmable controllers – Part 3: Programming languages. Published By: International Electrotechnical Commission (IEC). – 468 p.
3. OpenPLC Overview – Autonomy (autonomylogic.com) [Electronic resource]– Access mode: <https://autonomylogic.com/docs/openplc-overview/>

METHOD OF PROTOTYPING CONTROL UNITS BASED ON THE OPENPLC APPLICATION

Oleksii Poliakov, Bohdan Zhurakovskiy

Abstract. *The problem of designing control devices based on programmable logic controllers (PLCs) is their high cost and, as a rule, unavailability at the initial design stage. The purpose 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 control algorithms in the languages of the MEK 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 software environment of the OpenPLC application in the form of program*

organization components (POU) in Ladder Diagram, Function Block Diagram, Sequential Function Chart and Structured Text 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 facility's temperature control system 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.

Keywords: *programmable logic controllers, controller programming languages, control system prototyping.*

REFERENCE

1. Parr, E. A. Programmable Controllers. An engineer's guide / E. A. Parr. 3rd ed. – Oxford: Newnes, 2003. – 429 p.
2. IEC 61131-3, Revision 3.0, February 2013 - Programmable controllers – Part 3: Programming languages. Published By: International Electrotechnical Commission (IEC). – 468 p.
3. OpenPLC Overview – Autonomy (autonomylogic.com) [Electronic resource]– Access mode: <https://autonomylogic.com/docs/openplc-overview/>